

Guide d'administration d'Oracle VM Server for SPARC 2.0

Copyright © 2007, 2010, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Copyright © 2007, 2010, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Table des matières

Préface	13
1 Présentation du logiciel Oracle VM Server for SPARC	17
Hyperviseur et domaines logiques	17
gestionnaire de domaines logiques	19
Rôles des domaines	20
Interface de ligne de commande	21
Entrée/Sortie virtuelle	21
Configuration des ressources	22
Configurations persistantes	22
Outil de conversion physique-à-virtuel Oracle VM Server for SPARC	23
Assistant de configuration Oracle VM Server for SPARC	23
2 Installation et activation du logiciel	25
Installation du logiciel Oracle VM Server for SPARC sur un nouveau système	26
Mise à jour du SE Oracle Solaris	26
Mise à niveau du microprogramme système	26
Téléchargement du gestionnaire de domaines logiques	29
Installation du gestionnaire de domaines logiques	29
Activation du démon du gestionnaire de domaines logiques	32
Mise à niveau d'un système utilisant déjà Oracle VM Server for SPARC	33
Mise à niveau du SE Oracle Solaris	33
Mise à niveau du gestionnaire de domaines logiques et du microprogramme du système	35
Mise à niveau du logiciel Oracle VM Server for SPARC 2.0	36
Configuration usine par défaut et désactivation des domaines logiques	37
▼ Suppression de tous les domaines invités	37
▼ Suppression de toutes les configurations de domaines logiques	38

▼ Restauration de la configuration usine par défaut	38
▼ Désactivation du gestionnaire de domaines logiques	38
▼ Suppression du gestionnaire de domaines logiques	39
▼ Restauration de la configuration usine par défaut à partir du processeur de service	39
3 Sécurité	41
Autorisation du gestionnaire de domaines logiques	41
Création d'une autorisation et de profils et assignation de rôles pour les comptes utilisateur ...	42
Gestion des autorisations utilisateur	43
Gestion des profils utilisateur	44
Assignation de rôles aux utilisateurs	44
Configuration de RBAC pour l'accès à la console invité	45
Activation et utilisation du contrôle BSM	46
▼ Activation du contrôle BSM	47
▼ Vérification de l'activation du contrôle BSM	47
▼ Désactivation du contrôle BSM	47
▼ Impression de la sortie d'audit	47
▼ Rotation des journaux d'audit	48
4 Configuration des services et du domaine de contrôle	49
Messages de sortie	49
Création des services par défaut	50
▼ Création des services par défaut	50
Configuration initiale du domaine de contrôle	51
▼ Configuration du domaine de contrôle	52
Redémarrage pour utiliser domaines logiques	53
▼ Redémarrage	53
Activation de la mise en réseau entre le domaine de contrôle/service et les autres domaines ...	53
▼ Configuration du commutateur virtuel en tant qu'interface principale	53
Activation du démon du serveur de terminal du réseau virtuel	54
▼ Activation du démon du serveur de terminal du réseau virtuel	55
5 Configuration des domaines invités	57
Création et démarrage d'un domaine invité	57

▼ Création et démarrage d'un domaine invité	57
Installation du SE Oracle Solaris sur un domaine invité	60
▼ Installation du SolOS; sur un domaine invité à partir d'un DVD	61
▼ Installation du SE Oracle Solaris sur un domaine invité à partir d'un fichier ISO de Oracle Solaris	62
▼ Réalisation d'une opération JumpStart sur un domaine invité	63
6 Configuration des domaines d'E/S	65
Présentation d'un domaine d'E/S	65
Assignation de bus PCIe	66
▼ Création d'un domaine d'E/S en assignant un bus PCIe	67
Assignation des périphériques d'extrémité PCIe	71
Configuration matérielle et logicielle requise pour les E/S directes	73
Restrictions des E/S directes	73
Planification de la configuration des périphériques d'extrémité PCIe	74
Redémarrage du domaine primary	75
Procédure de modification matérielle PCIe	76
▼ Création d'un domaine d'E/S en assignant un périphérique d'extrémité PCIe	77
7 Utilisation des disques virtuels	83
Présentation des disques virtuels	83
Gestion des disques virtuels	84
▼ Ajout d'un disque virtuel	85
▼ Exportation multiples de l'arrière-plan du disque virtuel	85
▼ Modification des options du disque virtuel	86
▼ Modification de l'option de délai d'attente	86
▼ Suppression d'un disque virtuel	86
Identificateur de disque virtuel et nom de périphérique	87
Apparence du disque virtuel	87
Disque complet	88
Disque à segment unique	88
Options de moteur de traitement du disque virtuel	88
Option de lecture seule ro	89
Option exclusive excl	89
Option de segment (slice)	90

Arrière-plan du disque virtuel	90
Disque physique ou LUN de disque	90
▼ Exportation d'un disque physique en tant que disque virtuel	91
Segment de disque physique	91
▼ Exportation d'un segment de disque physique en tant que disque virtuel	92
▼ Exportation du segment 2	92
Fichier et volume	92
Configuration du multivoie d'un disque virtuel	96
▼ Configuration du multivoie d'un disque virtuel	98
CD, DVD et images ISO	99
▼ Exportation d'un CD ou d'un DVD à partir du domaine de service vers le domaine invité	100
▼ Exportation d'une image ISO à partir du domaine primary pour installation sur un domaine invité	101
Délai d'attente du disque virtuel	102
Disque virtuel et SCSI	103
Disque virtuel et commande format(1M)	104
Utilisation de ZFS avec les disques virtuels	104
Configuration d'un pool ZFS dans un domaine de service	104
Stockage des images de disque avec ZFS	105
Création d'un instantané d'une image de disque	106
Utilisation du clone pour mettre à disposition un nouveau domaine	107
Utilisation du gestionnaire de volumes dans un environnement de domaines logiques	108
Utilisation de disques virtuels au sommet de gestionnaires de volumes	109
Utilisation des gestionnaires de volumes au sommet de disques virtuels	111
8 Utilisation des réseaux virtuels	113
Introduction au réseau virtuel	113
Commutateur virtuel	114
Périphérique réseau virtuel	114
Gestion d'un commutateur virtuel	116
▼ Ajout d'un commutateur virtuel	116
▼ Définition des options pour un commutateur virtuel existant	117
▼ Suppression d'un commutateur virtuel	117
Gestion d'un périphérique réseau virtuel	118
▼ Ajout d'un périphérique réseau virtuel	118

▼ Définition des options pour un périphérique réseau virtuel existant	119
▼ Suppression d'un périphérique réseau virtuel	119
Identificateur de périphérique virtuel et nom d'interface réseau	120
▼ Recherche du nom de l'interface réseau du SE Oracle Solaris	121
Assignation automatique et manuelle des adresses MAC	122
Plage d'adresses MAC assignées aux domaines logiques	123
Algorithme d'assignation automatique	123
Détection des adresses MAC en doublon	124
Adresses MAC libérées	124
Utilisation des adaptateurs réseau avec les domaines logiques	125
▼ Procédure de détermination de la compatibilité GLDv3 d'un adaptateur réseau	125
Configuration du commutateur virtuel et du domaine de service pour le NAT et le routage ..	126
▼ Configuration d'un commutateur virtuel pour fournir une connectivité externe aux domaines	127
Configuration d'IPMP dans un environnement de domaines logiques	128
Configuration des périphériques de réseau virtuel dans un groupe IPMP dans un domaine	128
Configuration et utilisation d'IPMP dans le domaine de service	129
Utilisation de l'IPMP basé sur liaison dans la mise en réseau virtuelle des domaines logiques	130
Configuration et utilisation d'IPMP dans les versions antérieures aux domaines logiques 1.3	133
Utilisation du balisage VLAN	135
ID du VLAN du port (PVID)	136
ID de VLAN (VID)	137
▼ Assignation de VLAN à un commutateur virtuel et à un périphérique de réseau virtuel .	137
▼ Installation d'un domaine invité lorsque le serveur d'installation est un VLAN	138
Utilisation des E/S hybrides NIU	139
▼ Configuration d'un commutateur virtuel avec un périphérique réseau NIU	141
▼ Activation du mode hybride	142
▼ Désactivation du mode hybride	142
Utilisation du groupement de liens avec un commutateur virtuel	142
Configuration des cadres Jumbo	144
▼ Configuration du réseau virtuel et des périphériques de réseau virtuel pour utiliser les cadres Jumbo	144
Compatibilité avec des versions antérieures (non connaissant pas Jumbo) des pilotes vnet et vsw	147

9 Migration des domaines	149
Introduction à la migration de domaines	149
Présentation d'une opération de migration	150
Compatibilité logicielle	150
Authentification pour les opérations de migration	151
Migration d'un domaine	151
Réalisation d'une simulation	151
Réalisation de migrations non interactives	151
Migration d'un domaine actif	152
Migration des CPU dans un domaine actif	152
Migration des CPU dans un domaine actif	153
Migration des périphériques d'E/S physiques dans un domaine actif	154
Migration des périphériques d'E/S virtuels dans un domaine actif	154
Migration d'une entrée/sortie hybride NIU sur un domaine actif	155
Migration des unités cryptographiques dans un domaine actif	156
Reconfiguration retardée dans un domaine actif	156
Migration pendant qu'un domaine actif est en mode élastique	156
Opérations sur d'autres domaines	157
Migration de domaines associés ou inactifs	157
Migration de CPU dans un domaine associé ou inactif	157
Migration d'une entrée/sortie virtuelle dans un domaine associé ou inactif	157
Migration de périphériques d'extrémité PCIe dans un domaine associé ou inactif	157
Surveillance d'une migration en cours	158
Annulation d'une migration en cours	159
Récupération sur un échec de migration	159
Exemples de migration	160
10 Gestion des ressources	163
Reconfiguration des ressources	163
Reconfiguration dynamique	163
Reconfiguration retardée	164
Allocation des ressources	165
Allocation de CPU	165
Activation de la contrainte Whole-Core	166
Désactivation de la contrainte Whole-Core	167

Allocation de CPU au domaine de contrôle	167
Interactions entre la contrainte Whole-core et les fonctions des autres domaines	168
Utilisation de la reconfiguration dynamique de la mémoire	169
Ajout de mémoire	170
Suppression de mémoire	170
Demandes partielles de reconfiguration dynamique de mémoire	171
Reconfiguration de la mémoire du domaine de contrôle	171
Reconfiguration dynamique et retardée	172
Alignement de la mémoire	172
Exemples de reconfiguration dynamique de mémoire	174
Utilisation de la gestion de l'alimentation	178
Liste des brins avec gestion de l'alimentation du CPU	179
Utilisation de la gestion dynamique des ressources	181
Liste des ressources du domaine	184
Sortie lisible par la machine	184
Définitions des balises	185
Utilisation de la définition des statistiques	185
Affichage des différentes listes	186
Liste des contraintes	189
11 Gestion des configurations	191
Enregistrement des configurations de domaine pour régénération ultérieure	191
▼ Enregistrement des configurations de domaine	192
▼ Restauration d'une configuration de domaine à partir d'un fichier XML (<code>ldm</code> <code>add-domain</code>)	192
▼ Restauration d'une configuration de domaine à partir d'un fichier XML (<code>ldm</code> <code>init-system</code>)	193
Gestion des configurations des domaines logiques	194
▼ Modification de la stratégie de récupération automatique	196
12 Réalisation d'autres tâches d'administration	199
Entrée de noms dans la CLI	199
Noms de fichier (<i>file</i>) et noms de variable (<i>var-name</i>)	199
<i>backend</i> du serveur de disque virtuel et noms de périphérique de commutateur virtuel ..	200
Nom de configuration (<i>config-name</i>)	200

Tous les autres noms	200
Connexion à une console invitée sur un réseau	200
Utilisation de groupes de consoles	201
▼ Association de plusieurs consoles en un groupe	201
Délai d'arrêt d'un domaine fortement chargé pouvant être dépassé	202
Utilisation du SE Oracle Solaris avec Oracle VM Server for SPARC	202
Microprogramme OpenBoot indisponible une fois que le SE Oracle Solaris a démarré ..	202
Cycle d'arrêt et de redémarrage d'un serveur	203
N'utilisez pas la commande <code>ps radm(1M)</code> sur les CPU actifs dans domaine avec gestion de l'alimentation	203
Résultat des interruptions du SE Oracle Solaris	203
Résultats de l'arrêt ou de la réinitialisation du domaine de contrôle	203
Utilisation des domaines logiques avec le processeur de service	204
▼ Réinitialisation de la configuration du domaine à la configuration par défaut ou à une configuration différente	205
Configuration des dépendances de domaine	205
Exemples de dépendance de domaine	207
Cycles de dépendance	208
Détermination de l'endroit où les erreurs sont survenues lors du mappage du CPU et des adresses de mémoire	209
Mappage du CPU	209
Mappage de la mémoire	210
Exemples de mappage de CPU et de mémoire	210
Utilisation des identificateurs uniques universellement	212
Commande et API d'information sur le domaine virtuel	212
A Outil de conversion physique-à-virtuel Oracle VM Server for SPARC	215
Présentation de l'outil P2V Oracle VM Server for SPARC	215
Phase de collecte	216
Phase de préparation	216
Phase de conversion	217
Périphériques du moteur de traitement	217
Installation de l'outil P2V Oracle VM Server for SPARC	218
Conditions requises	219
Restrictions	219
▼ Installation de l'outil P2V Oracle VM Server for SPARC	220

Utilisation de la commande <code>ldmp2v</code>	221
B Assistant de configuration Oracle VM Server for SPARC	227
Utilisation de l'assistant de configuration (interface graphique)	227
Utilisation de l'assistant de configuration (<code>ldmconfig</code>)	228
Installation de l'assistant de configuration	228
<code>ldmconfig</code> : Fonctions	229
C Recherche du gestionnaire de domaines logiques	233
Recherche des systèmes exécutant le gestionnaire de domaines logiques	233
Communication en multidiffusion	233
Format du message	234
▼ Recherche du gestionnaire de domaines logiques s'exécutant sur votre sous-réseau	234
D Utilisation de l'interface XML avec le gestionnaire de domaines logiques	237
Transport XML	237
Serveur XMPP	238
Connexions locales	238
Protocole XML	238
Messages de demande et de réponse	239
Messages d'événements	243
Enregistrement et annulation de l'enregistrement	243
Messages <code><LDM_event></code>	244
Types d'événements	244
Actions du gestionnaire de domaines logiques	247
Ressources et propriétés du gestionnaire de domaines logiques	248
Ressource d'informations sur le domaine (<code>ldom_info</code>)	249
Ressource du CPU (<code>cpu</code>)	250
Ressource MAU (<code>mau</code>)	251
Ressource de mémoire (<code>memory</code>)	251
Ressource de serveur de disque virtuel (<code>vds</code>)	252
Ressource du volume de disque virtuel (<code>vds_volume</code>)	252
Ressource de disque (<code>disk</code>)	253
Ressource de commutateur virtuel (<code>vsw</code>)	253

Ressource réseau (network)	254
Ressource de concentrateur de console virtuelle (vcc)	255
Ressource de variable (var)	256
Ressource de périphérique d'E/S physique (physio_devise)	256
Ressource de configuration du SP (spconfig)	257
Ressource de service de canal de plan de données virtuelles (vdpcs)	257
Ressource de client de canal de plan de données virtuelles (vdpc)	258
Ressource de console (console)	259
Migration de domaine	259
E Schémas XML du gestionnaire de domaines logiques	261
LDM_interface (Schéma XML)	261
LDM_Event (Schéma XML)	263
Schéma ovf-envelope.xsd	264
Schéma ovf-section.xsd	266
Schéma ovf-core.xsd	266
Schéma ovf-virtualhardware.xsc	271
Schéma cim-rasd.xsd	272
Schéma cim-vssd.xsd	276
Schéma cim-common.xsd	276
Schéma XML GenericProperty	280
Binding_Type (Schéma XML)	280
Glossaire	283
Index	293

Préface

Le *Guide d'administration d'Oracle VM Server for SPARC 2.0* fournit des informations et des procédures détaillées qui décrivent la présentation, les considérations relatives à la sécurité, l'installation, la configuration, la modification et l'exécution des tâches communes pour le logiciel Oracle VM Server for SPARC 2.0 sur les serveurs, lames et modules serveur pris en charge. Reportez-vous à la section “Plates-formes prises en charge” des *Notes de version d'Oracle VM Server for SPARC 2.0*.

Ce guide est destiné aux administrateurs système de ces serveurs qui ont des connaissances des systèmes UNIX et du système d'exploitation Oracle Solaris (SE Oracle Solaris).

Documentation connexe

Le tableau suivant présente la documentation disponible pour la version Oracle VM Server for SPARC 2.0. Ces documents sont disponibles aux formats HTML et PDF, sauf mention contraire.

TABLEAU P-1 Documentation connexe

Application	Titre	Référence
Logiciel Oracle VM Server for SPARC 2.0	<i>Guide d'administration d'Oracle VM Server for SPARC 2.0</i>	821-2932
	<i>Notes de version Oracle VM Server for SPARC 2.0</i>	821-2936
	<i>Oracle VM Server for SPARC 2.0 Reference Manual</i>	821-1486
	Ensemble de documents de référence Oracle Solaris 10	
	<ul style="list-style-type: none">■ <i>drd(1M)</i> (page de manuel)■ <i>vntsd(1M)</i> (page de manuel)	
Notions de base des domaines logiques	<i>Beginners Guide to LDOMs: Understanding and Deploying Logical Domains Software</i> (PDF)	820-0832
domaines logiques Management Information Base (MIB)	<i>Logical Domains (LDoms) MIB 1.0.1 Administration Guide</i>	820-2319-10
	<i>Logical Domains (LDoms) MIB 1.0.1 Release Notes</i>	820-2320-10

TABLEAU P-1 Documentation connexe (Suite)

Application	Titre	Référence
SE Oracle Solaris: Installation and Configuration	Oracle Solaris 10 9/10 Release and Installation Collection	S/O

Vous pouvez trouver des documents relatifs à votre serveur, logiciel ou au SE Oracle Solaris à l'adresse <http://docs.sun.com>. Utilisez la zone de recherche pour rechercher les documents et les informations dont vous avez besoin.

Documentation, support et formation

Consultez les sites Web suivants pour obtenir des documents supplémentaires :

- [Documentation \(http://docs.sun.com\)](http://docs.sun.com)
- [Support \(http://www.oracle.com/us/support/systems/index.html\)](http://www.oracle.com/us/support/systems/index.html)
- [Formation \(http://education.oracle.com\)](http://education.oracle.com) – Cliquez sur le lien Sun dans la barre de navigation à gauche.

Vos commentaires sont les bienvenus

Oracle se réjouit de vos commentaires et suggestions sur la qualité et l'utilité de sa documentation. Si vous trouvez des erreurs ou avez des suggestions d'amélioration, accédez à la page <http://docs.sun.com> et cliquez sur Feedback. Indiquez le titre et le numéro de référence de la documentation ainsi que le chapitre, la section et le numéro de page, si disponible. Faites-nous savoir si vous souhaitez une réponse.

[Oracle Technology Network \(http://www.oracle.com/technetwork/index.html\)](http://www.oracle.com/technetwork/index.html) offre un grand nombre de ressources relatives aux logiciels Oracle :

- Discutez de problèmes techniques et de solutions sur les [Forums de discussion \(http://forums.oracle.com\)](http://forums.oracle.com).
- Obtenez des didacticiels détaillés dans [Oracle By Example \(http://www.oracle.com/technetwork/tutorials/index.html\)](http://www.oracle.com/technetwork/tutorials/index.html).
- Téléchargez des [Exemples de code \(http://www.oracle.com/technology/sample_code/index.html\)](http://www.oracle.com/technology/sample_code/index.html).

Conventions typographiques

Le tableau suivant décrit les conventions typographiques utilisées dans ce manuel.

TABLEAU P-2 Conventions typographiques

Police de caractère	Signification	Exemple
AaBbCc123	Noms de commandes, de fichiers et de répertoires et informations affichées à l'écran.	Modifiez le fichier <code>.login</code> . Utilisez <code>ls -a</code> pour dresser la liste des fichiers. <code>machine_name% you have mail.</code>
AaBbCc123	Ce que vous tapez, mis en évidence par rapport aux informations affichées à l'écran.	<code>machine_name% su</code> <code>Password:</code>
<i>aabbcc123</i>	Paramètre substituable : à remplacer par une valeur ou un nom réel	La commande permettant de supprimer un fichier est <code>rm nomfichier</code> .
<u>AaBbCc123</u>	Titres de livre, nouveaux termes et termes à souligner	Lisez le chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie stockée localement. N'enregistrez <i>pas</i> le fichier . Remarque : Certains éléments mis en évidence apparaissent en caractères gras.

Invites de shell dans les exemples de commande

Le tableau suivant présente l'invite du système UNIX par défaut et l'invite du superutilisateur pour les shells inclus dans le SE Oracle Solaris. Notez que l'invite du système par défaut qui est affiché dans les exemples de commande varie en fonction de la version d'Oracle Solaris.

TABLEAU P-3 Invites de shell

Shell	Invite
Shell Bash, shell Korn et shell Bourne	\$
Shell Bash, shell Korn et shell Bourne pour un superutilisateur	#
Shell C	machine_name%

TABLEAU P-3 Invites de shell (Suite)

Shell	Invite
Shell C pour un superutilisateur	machine_name#

Présentation du logiciel Oracle VM Server for SPARC

Ce chapitre fournit une présentation du logiciel Oracle VM Server for SPARC.

Le logiciel Oracle VM Server for SPARC dépend de versions spécifiques du SE Oracle Solaris, des patches logiciels requis et de versions spécifiques du microprogramme du système. Pour plus d'informations, reportez-vous à la section “SE Oracle Solaris requis et recommandé” des *Oracle VM Server for SPARC 2.0 Release Notes (Notes de version Oracle VM Server for SPARC 2.0)*.

Ce chapitre aborde les sujets suivants :

- “Hyperviseur et domaines logiques” à la page 17
- “gestionnaire de domaines logiques” à la page 19
- “Outil de conversion physique-à-virtuel Oracle VM Server for SPARC” à la page 23
- “Assistant de configuration Oracle VM Server for SPARC” à la page 23

Hyperviseur et domaines logiques

Cette section fournit une présentation de l'hyperviseur SPARC prenant en charge les domaines logiques.

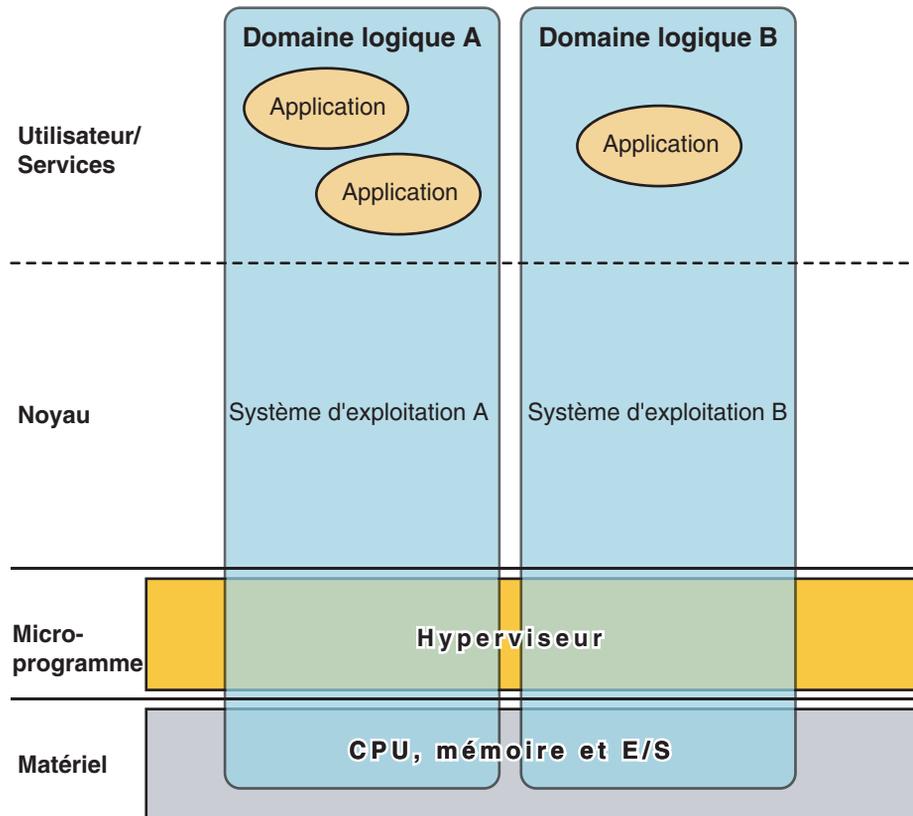
Le SPARC *hyperviseur* est une petite couche de microprogramme qui fournit une architecture de machine virtualisée stable sur laquelle on peut installer un système d'exploitation. Le serveurs Sun d'Oracle utilisant cet hyperviseur fournissent des fonctions matérielles qui prennent en charge le contrôle de l'hyperviseur sur les activités du système d'exploitation logique.

Un *domaine logique* est une machine virtuelle comprenant un groupement logique discret de ressources. Un domaine logique a son propre système d'exploitation et sa propre identité dans un système informatique unique. Chaque domaine logique peut être créé, supprimé, reconfiguré et réinitialisé individuellement, sans avoir à exécuter une remise sous tension du serveur. Il est possible d'exécuter une grande variété d'applications dans des domaines logiques différents et de préserver l'indépendance de ceux-ci à des fins de performances ou de sécurité.

Chaque domaine logique est uniquement autorisé à observer et à interagir avec les ressources du serveur qui sont mises à sa disposition par l'hyperviseur. Le gestionnaire de domaines logiques vous permet de spécifier ce que l'hyperviseur doit faire dans le domaine logique. Ainsi, l'hyperviseur force le partitionnement des ressources du serveur et fournit des sous-ensembles limités à plusieurs environnements de système d'exploitation. Ce partitionnement et cette mise à disposition sont le mécanisme fondamental de création des domaines logiques. Le schéma suivant représente l'hyperviseur prenant en charge deux domaines logiques. Il montre également les couches suivantes constituant la fonctionnalité des domaines logiques :

- Applications ou utilisateur/services
- Noyau ou systèmes d'exploitation
- Microprogramme ou hyperviseur
- Matériel, y compris le CPU, la mémoire et les E/S

FIGURE 1-1 Hyperviseur prenant en charge deux domaines



Le nombre et les fonctions de chaque domaine logique qu'un hyperviseur SPARC spécifique prend en charge sont des fonctions dépendantes du serveur. L'hyperviseur peut allouer des sous-ensembles de ressources globales de CPU, mémoire et E/S d'un serveur à un domaine logique donné. Cela permet de prendre en charge plusieurs systèmes d'exploitation en même temps, chacun dans son propre domaine logique. Les ressources peuvent être réorganisées entre des domaines logiques distincts avec une granularité arbitraire. Par exemple, des CPU sont assignés à un domaine logique avec la granularité d'un thread de CPU.

Chaque domaine logique peut être géré comme une machine totalement indépendante avec ses propres ressources, notamment :

- le noyau, les patches et les paramètres de réglage ;
- les comptes utilisateur et les administrateurs ;
- les disques ;
- les interfaces réseau, les adresses MAC et IP.

Chaque domaine logique peut être arrêté, démarré et redémarré indépendamment des autres sans nécessiter une remise sous tension du serveur.

Le logiciel de l'hyperviseur est responsable du maintien de la séparation entre les domaines logiques. Le logiciel de l'hyperviseur fournit également les canaux de domaine logique (LDC) qui permettent aux domaines logiques de communiquer les uns avec les autres. Les LCD permettent aux domaines de se fournir des services mutuellement, notamment des services de mise en réseau ou de disque.

Le processeur de service (SP), également connu sous le nom de contrôleur système (SC), surveille et exécute la machine virtuelle, mais il ne gère pas les domaines logiques. Le gestionnaire de domaines logiques gère les domaines logiques.

gestionnaire de domaines logiques

Le gestionnaire de domaines logiques est utilisé pour créer et gérer les domaines logiques, ainsi que pour mapper les domaines logiques sur des ressources physiques. Seul un gestionnaire de domaines logiques peut s'exécuter sur un serveur.

Rôles des domaines

Tous les domaines logiques sont identiques et peuvent se différencier les uns des autres en fonction de rôles que vous définissez pour eux. Vous trouverez ci-dessous les rôles que les domaines logiques peuvent avoir :

- **Domaine de contrôle.** Le gestionnaire de domaines logiques s'exécute dans ce domaine, ce qui vous permet de créer et de gérer d'autres domaines logiques et d'allouer des ressources virtuelles aux autres domaines. Vous ne pouvez avoir qu'un seul domaine de contrôle par serveur. Le domaine de contrôle est le premier domaine créé lorsque vous installez le logiciel Oracle VM Server for SPARC. Le domaine de contrôle est nommé `primary`.
- **Domaine de service.** Un domaine de service fournit des services de périphérique virtuel aux autres domaines, notamment en tant que commutateur virtuel, concentrateur de console virtuelle ou serveur de disque virtuel. N'importe quel domaine peut être configuré en tant que domaine de service.
- **Domaine d'E/S.** Un domaine d'E/S a un accès direct à un périphérique d'E/S physique, notamment une carte réseau dans un contrôleur PCI EXPRESS (PCIe). Une domaine d'E/S peut posséder un complexe racine PCIe ou il peut posséder un emplacement PCIe ou un périphérique PCIe intégré à l'aide de la fonction d'E/S directe (DIO). Voir "[Assignation des périphériques d'extrémité PCIe](#)" à la page 71.

Un domaine d'E/S peut partager des périphériques d'E/S physiques avec d'autres domaines sous la forme de périphériques virtuels lorsque le domaine d'E/S est également utilisé en tant que domaine de service.

- **Domaine racine.** Un domaine racine a un complexe racine PCIe qui lui est assigné. Ce domaine possède la fabric PCIe et fournit tous les services associés à la fabric, notamment le traitement des erreurs fabric. Un domaine racine est également un domaine d'E/S, car il possède et a un accès direct aux périphériques d'E/S.

Le nombre de domaines racines que vous pouvez avoir dépend de l'architecture de votre plate-forme. Par exemple, si vous utilisez un serveur Sun SPARC Enterprise T5440, vous pouvez avoir jusqu'à quatre domaines racine.

- **Domaine invité.** Un domaine invité est un domaine non E/S qui consomme des services de périphérique virtuel fournis par un ou plusieurs domaines. Un domaine invité n'a pas de périphérique d'E/S physiques, mais a uniquement des périphériques d'E/S virtuels, notamment des disques virtuels et des interfaces réseau virtuelles.

Vous pouvez installer le gestionnaire de domaines logiques sur un système existant qui n'est pas déjà configuré avec les domaines logiques. Dans ce cas, l'instance actuelle du SE devient le domaine de contrôle. Le système est également configuré en tant que système de domaines logiques qui n'a qu'un seul domaine, le domaine de contrôle. Après la configuration du domaine de contrôle, vous pouvez équilibrer la charge des applications au sein des autres domaines pour une utilisation optimale de tout le système. Vous effectuez cette opération en ajoutant des domaines et en déplaçant ces applications du domaine de contrôle vers les nouveaux domaines.

Interface de ligne de commande

Le gestionnaire de domaines logiques utilise une interface de ligne de commande (CLI) pour créer et configurer des domaines logiques. La CLI est une commande simple, `ldm`, ayant plusieurs sous-commandes. Reportez-vous à la page de manuel `ldm(1M)`.

Le démon du gestionnaire de domaines logiques, `ldmd`, doit être en cours d'exécution pour utiliser la CLI du gestionnaire de domaines logiques.

Entrée/Sortie virtuelle

Dans un environnement de domaines logiques, vous pouvez mettre à disposition jusqu'à 128 domaine sur un système de processeur UltraSPARC T2 Plus et un système de processeur SPARC T3. Ces systèmes ont un nombre limité de bus d'E/S et d'emplacements d'E/S physiques. En conséquence, vous ne pouvez pas fournir un accès exclusif à des périphériques physiques de disque ou réseau à tous les domaines sur ces systèmes. Vous pouvez assigner un bus PCIe ou un périphérique d'extrémité à un domaine pour lui fournir l'accès à un périphérique physique. Notez que cette solution est insuffisante pour fournir un accès exclusif au périphérique à tous les domaines. Reportez-vous au [Chapitre 6, "Configuration des domaines d'E/S"](#). Cette limitation sur le nombre de périphériques d'E/S physiques à accès direct est traitée en implémentant un modèle d'E/S virtualisé.

Les domaines logiques n'ayant pas d'accès E/S physique sont configurés avec des périphériques d'E/S virtuels communiquant avec un domaine de service. Le domaine de service exécute le service de périphérique virtuel pour fournir l'accès au périphérique physique ou à ses fonctions. Dans ce modèle client-serveur, les périphériques d'E/S virtuels communiquent les uns avec les autres ou avec un service homologue via les canaux de communication interdomaine appelés canaux de domaine logique (LDC). La fonctionnalité d'E/S virtualisée comprend la prise en charge de la mise en réseau, du stockage et des consoles virtuels.

Réseau virtuel

Les domaines logiques utilisent le périphérique réseau virtuel et le périphérique de commutation de réseau virtuel pour implémenter la mise en réseau virtuelle. Le périphérique réseau virtuel (`vnet`) émule un périphérique Ethernet et communique avec les autres périphériques `vnet` dans le système à l'aide d'un canal point à point. Le périphérique de commutateur virtuel (`vsw`) fonctionne comme un multiplexeur de tous les paquets entrants et sortants du réseau virtuel. Le périphérique `vsw` communique directement avec un adaptateur réseau physique sur un domaine de service, et envoie et reçoit des paquets pour le compte d'un réseau virtuel. Le périphérique `vsw` fonctionne également comme un commutateur simple de couche 2 et commute les paquets entre les périphériques `vnet` connectés à celui-ci dans le système.

Stockage virtuel

L'infrastructure de stockage virtuel utilise un modèle client-serveur pour activer les domaines logiques afin d'accéder au stockage au niveau du bloc qui n'est pas directement assigné à ces derniers. Le modèle utilise les composants suivants :

- Client de disque virtuel (vdc) qui exporte l'interface de périphérique de bloc
- Service de disque virtuel (vds) qui traite les demandes de disque pour le compte du client de disque virtuel et les envoie au stockage du moteur de traitement qui réside sur le domaine de service

Bien que les disques virtuels apparaissent comme des disques classiques sur le domaine client, la plupart des opérations de disque sont transmises au service de disque virtuel et traitées sur le domaine de service.

Console virtuelle

Dans un environnement de domaines logiques, la console d'E/S du domaine `primary` est dirigée vers le processeur de service. Les E/S de la console de tous les autres domaines sont redirigées vers le domaine de service exécutant le concentrateur de console virtuelle (vcc). Le domaine exécutant le vcc est généralement le domaine `primary`. Le service du concentrateur de console virtuelle fonctionne comme un concentrateur pour tout le trafic de la console du domaine et communique avec le démon du serveur de terminal réseau virtuel (vntsd) pour fournir l'accès à chaque console via le socket UNIX.

Configuration des ressources

Une système exécutant le logiciel Oracle VM Server for SPARC peut configurer des ressources, notamment des CPU virtuels, des périphériques d'E/S virtuels, des unités cryptographiques et de la mémoire. Certaines ressources peuvent être configurées de manière dynamique sur un domaine en cours d'exécution, tandis que d'autres doivent être configurées sur un domaine arrêté. Si une ressource ne peut pas être configurée de manière dynamique sur le domaine de contrôle, vous devez d'abord lancer une reconfiguration retardée. La reconfiguration retardée reporte les activités de configuration à après le redémarrage du domaine de contrôle. Pour plus d'informations, reportez-vous à la section [“Reconfiguration des ressources” à la page 163](#).

Configurations persistantes

Vous pouvez utiliser la commande `ldm` pour stocker la configuration actuelle d'un domaine logique sur le processeur de service. Vous pouvez ajouter une configuration, spécifier la configuration à utiliser, supprimer une configuration et répertorier les configurations. Reportez-vous à la page de manuel `ldm(1M)`. Vous pouvez également indiquer une configuration pour démarrer à partir du SP. Reportez-vous à la section [“Utilisation des domaines logiques avec le processeur de service” à la page 204](#).

Pour plus d'informations sur la gestion des configurations, reportez-vous à la section “[Gestion des configurations des domaines logiques](#)” à la page 194.

Outil de conversion physique-à-virtuel Oracle VM Server for SPARC

L'outil de conversion physique-à-virtuel (P2V) Oracle VM Server for SPARC convertit automatiquement un système physique existant en système virtuel qui s'exécute dans un domaine logique sur un système multithreading de puce (CMT). Le système source peut être l'un des suivants :

- Un système sun4u SPARC qui exécute au moins le système d'exploitation Solaris 8
- Un système sun4v qui exécute le SE Oracle Solaris 10, mais n'exécute pas le logiciel Oracle VM Server for SPARC

Pour plus d'informations sur l'outil et son installation, reportez-vous à l'[Annexe A](#), “[Outil de conversion physique-à-virtuel Oracle VM Server for SPARC](#)”. Pour plus d'informations sur la commande `ldmp2v`, reportez-vous à la page de manuel `ldmp2v(1M)`.

Assistant de configuration Oracle VM Server for SPARC

L'assistant de configuration Oracle VM Server for SPARC vous guide tout au long de la configuration d'un domaine logique en paramétrant les propriétés de base. Il peut être utilisé pour configurer un système sur lequel le logiciel Oracle VM Server for SPARC est installé, mais pas encore configuré.

Après avoir rassemblé les données de configuration, l'assistant de configuration crée une configuration adaptée à l'initialisation en tant que domaine logique. Vous pouvez également utiliser les valeurs par défaut sélectionnées par l'assistant de configuration afin de créer une configuration système utilisable.

L'assistant de configuration est disponible sous forme d'interface graphique (GUI) et d'outil basé sur un terminal.

Pour plus d'informations, reportez-vous à l'[Annexe B](#), “[Assistant de configuration Oracle VM Server for SPARC](#)” et à la page de manuel `ldmconfig(1M)`.

Installation et activation du logiciel

Ce chapitre décrit la procédure d'installation ou de mise à niveau des différents composants logiciels requis pour activer le logiciel Oracle VM Server for SPARC 2.0. L'utilisation du logiciel Oracle VM Server for SPARC nécessite les composants suivants :

- Plate-forme prise en charge, reportez-vous à la section “Plates-formes prises en charge” des *Notes de version d'Oracle VM Server for SPARC 2.0* pour obtenir la liste des plates-formes prises en charge.
- Domaine de contrôle exécutant un système d'exploitation au moins équivalent au SE Oracle Solaris 10 9/10 avec les patches recommandés dans la section “Logiciels et patches recommandés” des *Notes de version Oracle VM Server for SPARC 2.0*. Reportez-vous à la section “Mise à niveau du SE Oracle Solaris” à la page 33.
- Un microprogramme système version 7.3.0 au moins pour votre plate-forme Sun UltraSPARC T2 ou T2 Plus et version 8.0.0 pour votre plate-forme SPARC T3. Reportez-vous à la section “Mise à niveau du microprogramme système” à la page 26.
- Logiciel Oracle VM Server for SPARC 2.0 installé et activé sur le domaine de contrôle. Reportez-vous à la section “Installation du gestionnaire de domaines logiques” à la page 29.
- (Facultatif) le package logiciel MIB (Management Information Base) Oracle VM Server for SPARC. Reportez-vous au *Logical Domains (LDDoms) MIB 1.0.1 Administration Guide* pour plus d'informations sur l'utilisation de la MIB des domaines logiques.

Le SE Oracle Solaris et le microprogramme système doivent être installés ou mis à niveau sur votre serveur avant d'installer ou de mettre à niveau le gestionnaire de domaines logiques. Si votre système utilise déjà le logiciel Oracle VM Server for SPARC, reportez-vous à la section “Mise à niveau d'un système utilisant déjà Oracle VM Server for SPARC” à la page 33. Sinon, reportez-vous à la section “Installation du logiciel Oracle VM Server for SPARC sur un nouveau système” à la page 26.

Ce chapitre aborde les sujets suivants :

- “Installation du logiciel Oracle VM Server for SPARC sur un nouveau système” à la page 26
- “Mise à niveau d'un système utilisant déjà Oracle VM Server for SPARC” à la page 33

- “Configuration usine par défaut et désactivation des domaines logiques” à la page 37

Remarque – Le logiciel Solaris Security Toolkit (SST) n'est plus fourni avec le logiciel Oracle VM Server for SPARC. Si vous souhaitez utiliser la version la plus récente du logiciel SST, reportez-vous aux *Notes de version d'Oracle VM Server for SPARC 2.0*.

Installation du logiciel Oracle VM Server for SPARC sur un nouveau système

Les plates-formes Sun d'Oracle qui prennent en charge le logiciel Oracle VM Server for SPARC sont fournies avec le SE Oracle Solaris 10 préinstallé. Initialement, la plate-forme apparaît comme un seul système hébergeant uniquement un système d'exploitation. Après l'installation du SE Oracle Solaris, du microprogramme système et du gestionnaire de domaines logiques, le système et l'instance d'origine du SE Oracle Solaris deviennent le domaine de contrôle. Ce premier domaine de la plate-forme est nommé `primary` et vous ne pouvez pas modifier ce nom ou détruire ce domaine. À partir de ce moment, la plate-forme peut être reconfigurée pour avoir plusieurs domaines hébergeant différentes instances du SE Oracle Solaris.

Mise à jour du SE Oracle Solaris

Sur un système totalement nouveau, vous pouvez installer le SE afin qu'il se conforme à votre stratégie d'installation. Dans ce cas, reportez-vous à la section “SE Oracle Solaris requis et recommandé” des *Notes de version Oracle VM Server for SPARC 2.0* pour rechercher le SE Oracle Solaris 10 que vous devez utiliser pour cette version du logiciel Oracle VM Server for SPARC. Reportez-vous au guide d'installation du SE Oracle Solaris 10 pour obtenir des instructions complètes sur l'installation du SE Oracle Solaris. Vous pouvez adapter votre installation aux besoins de votre système.

Si votre système est déjà installé, il doit être mis à niveau au SE Oracle Solaris 10 correspondant devant être utilisé pour cette version du logiciel Oracle VM Server for SPARC. Reportez-vous à la section “Logiciels et patches requis” des *Notes de version Oracle VM Server for SPARC 2.0* pour trouver le SE Oracle Solaris 10 que vous devez utiliser pour cette version du logiciel Oracle VM Server for SPARC et les patches requis et recommandés. Consultez le site [Collection de manuels d'installation et de mise à jour Solaris 10 9/10 \(http://docs.sun.com/app/docs/coll/1236.11\)](http://docs.sun.com/app/docs/coll/1236.11) pour obtenir des instructions complètes de mise à niveau du SE.

Mise à niveau du microprogramme système

Les tâches suivantes décrivent comment mettre à niveau le microprogramme système à l'aide du logiciel ILOM (Integrated Lights Out Manager).

Pour plus d'informations sur la mise à niveau du microprogramme système à l'aide du logiciel ILOM, reportez-vous à la section "Mise à jour du microprogramme" dans l'*Sun SPARC Enterprise T5120 and T5220 Servers Topic Set* et la section "Mise à jour du microprogramme ILOM" dans le *Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide (Guide des procédures de la CLI Sun Integrated Lights Out Manager (ILOM) 3.0)* (<http://dlc.sun.com/pdf/820-6412-12/820-6412-12.pdf>).

▼ Mise à niveau du microprogramme du système

Vous pouvez trouver le microprogramme du système pour votre plate-forme à l'adresse <http://www.oracle.com/technetwork/systems/patches/firmware/index.html>.

Pour plus d'informations sur le microprogramme du système pour les serveurs pris en charge, reportez-vous à la section "Patchs requis pour le microprogramme du système" dans les *Notes de version Oracle VM Server for SPARC 2.0*.

Pour mettre à niveau le microprogramme du système à partir du domaine de contrôle, reportez-vous aux notes de version du microprogramme du système.

Reportez-vous aux guides d'administration ou aux notes sur le produit pour les serveurs pris en charge pour plus d'informations sur l'installation et la mise à niveau du microprogramme du système pour ces serveurs.

Vous pouvez également utiliser l'interface Web d'ILOM pour mettre à niveau le système. Reportez-vous à la section "Mise à jour du microprogramme ILOM" dans le *Sun Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (<http://dlc.sun.com/pdf/820-6411-12/820-6411-12.pdf>).

1 Téléchargez l'image du microprogramme du système sur un autre système exécutant le service `tftp`.

a. Vérifiez que le service `tftp` est en ligne sur le serveur.

```
# svcs tftp/udp6
STATE      STIME      FMRI
onLine     Mar_26    svc:/network/tftp/udp6:default
```

b. Activez le service `tftp` s'il n'est pas dans l'état en ligne.

```
# svcadm enable tftp/udp6
```

c. Téléchargez l'image du microprogramme du système dans le répertoire `/tftpboot`.

2 Assurez-vous que le port de gestion réseau du processeur de service ILOM est configuré.

Cette configuration est nécessaire pour accéder à la nouvelle image flash via le réseau. Reportez-vous à la section "Configuration du port de gestion réseau du processeur de service" de l'*Sun SPARC Enterprise T5120 and T5220 Servers Topic Set* et à la section "Mise à jour du microprogramme ILOM" du *Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (<http://dlc.sun.com/pdf/820-6412-12/820-6412-12.pdf>).

3 Ouvrez une session SSH pour vous connecter au processeur de service.

```
$ ssh root@system-name
...
Are you sure you want to continue connecting (yes/no)? yes
...
Password: password
...
->
```

4 Vérifiez que l'hôte est hors tension.

a. Saisissez la commande suivante :

```
-> show /SYS power_state
```

b. Si l'hôte n'est pas sous tension, tapez la commande suivante :

```
-> stop /SYS
```

5 Vérifiez que le paramètre `keyswitch_state` est défini sur `normal`.

a. Saisissez la commande suivante :

```
-> show /SYS keyswitch_state
```

b. Si la valeur est différente de `normal`, définissez-la à l'aide de la commande suivante :

```
-> set /SYS keyswitch_state=normal
```

6 Mettez à niveau l'image flash du processeur de service et le microprogramme hôte.

```
-> load -source \
tftp://IP-addr/pathname/Sun_System_Firmware-x_x_x_build_nn-server-name.pkg
```

L'option `-source` spécifie l'adresse IP et le nom de chemin complet (URL) vers l'image flash du microprogramme du système.

- *IP-addr* est l'adresse IP d'un serveur tftp sur le réseau pouvant accéder à l'image flash.
- *pathname* est le nom de chemin complet vers l'image flash sur le serveur tftp.
- *x_x_x* est le numéro de version du microprogramme du système
- *nn* est le numéro de construction qui s'applique à cette version.
- *server-name* est le nom de votre serveur.

Par exemple, pour le serveur SPARC Enterprise T5440, *server-name* est `SPARC_Enterprise_T5440`.

Par exemple, l'option `-source`

```
tftp://192.168.1.1/Sun_System_Firmware-7_3_0-SPARC_Enterprise_T5440.pkg
```

pointe sur le fichier `/tftpboot/Sun_System_Firmware-7_3_0-SPARC_Enterprise_T5440.pkg` sur le serveur avec l'adresse IP `192.168.1.1`.

Une fois l'image flash mise à niveau, le système redémarre automatiquement.

Le processeur de service redémarre, exécute les diagnostics et revient à l'invite de connexion (sur la console de série).

Téléchargement du gestionnaire de domaines logiques

▼ Téléchargement du logiciel

1 Téléchargez le fichier zip (OVM_Server_SPARC-2_0.zip).

Le logiciel est disponible à l'adresse <http://www.oracle.com/virtualization>.

2 Décompressez le fichier zip.

```
$ unzip OVM_Server_SPARC-2_0.zip
```

Reportez-vous à la section “Emplacement du logiciel Oracle VM Server for SPARC 2.0” dans les *Notes de version Oracle VM Server for SPARC 2.0* pour plus d'informations sur la structure du fichier et son contenu.

Installation du gestionnaire de domaines logiques

Il y a trois méthodes d'installation du logiciel gestionnaire de domaines logiques :

- Utilisation du script d'installation pour installer les packages et les patches. Cette opération installe automatique le logiciel du gestionnaire de domaines logiques. Reportez-vous à la section “[Installation automatique du logiciel du gestionnaire de domaines logiques](#)” à la page 30.
- Utilisation de JumpStart pour installer les packages dans le cadre d'une installation en réseau d'Oracle Solaris. Reportez-vous à la section “[Utilisation de JumpStart pour installer le logiciel Oracle VM Server for SPARC 2.0](#)” à la page 31.
- Installation manuelle du package. Reportez-vous à la section “[Installation manuelle du logiciel du gestionnaire de domaines logiques](#)” à la page 32.

Remarque – Notez que vous devez installer manuellement le package logiciel MIB des domaines logiques après avoir installé les packages Oracle VM Server for SPARC. Il n'est pas installé automatiquement avec les autres packages. Reportez-vous au [Logical Domains \(LDoms\) MIB 1.0.1 Administration Guide](#) pour plus d'informations sur l'installation et l'utilisation de la MIB des &LDoms.

Installation automatique du logiciel du gestionnaire de domaines logiques

Si vous utilisez le script d'installation `install-ldm`, vous avez plusieurs choix pour indiquer comment vous souhaitez exécuter le script. Chaque choix est décrit dans les procédures suivantes.

- **L'utilisation du script `install-ldm` sans aucune option n'effectue pas les opérations suivantes automatiquement :**
 - Vérification que la version du SE Oracle Solaris est le SE Oracle Solaris 10 9/10 au minimum
 - Vérification que les sous-répertoires du package `SUNWldm/` et `SUNWldmp2v/` sont présents
 - Vérification que les packages du pilote des domaines logiques prérequis, `SUNWldomr` et `SUNWldomu`, sont présents
 - Vérification que les packages `SUNWldm` et `SUNWldmp2v` n'ont pas été installés
 - Installation du logiciel Oracle VM Server for SPARC 2.0
 - Vérification que tous les packages sont installés
 - Si le SST `SUNWjass` est déjà installé, vous êtes invité à renforcer le SE Oracle Solaris sur le domaine de contrôle.
 - Déterminer s'il faut utiliser l'assistant de configuration Oracle VM Server for SPARC (`ldmconfig`) pour effectuer l'installation.
- **L'utilisation du script `install-ldm` avec l'option `-c` exécute automatiquement l'assistant de configuration d'Oracle VM Server for SPARC après l'installation du logiciel.**
- **L'utilisation du script `install-ldm` avec l'option `-s` passe l'exécution de l'assistant de configuration Oracle VM Server for SPARC.**
- **L'utilisation du script `install-ldm` et les options suivantes avec les logiciel SST vous permettent d'effectuer les opérations suivantes :**
 - `install-ldm -d`. Vous permet d'indiquer un pilote SST différent d'un pilote se terminant par `-secure.driver`. Cette option effectue automatiquement toutes les fonctions répertoriées dans le choix précédent et renforce le SE Oracle Solaris sur le domaine de contrôle avec le pilote personnalisé SST que vous indiquez. Par exemple, `server-secure-myname.driver`.
 - `install-ldm -d none`. Indique que vous *ne* voulez pas renforcer le SE Oracle Solaris s'exécutant sur votre domaine de contrôle à l'aide de SST. Cette option effectue automatiquement toutes les fonctions sauf le renforcement répertorié dans les choix précédents. Ne pas utiliser le SST n'est pas recommandé et doit uniquement être effectué lorsque vous avez l'intention de renforcer votre domaine de contrôle à l'aide d'un autre processus.

- `install -ldm -p`. Indique que vous voulez uniquement effectuer des actions de post-installation pour activer le démon du gestionnaire de domaines logiques (`ldmd`) et exécuter le SST. Par exemple, vous devez utiliser cette option si les packages `SUNWldm` et `SUNWjass` sont préinstallés sur votre serveur.

Utilisation de JumpStart pour installer le logiciel Oracle VM Server for SPARC 2.0

Reportez-vous à *Technologie JumpStart : utilisation dans l'environnement d'exploitation Solaris* pour obtenir des informations complètes sur l'utilisation de JumpStart.

▼ Configuration d'un serveur JumpStart

Si vous n'avez pas encore configuré un serveur JumpStart, vous devez effectuer cette opération. Reportez-vous au *Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée* pour obtenir des informations complètes sur cette procédure.

1 Reportez-vous au *Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée*.

Effectuez les opérations suivantes :

- Reportez-vous à la section "Liste des tâches à effectuer lors de la préparation à une installation JumpStart personnalisée" du *Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée*.**
- Configurez les systèmes réseau avec les procédures de la section "Création d'un serveur de profils pour les systèmes réseau".**
- Créez un profil avec la procédure de la section "Création de profil", puis ajoutez une ligne au profil pour installer le package `SUNWldm.v` à l'aide du mot-clé de profil package.**

Par exemple, ajoutez la ligne suivante à votre profil pour installer le package `SUNWldm.v` du répertoire `extra` à partir du serveur HTTP `192.168.254.255`.

```
package          SUNWldm.v http://192.168.254.255/extra timeout 5
```

- Créez le fichier `rules` avec la procédure de la section "Création du fichier `rules`".**

2 Validez le fichier `rules` avec la procédure de la section "Validation du fichier `rules`".

Installation manuelle du logiciel du gestionnaire de domaines logiques

▼ Installation manuelle d'Oracle VM Server for SPARC 2.0

Avant de commencer

Téléchargez le logiciel Oracle VM Server for SPARC 2.0 (les packages `SUNWldm` et `SUNWldmp2v`). Consultez la section “[Téléchargement du logiciel](#)” à la page 29 pour obtenir des instructions détaillées.

- 1 Utilisez la commande `pkgadd` pour installer les packages `SUNWldm.v` et `SUNWldmp2v`.

Pour plus d'informations sur la commande `pkgadd`, consultez la page de manuel [pkgadd\(1M\)](#).

L'option `-G` installe le package dans la zone globale uniquement et l'option `-d` définit le chemin vers le répertoire contenant les packages `SUNWldm.v` et `SUNWldmp2v`.

```
# pkgadd -Gd . SUNWldm.v SUNWldmp2v
```

- 2 Répondez `y` pour oui à toutes les questions des invites interactives.

- 3 Utilisez la commande `pkginfo` pour vérifier que les packages Oracle VM Server for SPARC 2.0, `SUNWldm` et `SUNWldmp2v`, sont installés.

Pour plus d'informations sur la commande `pkginfo`, reportez-vous à la page de manuel [pkginfo\(1\)](#).

Les informations de révision (REV) présentées ci-dessous sont un exemple.

```
# pkginfo -l SUNWldm | grep VERSION
VERSION=2.0,REV=2010.08.03.10.20
```

Activation du démon du gestionnaire de domaines logiques

Le script d'installation `install_ldm` active automatiquement le démon du gestionnaire de domaines logiques (`ldmd`). Le démon `ldmd` est également automatiquement activé lorsque le package `SUNWldm` est installé. Une fois le démon activé, vous pouvez créer, modifier et contrôler les domaines logiques.

▼ Activation du démon du gestionnaire de domaines logiques

Utilisez cette procédure pour activer le démon `ldmd` s'il a été désactivé.

- 1 Utilisez la commande `svcadm` pour activer le démon du gestionnaire de domaines logiques, `ldmd`.

Pour plus d'informations sur la commande `svcadm`, reportez-vous à la page de manuel [svcadm\(1M\)](#).

```
# svcadm enable ldmd
```

- 2 **Utilisez la commande `ldm list` pour vérifier que le gestionnaire de domaines logiques est en cours d'exécution.**

La commande `ldm list` doit répertorier tous les domaines actuellement définis sur le système. Le domaine `primary` en particulier doit être répertorié et être à l'état `active`. L'exemple de sortie suivant indique que seul le domaine `primary` est défini sur le système.

```
# /opt/SUNWldm/bin/ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active ---c-  SP    64    3264M  0.3%  19d 9m
```

Mise à niveau d'un système utilisant déjà Oracle VM Server for SPARC

Cette section décrit le processus de mise à niveau du microprogramme du SE Oracle Solaris et les composants du gestionnaire de domaines logiques sur un système utilisant déjà le logiciel Oracle VM Server for SPARC.

Si votre système est déjà configuré avec le logiciel Oracle VM Server for SPARC, le domaine de contrôle doit par conséquent être mis à niveau. Les autres domaines existants doivent également être mis à niveau si vous souhaitez pouvoir utiliser toutes les fonctions du logiciel Oracle VM Server for SPARC 2.0.

Mise à niveau du SE Oracle Solaris

Reportez-vous à la section “Logiciels et patches requis” des *Notes de version Oracle VM Server for SPARC 2.0* pour trouver le SE Oracle Solaris 10 que vous devez utiliser pour cette version du logiciel Oracle VM Server for SPARC et les patches requis et recommandés pour les différents domaines. Reportez-vous au guide d'installation d'Oracle Solaris 10 pour obtenir des instructions complètes sur la mise à niveau du SE Oracle Solaris.

Lors de la réinstallation du SE Oracle Solaris dans le domaine de contrôle, vous devez enregistrer et restaurer les données de configuration enregistrées automatiquement des domaines logiques et le fichier de la base de données de contraintes, comme décrit dans cette section.

Sauvegarde et restauration des répertoires de configuration enregistrés automatiquement

À partir de la version 1.2 des domaines logiques, vous pouvez sauvegarder et restaurer les répertoires de configuration enregistrés automatiquement avant de réinstaller le système d'exploitation sur le domaine de contrôle. À chaque fois que vous réinstallez le système d'exploitation sur le domaine de contrôle, vous devez sauvegarder et restaurer les données de configuration enregistrées automatiquement des domaines logiques, qui se trouvent dans les répertoires `/var/opt/SUNWldm/autosave-autosave-name`.

Vous pouvez utiliser la commande `tar` ou `cpio` pour sauvegarder et restaurer tout le contenu des répertoires.

Remarque – Chaque répertoire enregistré automatiquement comprend un horodatage pour la dernière mise à jour de la configuration du SP pour la configuration concernée. Si vous restaurez les fichiers enregistrés automatiquement, l'horodatage risque de ne plus être synchronisé. Dans ce cas, les configurations enregistrées automatiquement restaurées sont affichées dans leur état précédent, [newer] ou à jour.

Pour plus d'informations sur les configurations enregistrées automatiquement, reportez-vous à la section “[Gestion des configurations des domaines logiques](#)” à la page 194.

▼ Sauvegarde et restauration des répertoires enregistrés automatiquement

Cette procédure montre comment sauvegarder et restaurer les répertoires enregistrés automatiquement.

1 Sauvegardez les répertoires enregistrés automatiquement.

```
# cd /  
# tar -cvpf autosave.tar var/opt/SUNWldm/autosave-*
```

2 (Facultatif) Supprimez les répertoires enregistrés automatiquement pour garantir une opération de restauration propre.

Parfois, un répertoire enregistré automatiquement peut comprendre des fichiers inutiles, peut-être laissés par une configuration précédente, risquant de corrompre la configuration qui a été téléchargée sur le SP. Dans ce cas, nettoyez le répertoire enregistré automatiquement avant l'opération de restauration comme indiqué dans cet exemple :

```
# cd /  
# rm -rf var/opt/SUNWldm/autosave-*
```

3 Restaurez les répertoires enregistrés automatiquement.

Ces commandes restaurent les fichiers et les répertoires du répertoire `/var/opt/SUNWldm`.

```
# cd /  
# tar -xvpf autosave.tar
```

Sauvegarde et restauration du fichier de la base de données de contraintes des domaines logiques

À chaque fois que vous mettez à niveau le système d'exploitation sur le domaine de contrôle, vous devez sauvegarder et restaurer le fichier de la base de données de contraintes des domaines logiques disponible dans `/var/opt/SUNWldm/ldom-db.xml`.

Remarque – Sauvegardez et restaurez également le fichier `/var/opt/SUNWldm/ldom-db.xml` lorsque vous effectuez une autre opération qui est destructive pour les données du fichier du domaine de contrôle, notamment un swap de disque.

Conservation du fichier de la base de données de contraintes des domaines logiques lors de l'utilisation de Live Upgrade

Si vous utilisez Live Upgrade sur le domaine de contrôle, envisagez d'ajouter la ligne suivante au fichier `/etc/lu/synclist` :

```
/var/opt/SUNWldm/ldom-db.xml OVERWRITE
```

Cette opération copie automatiquement la base de données de l'environnement de démarrage actif vers le nouvel environnement de démarrage lors de la commutation des environnements de démarrage. Pour plus d'informations sur `/etc/lu/synclist` et la synchronisation des fichiers entre les environnements de démarrage, reportez-vous à la section [“Synchronisation de fichiers entre les environnements d'initialisation”](#) du *Guide d'installation Oracle Solaris 10 9/10 : planification des mises à niveau et de Solaris Live Upgrade*.

Mise à niveau à partir d'un SE Oracle Solaris 10 antérieur au SE Oracle Solaris 10 5/08

Si le domaine de contrôle est mis à niveau à partir d'une version du SE Oracle Solaris 10 antérieure au SE Oracle Solaris 10 5/08 (ou sans le patch 127127-11), et si les volumes du gestionnaire de volumes ont été exportés en tant que disques virtuels, les moteurs de traitement des disques virtuels doivent être ré-exportés avec `options=slice` après la mise à niveau du gestionnaire de domaines logiques. Reportez-vous à la section [“Exportation de volumes et rétrocompatibilité”](#) à la page 95 pour plus d'informations.

Mise à niveau du gestionnaire de domaines logiques et du microprogramme du système

Cette section décrit comment mettre à niveau le logiciel Oracle VM Server for SPARC 2.0.

Téléchargez d'abord le gestionnaire de domaines logiques sur le domaine de contrôle. Voir [“Téléchargement du gestionnaire de domaines logiques”](#) à la page 29.

Arrêtez ensuite tous les domaines (sauf le domaine de contrôle) s'exécutant sur la plate-forme :

▼ Arrêt de tous les domaines s'exécutant sur la plate-forme, sauf le domaine de contrôle

- 1 Affichez chaque domaine à l'invite ok.
- 2 Arrêtez tous les domaines à l'aide de l'option `-a`.

```
primary# ldm stop-domain -a
```
- 3 Émettez la sous-commande `unbind-domain` à partir du domaine de contrôle pour chaque domaine.

```
primary# ldm unbind-domain ldom
```

Mise à niveau du logiciel Oracle VM Server for SPARC 2.0

Cette section décrit comment mettre à niveau le logiciel Oracle VM Server for SPARC 2.0.

Effectuez la procédure “Mise à niveau à partir du logiciel 1.0 des domaines logiques uniquement” des *Notes de version Oracle VM Server for SPARC 2.0*) si vous souhaitez utiliser vos configurations des domaines logiques 1.0 existants avec le logiciel Oracle VM Server for SPARC 2.0. Les configurations des domaines logiques 1.0 existants ne fonctionnent *pas* avec le logiciel Oracle VM Server for SPARC 2.0.

Si vous effectuez une mise à niveau à partir de versions plus récentes du logiciel Oracle VM Server for SPARC, suivez la procédure “[Mise à niveau du logiciel Oracle VM Server for SPARC 2.0](#)” à la page 36. De telles configurations de domaines logiques existants ne fonctionnent *pas* avec le logiciel Oracle VM Server for SPARC 2.0.

▼ Mise à niveau du logiciel Oracle VM Server for SPARC 2.0

- 1 Effectuez une mise à jour flash du microprogramme du système.
Pour obtenir toute la procédure, reportez-vous à la section “[Mise à niveau du microprogramme du système](#)” à la page 27.
- 2 Désactivez le démon du gestionnaire de domaines logiques (`ldmd`).

```
# svcadm disable ldmd
```
- 3 Supprimez le package `SUNWldm`.

```
# pkgrm SUNWldm
```

4 Ajoutez le nouveau package SUNWldm.

L'utilisation de l'option `-d` suppose que le package se trouve dans le répertoire courant.

```
# pkgadd -Gd . SUNWldm
```

5 Utilisez la commande `ldm list` pour vérifier que le gestionnaire de domaines logiques est en cours d'exécution.

La commande `ldm list` doit répertorier tous les domaines actuellement définis sur le système. Le domaine `primary` en particulier doit être répertorié et être à l'état `active`. L'exemple de sortie suivant indique que seul le domaine `primary` est défini sur le système.

```
# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active ---c-  SP    32    3264M  0.3%  19d 9m
```

Configuration usine par défaut et désactivation des domaines logiques

La configuration initiale sur laquelle la plate-forme apparaît en tant que système unique hébergeant uniquement un système d'exploitation est appelée configuration usine par défaut. Si vous souhaitez désactiver les domaines logiques, vous voudrez probablement également restaurer cette configuration afin que le système récupère l'accès à toutes les ressources (CPU, mémoire, E/S), pouvant avoir été assignées à d'autres domaines.

Cette section décrit la procédure pour supprimer tous les domaines invités, supprimer toutes les configurations de domaines logiques et remettre la configuration sur les valeurs par défaut d'usine.

▼ Suppression de tous les domaines invités

1 Arrêtez tous les domaines à l'aide de l'option `-a`.

```
primary# ldm stop-domain -a
```

2 Dissociez tous les domaines sauf le domaine `primary`.

```
primary# ldm unbind-domain ldom
```

Remarque – Vous risquez de ne pas pouvoir dissocier un domaine d'E/S s'il fournit des services requis par le domaine de contrôle. Dans ce cas, passez cette étape.

3 Détruisez tous les domaines sauf le domaine `primary`.

```
primary# ldm remove-domain -a
```

▼ Suppression de toutes les configurations de domaines logiques

- 1 Répertoriez toutes les configurations de domaines logiques stockées sur le processeur de service (SP).

```
primary# ldm list-config
```

- 2 Supprimez toutes les configurations (*config-name*) enregistrées précédemment sur le SP sauf la configuration `factory-default`.

Utilisez la commande suivante pour chacune de ces configurations :

```
primary# ldm rm-config config-name
```

Après avoir supprimé toutes les configurations précédemment enregistrées sur le SP, le domaine `factory-default` est le domaine suivant à utiliser lorsque le domaine de contrôle (`primary`) est réinitialisé.

▼ Restauration de la configuration usine par défaut

- 1 Sélectionnez la configuration usine par défaut.

```
primary# ldm set-config factory-default
```

- 2 Arrêtez le domaine de contrôle.

```
primary# shutdown -i1 -g0 -y
```

- 3 Remettez le système sous tension pour charger la configuration usine par défaut.

```
-> stop /SYS  
-> start /SYS
```

▼ Désactivation du gestionnaire de domaines logiques.

- Désactivez le gestionnaire de domaines logiques à partir du domaine de contrôle.

```
primary# svcadm disable ldmd
```

Remarque – La désactivation du gestionnaire de domaines logiques n'arrête pas les domaines en cours d'exécution, mais désactive la capacité à créer de nouveaux domaines, à modifier la configuration des domaines existants ou à surveiller l'état des domaines.



Attention – Si vous désactivez le gestionnaire de domaines logiques, vous désactivez certains services, notamment le compte-rendu des erreurs ou la gestion de l'alimentation. Dans le cas du compte-rendu d'erreurs, si vous êtes dans la configuration `factory-default`, vous pouvez redémarrer le domaine de contrôle pour restaurer le compte-rendu d'erreurs. Cependant, ce n'est pas le cas avec la gestion de l'alimentation. Par ailleurs, certains outils de gestion ou de surveillance du système reposent sur le gestionnaire de domaines logiques.

▼ Suppression du gestionnaire de domaines logiques

Après la restauration de la configuration usine par défaut et la désactivation du gestionnaire de domaines logiques, vous pouvez supprimer le logiciel du gestionnaire de domaines logiques.

- **Supprimez le logiciel du gestionnaire de domaines logiques.**

```
primary# pkgrm SUNWldm SUNWldmp2v
```

Remarque – Si vous supprimez le gestionnaire de domaines logiques avant de restaurer la configuration usine par défaut, vous pouvez restaurer celle-ci à partir du processeur de service comme indiqué dans la procédure suivante.

▼ Restauration de la configuration usine par défaut à partir du processeur de service

Si vous supprimez le gestionnaire de domaines logiques avant de restaurer la configuration usine par défaut, vous pouvez restaurer celle-ci à partir du processeur de service.

- 1 **Restaurer la configuration usine par défaut à partir du processeur de service.**
-> `set /HOST/bootmode config=factory-default`
- 2 **Remettez le système sous tension pour charger la configuration usine par défaut.**
-> `reset /SYS`

Sécurité

Ce chapitre décrit certaines fonctions de sécurité que vous pouvez activer sur le système de vos domaines logiques.

Ce chapitre aborde les sujets suivants :

- “Autorisation du gestionnaire de domaines logiques” à la page 41
- “Création d'une autorisation et de profils et assignation de rôles pour les comptes utilisateur” à la page 42
- “Configuration de RBAC pour l'accès à la console invité” à la page 45
- “Activation et utilisation du contrôle BSM” à la page 46

Autorisation du gestionnaire de domaines logiques

L'autorisation du gestionnaire de domaines logiques a deux niveaux :

- Lecture – permet de consulter, mais pas de modifier la configuration.
- Lecture et écriture – permet de consulter et de modifier la configuration.

Les modifications ne sont pas effectuées sur le SE Oracle Solaris, mais sont ajoutées au fichier d'autorisation par le script du package `postinstall` lorsque le gestionnaire de domaines logiques est installé. De même, les entrées d'autorisation sont supprimées par le script du package `preremove`.

Le tableau suivant répertorie les sous-commandes `ldm` avec l'autorisation utilisateur requise pour utiliser les commandes.

TABLEAU 3-1 Sous-commandes `ldm` et autorisations utilisateur

Sous-commande <code>ldm</code> ¹	Autorisation utilisateur
<code>add -*</code>	<code>solaris.ldoms.write</code>

¹ Concerne toutes les ressources que vous pouvez ajouter, répertorier, supprimer ou configurer.

TABLEAU 3-1 Sous-commandes ldm et autorisations utilisateur (Suite)

Sous-commande ldm ¹	Autorisation utilisateur
bind-domain	solaris.ldoms.write
list	solaris.ldoms.read
list-*	solaris.ldoms.read
panic-domain	solaris.ldoms.write
remove-*	solaris.ldoms.write
set-*	solaris.ldoms.write
start-domain	solaris.ldoms.write
stop-domain	solaris.ldoms.write
unbind-domain	solaris.ldoms.write

¹ Concerne toutes les ressources que vous pouvez ajouter, répertorier, supprimer ou configurer.

Création d'une autorisation et de profils et assignation de rôles pour les comptes utilisateur

Vous configurez une autorisation et des profils et assignez des rôles aux comptes utilisateur à l'aide du RBAC (contrôle d'accès basé sur les rôles) adapté pour le gestionnaire de domaines logiques. Reportez-vous à la page [Ensemble de documents administrateur du système Solaris 10](http://docs.sun.com/app/docs/coll/47.16) (<http://docs.sun.com/app/docs/coll/47.16>) pour plus d'informations sur le RBAC.

L'autorisation du gestionnaire de domaines logiques a deux niveaux :

- Lecture – vous permet de consulter, mais pas de modifier la configuration.
- Lecture et écriture – vous permet de consulter et de modifier la configuration.

Vous trouverez ci-dessous les entrées des domaines logiques ajoutées automatiquement au fichier `/etc/security/auth_attr` du SE Oracle Solaris :

- `solaris.ldoms:::LDom administration::`
- `solaris.ldoms.grant:::Delegate LDom configuration::`
- `solaris.ldoms.read:::View LDom configuration::`
- `solaris.ldoms.write:::Manage LDom configuration::`

Gestion des autorisations utilisateur

▼ Ajout d'une autorisation pour un utilisateur

Utilisez la procédure suivante pour ajouter les autorisations dans le fichier `/etc/security/auth_attr` pour les utilisateurs du gestionnaire de domaines logiques. Comme le superutilisateur a déjà l'autorisation `solaris.*`, ce dernier a déjà les droits pour les autorisations `solaris.ldoms.*`.

- 1 Créez un compte utilisateur local pour chaque utilisateur qui a besoin d'une autorisation pour utiliser les sous-commandes `ldm(1M)`.

Remarque – Pour ajouter l'autorisation du gestionnaire de domaines logiques pour un utilisateur, un compte local (non LDAP) doit être créé pour cet utilisateur. Reportez-vous à l'Ensemble de documents administrateur du système du SE Oracle Solaris (<http://docs.sun.com/app/docs/coll/47.16>) pour plus d'informations.

- 2 Effectuez l'une des opérations suivantes en fonction de la sous-commande `ldm(1M)` à laquelle vous souhaitez que l'utilisateur puisse accéder.

Voir le [Tableau 3-1](#) pour obtenir une liste des commandes `ldm(1M)` et de leurs autorisations utilisateur.

- Ajoutez une autorisation en lecture seule pour un utilisateur à l'aide de la commande `usermod(1M)`.

```
# usermod -A solaris.ldoms.read username
```

- Ajoutez une autorisation en lecture et en écriture pour un utilisateur à l'aide de la commande `usermod(1M)`.

```
# usermod -A solaris.ldoms.write username
```

▼ Suppression de toutes les autorisations d'un utilisateur

- Supprimez toutes les autorisations pour un compte utilisateur local (seule option possible).

```
# usermod -A '' username
```

Gestion des profils utilisateur

Le package SUNWldm ajoute deux profils RBAC définis par le système dans le fichier `/etc/security/prof_attr` à utiliser pour autoriser l'accès au gestionnaire de domaines logiques par les utilisateurs autres que les superutilisateurs. Les deux profils spécifiques aux domaines logiques sont :

- LDoms Review::`Review LDoms configuration:auths=solaris.ldoms.read`
- LDoms Management::`Manage LDoms domains:auths=solaris.ldoms.*`

Le package SUNWldm définit également l'attribut d'exécution suivant qui est associé au profil de gestion LDoms :

```
LDoms Management:suser:cmd:::/usr/sbin/ldm:privs=file_dac_read,file_dac_search
```

L'un des profils précédents peut être assigné à un compte utilisateur à l'aide de la procédure suivante.

▼ Ajout d'un profil pour un utilisateur

Les utilisateurs qui ont été directement assignés au profil de gestion LDoms doivent invoquer un shell de profil pour exécuter la commande `ldm` avec les attributs de sécurité. Pour plus d'informations, reportez-vous à [l'ensemble de documents administrateur du système Oracle Solaris 10](#).

- Ajoutez un profil d'administration pour un compte utilisateur local. Par exemple, gestion LDoms.

```
# usermod -P "LDoms Management" username
```

▼ Suppression de tous les profils d'un utilisateur

- Supprimez tous les profils pour un compte utilisateur local (seule option possible).

```
# usermod -P '' username
```

Assignation de rôles aux utilisateurs

L'avantage de l'utilisation de cette procédure est que seul un utilisateur ayant été assigné à un rôle spécifique peut assumer ce rôle. En assumant un rôle, un mot de passe est requis si le rôle est associé à un mot de passe. Cela fournit deux couches de sécurité. Si aucun rôle n'a été assigné à un utilisateur, celui-ci ne peut pas assumer de rôle (en utilisant la commande `su nom-rôle`) même si l'utilisateur a le mot de passe correct.

▼ Création d'un rôle et assignation du rôle à un utilisateur

1 Créez un rôle.

```
# roleadd -P "LDoms Review" ldm_read
```

2 Assignez un mot de passe au rôle.

```
# passwd ldm_read
```

3 Assignez le rôle à un utilisateur.

Par exemple, `user_1`.

```
# useradd -R ldm_read user_1
```

4 Assignez un mot de passe à l'utilisateur (`user_1`).

```
# passwd user_1
```

5 Assignez l'accès uniquement au compte `user_1` pour qu'il devienne le compte `ldm_read`.

```
# su user_1
```

6 Tapez le mot de passe utilisateur lorsque vous y êtes invité.

7 Vérifiez l'ID utilisateur et l'accès au rôle `ldm_read`.

```
$ id
uid=nn(user_1) gid=nn(group-name)
$ roles
ldm_read
```

8 Fournissez l'accès à l'utilisateur pour les sous-commandes `ldm` ayant une autorisation en lecture.

```
# su ldm_read
```

9 Tapez le mot de passe utilisateur lorsque vous y êtes invité.

10 Tapez la commande `id` pour afficher l'utilisateur.

```
$ id
uid=nn(ldm_read) gid=nn(group-name)
```

Configuration de RBAC pour l'accès à la console invité

Le démon `vntsd` fournit une propriété nommée `vntsd/authorization`. Cette propriété peut être configurée pour activer le contrôle d'autorisation pour les utilisateurs et les rôles pour une console de domaine ou un groupe de consoles. Pour activer le contrôle d'autorisation, utilisez la commande `svccfg` pour définir la valeur de cette propriété sur `true`. Lorsque cette option est activée, `vntsd` écoute et accepte les connexions uniquement sur `localhost`. Si la propriété

`listen_addr` indique une autre adresse IP lorsque `vntsd/authorization` est activé, `vntsd` ignore l'autre adresse IP et continue à écouter uniquement sur `localhost`.

Par défaut, une autorisation à accéder à toutes les consoles invité est ajoutée à la base de données `auth_attr`, lorsque le service `vntsd` est activé.

```
solaris.vntsd.consoles::Access All LDoms Guest Consoles::
```

Le superutilisateur peut utiliser la commande `usermod` pour assigner les autorisations requises à d'autres utilisateurs ou rôles. Cette opération accorde des droits uniquement à l'utilisateur ou au rôle qui a les autorisations requises pour accéder à une console de domaine ou à un groupe de consoles spécifique.

L'exemple suivant donne à l'utilisateur `terry` l'autorisation d'accéder à toutes les consoles de domaine :

```
# usermod -A "solaris.vntsd.consoles" terry
```

L'exemple suivant ajoute une nouvelle autorisation pour une console de domaine spécifique avec le nom `ldg1` et assigne cette autorisation à un utilisateur `sam` :

1. Ajoutez la nouvelle entrée d'autorisation au fichier `auth_attr` pour le domaine `ldg1`.

```
solaris.vntsd.console-ldg1::Access Specific LDoms Guest Console::
```

2. Assignez cette autorisation à l'utilisateur `sam` :

```
# usermod -A "solaris.vntsd.console-ldg1" sam
```

Pour plus d'informations sur les autorisations et le RBAC, reportez-vous au [System Administration Guide: Security Services](#).

Activation et utilisation du contrôle BSM

Le gestionnaire de domaines logiques utilise la capacité de contrôle du module de sécurité de base (BSM) du SE Oracle Solaris. Le contrôle BSM fournit des moyens pour examiner l'historique des actions et des événements de votre domaine de contrôle afin de déterminer ce qui s'est produit. L'historique est conservé dans un journal de ce qui a été fait, quand cela a été fait, par qui et ce qui a été affecté.

Pour activer et désactiver la capacité de contrôle, utilisez les commandes du SE Oracle Solaris `bsmconv(1M)` et `bsmunconv(1M)`. Cette section comprend également des tâches qui indiquent comment vérifier la capacité de contrôle, imprimer la sortie d'audit et faire tourner les journaux d'audit. Vous trouverez de plus amples informations sur le contrôle BSM dans le *Guide d'administration du système : services de sécurité* de Solaris 10.

▼ Activation du contrôle BSM

- 1 Ajoutez `vs` dans la ligne `fFlags` : du fichier `/etc/security/audit_control`.
- 2 Exécutez la commande `bsmconv(1M)`.

```
# /etc/security/bsmconv
```

Pour plus d'informations sur cette commande, reportez-vous à la page de manuel [bsmconv\(1M\)](#).
- 3 Redémarrez le SE Oracle Solaris pour que le contrôle entre en vigueur.

▼ Vérification de l'activation du contrôle BSM

- 1 Saisissez la commande suivante.

```
# auditconfig -getcond
```
- 2 Vérifiez que `audit condition = auditing` apparaît dans la sortie.

▼ Désactivation du contrôle BSM

- 1 Exécutez la commande `bsmunconv` pour désactiver le contrôle BSM.

```
# /etc/security/bsmunconv
```

Pour plus d'informations sur cette commande, reportez-vous à la page de manuel [bsmunconv\(1M\)](#).
- 2 Redémarrez le SE Oracle Solaris pour que la désactivation du contrôle entre en vigueur.

▼ Impression de la sortie d'audit

- Utilisez l'une des méthodes suivantes pour imprimer la sortie d'audit BMS :
 - Utilisez les commandes [auditreduce\(1M\)](#) et [praudit\(1M\)](#) pour imprimer la sortie d'audit.

```
# auditreduce -c vs | praudit  
# auditreduce -c vs -a 20060502000000 | praudit
```
 - Utilisez la commande `praudit -x` pour imprimer la sortie XML.

▼ **Rotation des journaux d'audit**

- Utilisez la commande `audit -n` pour faire tourner les journaux d'audit.

Configuration des services et du domaine de contrôle

Ce chapitre décrit la procédure de configuration des services par défaut et de votre domaine de contrôle.

Vous pouvez également utiliser l'assistant de configuration d'Oracle VM Server for SPARC pour configurer les domaines logiques et les services. Reportez-vous à l'[Annexe B](#), “Assistant de configuration Oracle VM Server for SPARC”.

Ce chapitre aborde les sujets suivants :

- “Messages de sortie” à la page 49
- “Création des services par défaut” à la page 50
- “Configuration initiale du domaine de contrôle” à la page 51
- “Redémarrage pour utiliser domaines logiques” à la page 53
- “Activation de la mise en réseau entre le domaine de contrôle/service et les autres domaines” à la page 53
- “Activation du démon du serveur de terminal du réseau virtuel” à la page 54

Messages de sortie

À partir de la version 2.0 d'Oracle VM Server for SPARC, si une ressource ne peut pas être configurée de manière dynamique sur le domaine de contrôle, il convient de lancer d'abord une reconfiguration retardée. La reconfiguration retardée reporte les activités de configuration à après le redémarrage du domaine de contrôle.

Vous recevez le message suivant lorsque vous démarrez une reconfiguration retardée sur le domaine primary :

```
Initiating a delayed reconfiguration operation on the primary domain.  
All configuration changes for other domains are disabled until the  
primary domain reboots, at which time the new configuration for the  
primary domain also takes effect.
```

Vous recevez l'avertissement suivant après chaque opération suivante sur le domaine `primary` jusqu'au redémarrage :

Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.

Création des services par défaut

Les services de périphérique virtuel suivants doivent être créés pour utiliser le domaine de contrôle comme un domaine de service et pour créer des périphériques virtuels pour les autres domaines.

- `vcc` – Service de concentrateur de console virtuelle
- `vds` – Serveur de disque virtuel
- `vsw` – Service de commutateur virtuel

▼ Création des services par défaut

- 1 **Créez un service de concentrateur de console virtuelle (`vcc`) à utiliser par le démon du serveur de terminal de réseau virtuel (`vntsd`) et en tant que concentrateur pour toutes les consoles de domaine logique.**

Par exemple, la commande suivante ajouterait un service de concentrateur de console virtuelle (`primary-vcc0`) avec une plage de port allant de 5000 à 5100 au domaine de contrôle (`primary`).

```
primary# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

- 2 **Créez un serveur de disque virtuel (`vds`) pour permettre l'importation de disques virtuels dans un domaine logique.**

Par exemple, la commande suivante ajoute un serveur de disque virtuel (`primary-vds0`) au domaine de contrôle (`primary`).

```
primary# ldm add-vds primary-vds0 primary
```

- 3 **Créez un service de commutateur virtuel (`vsw`) pour permettre la mise en réseau des périphériques du réseau virtuel (`vnet`) dans les domaines logiques.**

Assignez un adaptateur réseau compatible GLDv3 au commutateur virtuel si chacun des domaines logiques doit communiquer différemment par l'intermédiaire du commutateur virtuel.

Par exemple, la commande suivante ajoute un service de commutateur virtuel (`primary-vsw0`) sur le pilote d'adaptateur réseau (`nxge0`) vers le domaine de contrôle (`primary`).

```
primary# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

Cette commande alloue automatiquement une adresse MAC au commutateur virtuel. Vous pouvez définir votre propre adresse MAC sous forme d'option à la commande `ldm add -vsw`).

Cependant, dans ce cas, il est de votre responsabilité de vous assurer que l'adresse MAC indiquée ne soit pas en conflit avec une adresse MAC déjà existante.

Si le commutateur virtuel ajouté remplace l'adaptateur physique sous-jacent en tant qu'interface réseau principale, il doit être associé à l'adresse MAC de l'adaptateur physique, afin que le serveur DHCP (Dynamic Host Configuration Protocol) assigne au domaine la même adresse IP. Reportez-vous à la section [“Activation de la mise en réseau entre le domaine de contrôle/service et les autres domaines”](#) à la page 53.

```
primary# ldm add-vsw mac-addr=2:04:4f:fb:9f:0d net-dev=nxge0 primary-vsw0 primary
```

4 Vérifiez que les services ont été créés à l'aide de la sous-commande `list-services`.

Votre sortie doit ressembler à ce qui suit.

```
primary# ldm list-services primary
VDS
  NAME                VOLUME          OPTIONS          DEVICE
  primary-vds0

VCC
  NAME                PORT-RANGE
  primary-vcc0        5000-5100

VSW
  NAME                MAC              NET-DEV          DEVICE          MODE
  primary-vsw0        02:04:4f:fb:9f:0d nxge0            switch@0        prog,promisc
```

Configuration initiale du domaine de contrôle

Initialement, toutes les ressources du système sont allouées au domaine de contrôle. Pour permettre la création d'autres domaines logiques, vous devez libérer certaines de ces ressources.

N'essayez *pas* d'utiliser la reconfiguration dynamique (DR) de mémoire pour effectuer la configuration initiale du domaine de contrôle. Bien que vous puissiez utiliser la reconfiguration dynamique de mémoire pour effectuer cette configuration sans redémarrage, cela n'est *pas* recommandé. L'approche de reconfiguration dynamique de mémoire peut prendre un certain temps (plus long qu'un redémarrage) et peut éventuellement échouer. Utilisez plutôt la commande `ldm start -reconf` pour faire passer le domaine de contrôle en mode de reconfiguration retardée avant de modifier la configuration de la mémoire. Vous pouvez ensuite redémarrer le domaine de contrôle après avoir terminé toutes les étapes de configuration.

▼ Configuration du domaine de contrôle

Remarque – Cette procédure contient des exemples de ressources pour configurer votre domaine de contrôle. Ces nombres sont des exemples uniquement et les valeurs utilisées peuvent ne pas être adaptées à votre domaine de contrôle.

1 Déterminez si vous avez des périphériques cryptographiques dans le domaine de contrôle.

```
primary# ldm list -o crypto primary
```

2 Assignez des ressources cryptographiques au domaine de contrôle.

L'exemple suivant assigne une ressource cryptographique au domaine de contrôle, `primary`. Cette opération laisse le reste des ressources cryptographiques disponibles pour un domaine invité.

```
primary# ldm set-mau 1 primary
```

3 Assignez des CPU virtuels au domaine de contrôle.

Par exemple, la commande suivante assigne 8 CPU virtuels au domaine de contrôle `primary`. Cette opération laisse le reste des CPU virtuels disponibles pour un domaine invité.

```
primary# ldm set-vcpu 8 primary
```

4 Démarrez une reconfiguration retardée sur le domaine de contrôle.

```
primary# ldm start-reconf primary
```

5 Assignez de la mémoire au domaine de contrôle.

Par exemple, la commande suivante assigne 4 giga-octets de mémoire au domaine de contrôle `primary`. Cette opération laisse le reste de la mémoire disponible pour un domaine invité.

```
primary# ldm set-memory 4G primary
```

6 Ajoutez une configuration de machine de domaine logique au processeur de service (SP).

Par exemple, la commande suivante ajoute une configuration appelée `initial`.

```
primary# ldm add-config initial
```

7 Vérifiez que la configuration est prête à être utilisée au prochain redémarrage.

```
primary# ldm list-config
factory-default
initial [next poweron]
```

Cette sous-commande de liste affiche la configuration `initial` définie qui sera utilisée à la prochaine remise sous tension.

Redémarrage pour utiliser domaines logiques

Vous devez redémarrer le domaine de contrôle pour que les modifications de configuration soient appliquées et que les ressources soient libérées pour une utilisation par les autres domaines logiques.

▼ Redémarrage

- Arrêtez et redémarrez le domaine de contrôle.

```
primary# shutdown -y -g0 -i6
```

Remarque – Une réinitialisation ou une remise sous tension instancie la nouvelle configuration. Seule une remise sous tension initialise réellement la configuration enregistrée sur le processeur de service (SP), ce qui est ensuite répercuté dans la sortie `list-config`.

Activation de la mise en réseau entre le domaine de contrôle/service et les autres domaines

Par défaut, la mise en réseau entre le domaine de contrôle et les autres domaines du système est désactivée. Pour l'activer, le périphérique de commutateur virtuel doit être configuré en tant que périphérique réseau. Le commutateur virtuel peut remplacer le périphérique sous-jacent (`nxge0` dans cet exemple) en tant qu'interface principale ou être configuré en tant qu'interface réseau supplémentaire dans le domaine.

Remarque – Procédez comme suit à partir de la console du domaine de contrôle, car cette procédure peut interrompre momentanément la connectivité au domaine.

▼ Configuration du commutateur virtuel en tant qu'interface principale

- 1 Imprimez les informations d'adressage pour toutes les interfaces.

```
primary# ifconfig -a
```

- 2 Montez le commutateur virtuel. Dans cet exemple, `vsw0` est le commutateur virtuel étant configuré.

```
primary# ifconfig vsw0 plumb
```

- 3 (Facultatif) Pour obtenir la liste des instances de commutateurs virtuels dans un domaine, vous pouvez les répertorier.**

```
primary# /usr/sbin/dladm show-link | grep vsw
vsw0          type: non-vlan mtu: 1500      device: vsw0
```

- 4 Démontez le périphérique réseau physique assigné au commutateur virtuel (net - dev), qui est nxge0 dans cet exemple.**

```
primary# ifconfig nxge0 down unplumb
```

- 5 Pour faire migrer les propriétés du périphérique réseau physique (nxge0) vers le périphérique du commutateur (vsw0), effectuez l'une des opérations suivantes :**

- Si la mise en réseau est configurée avec une adresse IP statique, réutilisez l'adresse IP et le masque réseau de nxge0 pour vsw0.

```
primary# ifconfig vsw0 IP_of_nxge0 netmask netmask_of_nxge0 broadcast + up
```

- Si la mise en réseau est configurée à l'aide du DHCP, activez le DHCP pour vsw0.

```
primary# ifconfig vsw0 dhcp start
```

- 6 Apportez les modifications requises au fichier de configuration pour rendre ce changement permanent.**

```
primary# mv /etc/hostname.nxge0 /etc/hostname.vsw0
primary# mv /etc/dhcp.nxge0 /etc/dhcp.vsw0
```

Remarque – Si nécessaire, vous pouvez également configurer le commutateur virtuel ainsi que le périphérique réseau physique. Dans ce cas, montez le commutateur virtuel comme à l'étape 2, et ne démontez pas le périphérique physique (passez l'étape 4). Vous devez ensuite configurer le commutateur virtuel avec une adresse IP statique ou une adresse IP dynamique. Vous pouvez obtenir une adresse IP dynamique à partir d'un serveur DHCP. Pour plus d'informations et pour obtenir un exemple de ce cas, reportez-vous à la section [“Configuration du commutateur virtuel et du domaine de service pour le NAT et le routage”](#) à la page 126

Activation du démon du serveur de terminal du réseau virtuel

Vous devez activer le démon du serveur de terminal du réseau virtuel (vntsd) pour fournir l'accès à la console virtuelle de chaque domaine logique. Reportez-vous à la page de manuel [vntsd\(1M\)](#) pour plus d'informations sur l'utilisation de ce démon.

▼ Activation du démon du serveur de terminal du réseau virtuel

Remarque – Assurez-vous que vous avez créé le service par défaut `vconscon` (`vcc`) sur le domaine de contrôle avant d'activer `vntsd`. Voir [“Création des services par défaut”](#) à la page 50 pour plus d'informations.

- 1 Utilisez la commande `svcadm(1M)` pour activer le démon du serveur de terminal du réseau virtuel, `vntsd(1M)`.

```
primary# svcadm enable vntsd
```

- 2 Utilisez la commande `svcs(1)` pour vérifier que le démon `vntsd` est désactivé.

```
primary# svcs vntsd
STATE          STIME          FMRI
online         Oct_08         svc:/ldoms/vntsd:default
```


Configuration des domaines invités

Ce chapitre décrit les procédures nécessaires à la configuration des domaines invités.

Vous pouvez également utiliser l'assistant de configuration d'Oracle VM Server for SPARC pour configurer les domaines logiques et les services. Reportez-vous à l'[Annexe B](#), “Assistant de configuration Oracle VM Server for SPARC”.

Ce chapitre aborde les sujets suivants :

- “Création et démarrage d'un domaine invité” à la page 57
- “Installation du SE Oracle Solaris sur un domaine invité” à la page 60

Création et démarrage d'un domaine invité

Le domaine invité doit exécuter le système d'exploitation qui comprend la plate-forme sun4v et les périphériques virtuels présentés par l'hyperviseur. Actuellement, cela signifie que vous devez exécuter au moins le SE Oracle Solaris 10 11/06. L'exécution du SE Oracle Solaris 10 9/10 vous fournit toutes les fonctions d'Oracle VM Server for SPARC 2.0. Reportez-vous aux *Notes de version Oracle VM Server for SPARC 2.0* pour obtenir les patches pouvant être nécessaires. Une fois les services par défaut créés et les ressources réallouées à partir du domaine de contrôle, vous pouvez créer et démarrer un domaine invité.

▼ Création et démarrage d'un domaine invité

1 Créez un domaine logique.

Par exemple, la commande suivante crée un domaine invité nommé `ldg1`.

```
primary# ldm add-domain ldg1
```

2 Ajoutez des CPU au domaine invité.

Par exemple, la commande suivante ajoute huit CPU virtuels au domaine invité `ldg1`.

```
primary# ldm add-vcpu 8 ldg1
```

3 Ajoutez de la mémoire au domaine invité.

Par exemple, la commande suivante ajoute 2 giga-octets de mémoire au domaine invité `ldg1`.

```
primary# ldm add-memory 2G ldg1
```

4 Ajoutez un périphérique de réseau virtuel au domaine invité.

Par exemple, la commande suivante ajoute un périphérique de réseau virtuel avec ces paramètres au domaine invité `ldg1`.

```
primary# ldm add-vnet vnet1 primary-vsw0 ldg1
```

où :

- `vnet1` est un nom d'interface unique vers le domaine logique, assigné à cette instance de périphérique de réseau virtuel pour référence sur les sous-commandes ultérieures `set -vnet` ou `remove -vnet`.
- `primary-vsw0` est le nom du service réseau existant (commutateur virtuel) auquel se connecter.

Remarque – Les étapes 5 et 6 sont des instructions simplifiées pour ajouter un périphérique de serveur de disque virtuel (`vdsdev`) au domaine principal et un disque virtuel (`vdisk`) au domaine invité. Pour en savoir plus sur l'utilisation des volumes et des systèmes de fichiers ZFS en tant que disques virtuels, reportez-vous aux sections [“Exportation d'un volume ZFS en tant que disque à segment unique”](#) à la page 94 et [“Utilisation de ZFS avec les disques virtuels”](#) à la page 104.

5 Indiquez le périphérique que le serveur de disque virtuel doit exporter en tant que disque virtuel sur le domaine invité.

Vous pouvez exporter un disque physique, un segment de disque ou un fichier en tant que périphérique en mode bloc. Les exemples suivants présentent un disque physique et un fichier.

- **Exemple de disque physique.** Le premier exemple ajoute un disque physique avec ces paramètres.

```
primary# ldm add-vdsdev /dev/dsk/c2t1d0s2 vol1@primary-vds0
```

où :

- `/dev/dsk/c2td0s2` est le nom du chemin du périphérique physique réel. Lors de l'ajout d'un périphérique, le nom de chemin doit être associé au nom du périphérique.
- `vol1` est un nom unique que vous définissez pour le périphérique ajouté au serveur de disque virtuel. Le nom de volume doit être unique pour cette instance de serveur de disque virtuel, car ce nom est exporté par le serveur de disque virtuel sur les clients pour l'ajout. Lors de l'ajout d'un périphérique, le nom de volume doit être associé au nom du chemin du périphérique réel.
- `primary-vds0` est le nom du serveur de disque virtuel auquel ajouter ce périphérique.
- **Exemple de fichier.** Le second exemple est l'exportation d'un fichier en tant que périphérique en mode bloc.

```
primary# ldm add-vdsdev backend vol1@primary-vds0
```

où :

- `backend` est le nom du chemin du fichier réel exporté en tant que périphérique en mode bloc. Lors de l'ajout d'un périphérique, le moteur de traitement doit être associé au nom du périphérique.
- `vol1` est un nom unique que vous définissez pour le périphérique ajouté au serveur de disque virtuel. Le nom de volume doit être unique pour cette instance de serveur de disque virtuel, car ce nom est exporté par le serveur de disque virtuel sur les clients pour l'ajout. Lors de l'ajout d'un périphérique, le nom de volume doit être associé au nom du chemin du périphérique réel.
- `primary-vds0` est le nom du serveur de disque virtuel auquel ajouter ce périphérique.

6 Ajoutez un disque virtuel au domaine invité.

L'exemple suivant ajout un disque virtuel au domaine invité `ldg1`.

```
primary# ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1
```

où :

- `vdisk1` est le nom du disque virtuel.
- `vol1` est le nom du volume existant auquel se connecter.
- `primary-vds0` est le nom du serveur de disque virtuel existant auquel se connecter.

Remarque – Les disques virtuels sont des blocs génériques associés à différents types de périphériques physiques, volumes ou fichiers. Un disque virtuel n'est pas synonyme de disque SCSI et, par conséquent, exclut l'ID cible dans l'étiquette de disque. Les disques virtuels dans un domaine logique ont le format suivant : `cNdNsN`, où `cN` est le contrôleur virtuel, `dN` est le numéro du disque virtuel et `sN` est le segment.

7 Définissez les variables `auto-boot?` et `boot-device` pour le domaine invité.

Le premier exemple de commande définit `auto-boot?` sur `true` pour le domaine invité `ldg1`.

```
primary# ldm set-var auto-boot\?=true ldg1
```

Le second exemple de commande définit `boot-device` sur `vdisk` pour le domaine invité `ldg1`.

```
primary# ldm set-var boot-device=vdisk1 ldg1
```

8 Liez les ressources au domaine invité `ldg1`, puis répertoriez le domaine pour vérifier qu'il est associé.

```
primary# ldm bind-domain ldg1
```

```
primary# ldm list-domain ldg1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	bound	-----	5000	8	2G		

9 Pour trouver le port de la console du domaine invité, vous pouvez consulter la sortie de la sous-commande précédente `list-domain`.

Vous pouvez voir sous le titre `CONS` que le domaine logique invité 1 (`ldg1`) a sa sortie de console associée au port `5000`.

10 Connectez-vous à la console d'un domaine invité à partir d'un autre terminal en vous connectant au domaine de contrôle et en vous connectant directement au port de la console sur l'hôte local.

```
$ ssh hostname.domain-name
```

```
$ telnet localhost 5000
```

11 Démarrez le domaine invité `ldg1`.

```
primary# ldm start-domain ldg1
```

Installation du SE Oracle Solaris sur un domaine invité

Cette section fournit des instructions sur différentes méthodes d'installation du SolOS; sur un domaine invité.



Attention – Ne vous déconnectez *pas* de la console virtuelle au cours de l'installation du SolOS;.

▼ Installation du SolOS; sur un domaine invité à partir d'un DVD

- 1 Insérez le DVD du SE Oracle Solaris 10 dans le lecteur DVD.

- 2 Arrêtez le démon de gestion du volume, `vold(1M)`, sur le domaine `primary`.

```
primary# svcadm disable volfs
```

- 3 Arrêtez et dissociez le domaine invité (`ldg1`).

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

- 4 Ajoutez le DVD avec le support DVD-ROM en tant que volume secondaire et disque virtuel.

L'exemple suivant utilise `c0t0d0s2` comme lecteur DVD dans lequel le support d'Oracle Solaris réside, `dvd_vol@primary-vds0` comme volume secondaire et `vdisk_cd_media` comme disque virtuel.

```
primary# ldm add-vdsdev /dev/dsk/c0t0d0s2 dvd_vol@primary-vds0
primary# ldm add-vdisk vdisk_cd_media dvd_vol@primary-vds0 ldg1
```

- 5 Vérifiez que le DVD est ajouté en tant que volume secondaire et disque virtuel.

```
primary# ldm list-bindings
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    4     4G      0.2%  22h 45m
...
VDS
NAME          VOLUME  OPTIONS  DEVICE
primary-vds0  voll             /dev/dsk/c2t1d0s2
dvd_vol                         /dev/dsk/c0t0d0s2
....
-----
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1         inactive -----  60    6G
...
DISK
NAME          VOLUME  TOUT DEVICE  SERVER
vdisk1        voll@primary-vds0
vdisk_cd_media  dvd_vol@primary-vds0
....
```

- 6 Associez et démarrez le domaine invité (`ldg1`).

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 027.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

7 Affichez les alias du périphérique dans le client PROM OpenBoot.

Dans cet exemple, recherchez les alias du périphérique pour `vdisk_cd_media`, qui est le DVD d'Oracle Solaris et `vdisk1`, qui est un disque virtuel sur lequel vous pouvez installer le SE Oracle Solaris.

```
ok devalias
vdisk_cd_media /virtual-devices@100/channel-devices@200/disk@1
vdisk1        /virtual-devices@100/channel-devices@200/disk@0
vnet1        /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name         aliases
```

8 Sur la console du domaine invité, démarrez à partir de `vdisk-cd_media (disk@1)` sur le fragment `f`.

```
ok boot vdisk_cd_media:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c), 1983-2010, Oracle and/or its affiliates. All rights reserved.
```

9 Poursuivez dans le menu d'installation du SolOS;

▼ Installation du SE Oracle Solaris sur un domaine invité à partir d'un fichier ISO de Oracle Solaris

1 Arrêtez et dissociez le domaine invité (`ldg1`).

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

2 Ajoutez le fichier ISO Oracle Solaris en tant que volume secondaire et disque virtuel.

L'exemple suivant utilise `solarisdvd.iso` en tant que fichier ISO Oracle Solaris, `iso_vol` en tant que volume secondaire et `vdisk_iso` en tant que disque virtuel :

```
primary# ldm add-vdsdev /export/solarisdvd.iso iso_vol@primary-vds0
primary# ldm-vdisk vdisk vdisk_iso iso_vol@primary-vds0 ldg1
```

3 Vérifiez que le fichier ISO Oracle Solaris est ajouté en tant que volume secondaire et disque virtuel.

```
primary# ldm list-bindings
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    4     4G     0.2%  22h 45m
...
VDS
NAME          VOLUME  OPTIONS  DEVICE
primary-vds0  vol1             /dev/dsk/c2t1d0s2
iso_vol      iso_vol          /export/solarisdvd.iso
....
-----
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1         inactive -----  60    6G
```

```

...
DISK
  NAME          VOLUME          TOUT DEVICE  SERVER
  vdisk1        vol1@primary-vds0
  vdisk_iso     iso_vol@primary-vds0
...

```

4 Associez et démarrez le domaine invité (ldg1).

```

primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..

```

5 Affichez les alias du périphérique dans le client PROM OpenBoot.

Dans cet exemple, recherchez les alias de périphérique pour `vdisk_iso`, qui est l'image ISO Oracle Solaris et `vdisk_install` qui est l'espace disque.

```

ok devalias
vdisk_iso      /virtual-devices@100/channel-devices@200/disk@1
vdisk1         /virtual-devices@100/channel-devices@200/disk@0
vnet1         /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name          aliases

```

6 Sur la console du domaine invité, démarrez à partir de `vdisk_iso (disk@1)` sur le fragment `f`.

```

ok boot vdisk_iso:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic 139555-08 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.

```

7 Poursuivez dans le menu d'installation du SolOS;

▼ Réalisation d'une opération JumpStart sur un domaine invité

Cette procédure décrit comment effectuer une opération JumpStart sur un domaine invité. Cette procédure suit la procédure JumpStart classique, mais elle décrit le format de nom de périphérique de disque différent à utiliser dans le profil JumpStart pour le domaine invité. Reportez-vous au [Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée](#).

Les noms de périphérique de disque virtuel dans un domaine logique diffèrent des noms de périphériques de disque physique. Les noms de périphérique de disque virtuel ne contiennent pas d'ID cible (tN).

Au lieu du format usuel $cNtNdNsN$, les noms de périphérique de disque virtuel utilise le format $cNdNsN$. cN est le contrôleur virtuel, dN est le numéro de disque virtuel et sN est le numéro de fragment.

- **Modifiez votre profil JumpStart pour prendre en compte cette modification.**

Un disque virtuel peut apparaître comme un disque complet ou un disque à fragment unique. Le SE Oracle Solaris peut être installé sur un disque complet l'aide d'un profil JumpStart normal qui définit plusieurs partitions. Un disque à fragment unique n'a qu'une seule partition, $s0$, qui utilise tout le disque. Pour installer le SE Oracle Solaris sur un seul disque, vous devez utiliser un profil ayant une seule partition ($/$) qui utilise tout le disque. Vous ne pouvez pas définir d'autres partitions, notamment un swap. Pour plus d'informations sur les disques complets et les disques à fragment unique, reportez-vous à la section “[Apparence du disque virtuel](#)” à la page 87.

- **Profil JumpStart d'installation d'un système de fichiers root UFS.**

Reportez-vous au [Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée](#).

Profil UFS normal

```
filesys c1t1d0s0 free /
filesys c1t1d0s1 2048 swap
filesys c1t1d0s5 120 /spare1
filesys c1t1d0s6 120 /spare2
```

Profil UFS réel pour installer un domaine sur un disque complet

```
filesys c0d0s0 free /
filesys c0d0s1 2048 swap
filesys c0d0s5 120 /spare1
filesys c0d0s6 120 /spare2
```

Profil UFS réel pour installer un domaine sur un disque à fragment unique

```
filesys c0d0s0 free /
```

- **Profil JumpStart d'installation d'un système de fichiers racine ZFS.**

Reportez-vous au [Chapitre 9, “Installation d’un pool racine ZFS à l’aide de JumpStart”](#) du [Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée](#).

Profil ZFS normal

```
pool rpool auto 2G 2G c1t1d0s0
```

Profil ZFS réel pour l'installation d'un domaine

```
pool rpool auto 2G 2G c0d0s0
```

Configuration des domaines d'E/S

Ce chapitre décrit les domaines d'E/S et la procédure de configuration de ces derniers dans un environnement de domaines logiques.

Ce chapitre aborde les sujets suivants :

- “Présentation d'un domaine d'E/S” à la page 65
- “Assignation de bus PCIe” à la page 66
- “Assignation des périphériques d'extrémité PCIe” à la page 71

Présentation d'un domaine d'E/S

Un domaine d'E/S a la propriété directe de l'accès direct aux périphériques d'E/S physiques. Il peut être créé en assignant un bus PCIe (PCI EXPRESS) ou un périphérique d'extrémité PCIe à un domaine. Utilisez la commande `ldm add-io` pour assigner un bus ou un périphérique à un domaine.

Vous pouvez vouloir configurer des domaines d'E/S pour les raisons suivantes :

- Un domaine d'E/S a un accès direct à un périphérique d'E/S physique, ce qui évite le temps système de performance associé à l'E/S virtuelle. En conséquence, les performances de l'E/S sur un domaine d'E/S correspondent plus étroitement aux performances d'E/S sur un système brut.
- Un domaine d'E/S peut héberger des services d'E/S virtuels pour une utilisation par d'autres domaines invités.

Pour plus d'informations sur la configuration des domaines d'e/S, reportez-vous aux documents suivants :

- “Assignation de bus PCIe” à la page 66
- “Assignation des périphériques d'extrémité PCIe” à la page 71

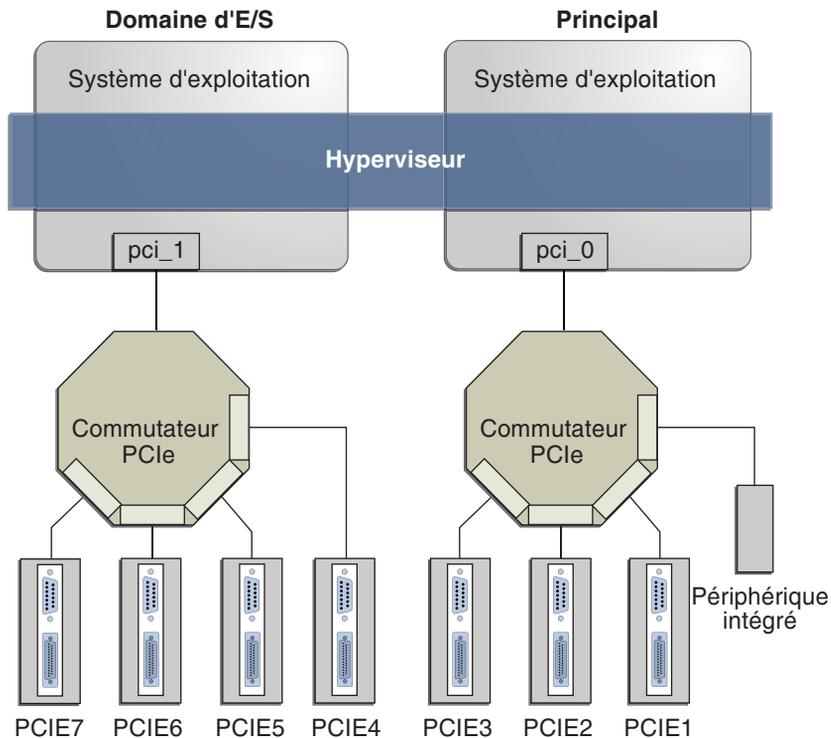
Remarque – Vous *ne pouvez pas* migrer un domaine d'E/S configuré avec des périphériques d'extrémité PCIe. Pour plus d'informations sur les autres limitations de migration, reportez-vous au [Chapitre 9, "Migration des domaines"](#).

Assignation de bus PCIe

Vous pouvez utiliser le logiciel Oracle VM Server for SPARC pour assigner un bus PCIe entier (également appelé *root complex*) à un domaine. Un bus PCIe entier est composé du bus PCIe lui-même et de tous ses commutateurs et périphériques PCI. Les bus PCIe présents sur un serveur sont identifiés par des noms tels que `pci@400` (`pci_0`). Un domaine d'E/S configuré avec un bus PCIe entier est également appelé *root domain*.

Le schéma suivante représente un système avec deux bus PCIe (`pci_0` et `pci_1`). Chaque bus est assigné à un domaine différent. Ainsi, le système est configuré avec deux domaines d'E/S.

FIGURE 6-1 Assignation d'un bus PCIe à un domaine d'E/S



Le nombre maximum de domaines d'E/S que vous pouvez créer avec des bus PCIe dépend du nombre de bus PCIe disponibles sur le serveur. Par exemple, si vous utilisez un serveur Sun SPARC Enterprise T5440, vous pouvez avoir jusqu'à quatre domaines d'E/S.

Remarque – Certains serveurs Sun UltraSPARC ne peuvent avoir qu'un seul bus PCIe. Dans ce cas, vous pouvez créer un domaine d'E/S en assignant un périphérique d'extrémité PCIe (ou périphérique assignable à une E/S directe) à un domaine. Voir [“Assignation des périphériques d'extrémité PCIe” à la page 71](#). Si le système a une unité d'interface réseau (NIU), vous pouvez également assigner une NIU à un domaine pour créer un domaine d'E/S.

Lorsque vous assignez un bus PCIe à un domaine d'E/S, tous les périphériques sur ce bus sont détenus par le domaine d'E/S. Vous n'êtes pas autorisé à assigner l'un des périphériques d'extrémité PCIe sur ce bus à d'autres domaines. Seuls les périphériques d'extrémité PCIe sur les bus PCIe qui sont assignés au domaine `primary` peuvent être assignés à d'autres domaines.

Si un serveur est configuré dès le départ dans un environnement de &LDoms ou utilise la configuration `factory-default`, le domaine `primary` a accès à toutes les ressources du périphérique physique. Cela signifie que le domaine `primary` est le seul domaine d'E/S configuré sur le système et qu'il possède tous les bus PCIe.

▼ Création d'un domaine d'E/S en assignant un bus PCIe

Cet exemple de procédure montre comment créer un nouveau domaine d'E/S à partir de la configuration initiale dans laquelle plusieurs bus sont détenus par le domaine `primary`. Par défaut, le domaine `primary` détient tous les bus présents sur le système. Cet exemple est pour un serveur Sun SPARC Enterprise T5440. Cette procédure peut également être utilisée sur d'autres serveurs. Les instructions pour les différents serveurs peuvent varier légèrement de celles-ci, mais vous pouvez obtenir les principales de base à partir de cet exemple.

En premier lieu, vous devez conserver le bus ayant le disque de démarrage du domaine `primary`. Supprimez ensuite un autre bus du domaine `primary` et assignez-le à un autre domaine.



Attention – Tous les disques internes sur les serveurs pris en charge sont connectés à un seul bus PCIe. Si un domaine est démarré à partir d'un disque interne, ne supprimez pas de bus du domaine. Vérifiez également que vous n'êtes pas en train de supprimer un bus avec des périphériques (notamment des ports réseau) utilisés par un domaine. Si vous supprimez le mauvais bus, le domaine risque de ne pas pouvoir accéder aux périphériques requis et peut devenir inutilisable. Pour supprimer un bus qui a des périphériques utilisés par un domaine, reconfigurez ce domaine pour qu'il utilise des périphériques d'autres bus. Par exemple, vous devrez peut-être reconfigurer le domaine pour qu'il utilise un port réseau intégré différent ou une carte PCIe d'un emplacement PCIe différent.

Dans cet exemple, le domaine `primary` n'utilise qu'un pool ZFS (`rpool (c0t1d0s0)`) et une interface réseau (`nxge0`). Si le domaine `primary` utilise plusieurs périphériques, répétez les étapes 2 à 4 pour chaque périphérique afin de vous assurer qu'aucun ne soit situé sur le bus qui va être supprimé.

1 Vérifiez que le domaine `primary` détient plusieurs bus PCIe.

```
primary# ldm list-io
IO          PSEUDONYM      DOMAIN
--          -
pci@400     pci_0          primary
pci@500     pci_1          primary
pci@600     pci_2          primary
pci@700     pci_3          primary
```

```
PCIIE          PSEUDONYM  STATUS  DOMAIN
-----
pci@400/pci@0/pci@d MB/PCIIE0  EMP     -
pci@400/pci@0/pci@c MB/PCIIE1  OCC     primary
pci@400/pci@0/pci@l MB/HBA     OCC     primary
pci@500/pci@0/pci@d MB/PCIIE4  EMP     -
pci@500/pci@0/pci@9 MB/PCIIE5  EMP     -
pci@500/pci@0/pci@c MB/NET0    OCC     primary
pci@600/pci@0/pci@c MB/PCIIE2  OCC     primary
pci@600/pci@0/pci@9 MB/PCIIE3  OCC     primary
pci@700/pci@0/pci@c MB/PCIIE6  OCC     primary
pci@700/pci@0/pci@9 MB/PCIIE7  EMP     -
```

2 Déterminez le chemin du périphérique du disque d'initialisation qui doit être conservé.

- Pour les systèmes de fichiers UFS, exécutez la commande `df /` pour déterminer le chemin de périphérique du disque d'initialisation.

```
primary# df /
/                (/dev/dsk/c0t1d0s0) : 1309384 blocks  457028 files
```

- Pour les systèmes de fichiers ZFS, exécutez d'abord la commande `df /` pour déterminer le nom du pool, puis exécutez la commande `zpool status` pour déterminer le chemin de périphérique du disque d'initialisation.

```
primary# df /
/ (rpool/ROOT/s10s_u8wos_08a):245176332 blocks 245176332 files
primary# zpool status rpool
zpool status rpool
  pool: rpool
  state: ONLINE
  scrub: none requested
  config:

    NAME          STATE          READ WRITE CKSUM
    rpool         ONLINE         0     0     0
    c0t1d0s0     ONLINE         0     0     0
```

3 Déterminez le périphérique physique auquel le périphérique en mode bloc est connecté.

L'exemple suivant utilise le périphérique en mode bloc `c1t0d0s0` :

```
primary# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx  1 root  root           49 Oct  1 10:39 /dev/dsk/c0t1d0s0 ->
../.. /devices/pci@400/pci@0/pci@1/scsi@0/sd@1,0:a
```

Dans cet exemple, le périphérique physique du disque d'initialisation du domaine primary est connecté au bus `pci@400`, qui correspond à la liste précédente de `pci_0`. Cela signifie que vous ne pouvez pas assigner `pci_0` (`pci@400`) à un autre domaine.

4 Déterminez l'interface réseau qui est utilisée par le système.

```
primary# dladm show-dev
vsw0      link: up      speed: 1000 Mbps      duplex: full
nxge0     link: up      speed: 1000 Mbps      duplex: full
nxge1     link: unknown speed: 0 Mbps          duplex: unknown
nxge2     link: unknown speed: 0 Mbps          duplex: unknown
nxge3     link: unknown speed: 0 Mbps          duplex: unknown
```

Les interfaces qui sont à l'état `unknown` ne sont pas configurées, par conséquent elles ne sont pas utilisées. Dans cet exemple, l'interface `nxge0` est utilisée.

5 Déterminez le périphérique physique auquel l'interface réseau est connectée.

La commande suivante utilise l'interface réseau `nxge0` :

```
primary# ls -l /dev/nxge0
lrwxrwxrwx  1 root  root           46 Oct  1 10:39 /dev/nxge0 ->
../devices/pci@500/pci@0/pci@c/network@0:nxge0
```

Dans cet exemple, le périphérique physique utilisé par le domaine primary se trouve sous le bus `pci@500`, qui correspond à la liste précédente de `pci_1`. Par conséquent, les deux autres bus, `pci_2` (`pci@600`) et `pci_3` (`pci@700`), peuvent être assignés à d'autres domaines en toute sécurité, car ils ne sont pas utilisés par le domaine primary.

Si l'interface réseau utilisée par le domaine primary se trouvait sur un bus que vous voulez assigner à un autre domaine, le domaine primary devrait être reconfiguré pour utiliser une interface réseau différente.

6 Supprimez les bus ne contenant pas le disque d'initialisation ou l'interface réseau du domaine primary.

Dans cet exemple, le bus pci_2 et le bus pci_3 sont supprimés du domaine primary. Vous risquez de voir un message de la commande ldm indiquant que le domaine primary passe en mode de reconfiguration retardée.

```
primary# ldm remove-io pci_2 primary
primary# ldm remove-io pci_3 primary
```

7 Enregistrez cette configuration sur le processeur de service.

Dans cet exemple, la configuration est io-domain.

```
primary# ldm add-config io-domain
```

Cette configuration, io-domain, est également définie comme la configuration suivante devant être utilisée après le redémarrage.

Remarque – Actuellement, il y a une limite de 8 configurations pouvant être enregistrées sur le SP, sans compter la configuration factory-default.

8 Redémarrez le domaine primary afin que les modifications soient appliquées.

```
primary# shutdown -i6 -g0 -y
```

9 Arrêtez le domaine auquel vous souhaitez ajouter le bus PCIe.

L'exemple suivant arrête le domaine ldg1 :

```
primary# ldm stop ldg1
```

10 Ajoutez le bus disponible au domaine ayant besoin d'un accès direct.

Le bus disponible est pci_2 et le domaine est ldg1.

```
primary# ldm add-io pci_2 ldg1
```

11 Redémarrez le domaine afin que les modifications soient appliquées.

Les commandes suivantes redémarre le domaine ldg1 :

```
primary# ldm start ldg1
```

12 Confirmez que le bus correct est toujours assigné au domaine primary et que le bus correct est assigné au domaine ldg1.

```
primary# ldm list-io
IO          PSEUDONYM      DOMAIN
--          -
pci@400     pci_0           primary
pci@500     pci_1           primary
pci@600     pci_2           ldg1
pci@700     pci_3

PCIe        PSEUDONYM      STATUS  DOMAIN
-----
```

pci@400/pci@0/pci@d	MB/PCIE0	EMP	-
pci@400/pci@0/pci@c	MB/PCIE1	OCC	primary
pci@400/pci@0/pci@l	MB/HBA	OCC	primary
pci@500/pci@0/pci@d	MB/PCIE4	EMP	-
pci@500/pci@0/pci@9	MB/PCIE5	EMP	-
pci@500/pci@0/pci@c	MB/NET0	OCC	primary
pci@600/pci@0/pci@c	MB/PCIE2	UNK	-
pci@600/pci@0/pci@9	MB/PCIE3	UNK	-
pci@700/pci@0/pci@c	MB/PCIE6	UNK	-
pci@700/pci@0/pci@9	MB/PCIE7	UNK	-

Cette sortie confirme que les bus PCIe pci_0 et pci_1 et les périphériques situés en dessous sont assignés au domaine primary, et que pci_2 et ses périphériques sont assignés à ldg1.

Assignation des périphériques d'extrémité PCIe

Cette utilisation avec la version 2.0 de Oracle VM Server for SPARC et le SE Oracle Solaris 10 9/10, vous pouvez assigner un périphérique d'extrémité PCIe (ou pouvant être associé à des E/S directement) à un domaine. Cette utilisation des périphériques d'extrémité PCIe augmente la granularité de l'assignation de périphériques aux domaines d'E/S. Cette fonctionnalité est fournie au moyen de la fonction d'E/S directes (DIO).

La fonction DIO vous permet de créer plus de domaines d'E/S que le nombre de bus PCIe dans un système. Le nombre possible de domaines d'E/S est maintenant limité uniquement au nombre de périphériques d'extrémité PCIe.

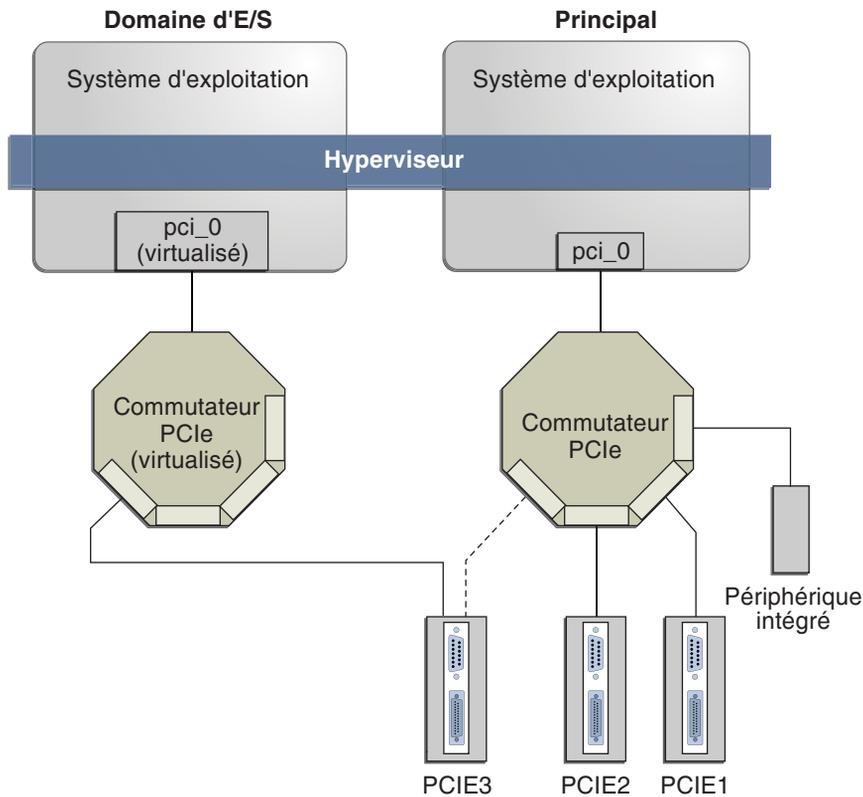
Un périphérique d'extrémité PCIe peut être l'un des suivants :

- Une carte PCIe dans un emplacement
- Un périphérique PCIe intégré identifié par la plate-forme

Le schéma suivant montre que le périphérique d'extrémité PCIe, PCIE3, est assigné à un domaine d'E/S. Le bus pci_0 et le commutateur du domaine d'E/S sont virtuels. Le périphérique d'extrémité PCIE3 n'est plus accessible dans le domaine primary.

Dans le domaine d'E/S, le bloc pci_0 et le commutateur sont un complexe racine virtuel et un commutateur PCIe virtuel, respectivement. Ce bloc et ce commutateur sont très semblables au bloc pci_0 et au commutateur du domaine primary. Dans le domaine primary, les périphériques de l'emplacement PCIE3 sont une forme d'ombre des périphériques d'origine et sont identifiés par SUNW, assigned.

FIGURE 6-2 Assignation d'un périphérique d'extrémité PCIe à un domaine d'E/S



Utilisez la commande `ldm list -io` pour répertorier les périphériques d'extrémité PCIe.

Bien que la fonction DIO permette à toute carte PCIe d'un emplacement d'être assignée à un domaine d'E/S, seules certaines cartes PCIe sont prises en charge. Reportez-vous à la section "Configuration matérielle et logicielle requise pour les E/S directes" des *Notes de version Oracle VM Server for SPARC 2.0*.

Remarque – Les cartes PCIe ayant un commutateur ou un pont ne sont pas prises en charge. L'assignation au niveau de la fonction PCIe n'est pas prise en charge. L'assignation d'une carte PCIe non prise en charge à un domaine d'E/S peut provoquer un comportement imprévisible.

Voici quelques détails importants sur la fonction DIO :

- Cette fonction n'est activée que lorsque la configuration logicielle requise est respectée. Reportez-vous à la section “Configuration matérielle et logicielle requise pour les E/S directes” des *Notes de version Oracle VM Server for SPARC 2.0*.
- Seules les extrémités connectées à un bus PCIe assigné au domaine `primary` peuvent être assignées à un autre domaine avec la fonction DIO.
- Les domaines d'E/S utilisant les DIO ont accès aux périphériques d'extrémité PCIe uniquement si le domaine `primary` est en cours d'exécution.
- Le redémarrage du domaine `primary` a une incidence sur les domaines d'E/S ayant des périphériques d'extrémité PCIe. Reportez-vous à la section “Redémarrage du domaine `primary`” à la page 75. Le domaine `primary` a également les responsabilités suivantes :
 - Initialise le bus PCIe et gère ce bus.
 - Traite toutes les erreurs de bus déclenchées par les périphériques d'extrémité PCIe assignés aux domaines d'E/S. Notez que seul le domaine `primary` reçoit toutes les erreurs relatives au bus PCIe.

Configuration matérielle et logicielle requise pour les E/S directes

Pour utiliser avec succès la fonction DIO, vous devez exécuter le logiciel approprié et n'assigner que des cartes PCIe prises en charge par la fonction DIO vers les domaines d'E/S. Pour plus d'informations sur la configuration matérielle et logicielle requise, reportez-vous à la section “Configuration matérielle et logicielle requise pour les E/S directes” des *Notes de version Oracle VM Server for SPARC 2.0*.

Remarque – Toutes les cartes PCIe prises en charge sur la plate-forme sont prises en charge dans le domaine `primary`. Reportez-vous à la documentation de votre plate-forme pour obtenir la liste des cartes PCIe prises en charge. Cependant, *seules* les cartes PCIe d'E/S directe peuvent être assignées aux domaines d'E/S.

Restrictions des E/S directes

Pour plus d'informations sur la manière de contourner les restrictions suivantes, reportez-vous à la section “Planification de la configuration des périphériques d'extrémité PCIe” à la page 74.

- Une reconfiguration retardée est démarrée lorsque vous assignez un périphérique d'extrémité PCIe au domaine `primary` ou que vous le supprimez, ce qui signifie que les modifications ne sont appliquées qu'après le redémarrage du domaine `primary`.

Le redémarrage du domaine `primary` affecte l'E/S directe. En conséquence, planifiez correctement vos modifications de configuration d'E/S directe pour optimiser les modifications relatives aux E/S directes sur le domaine `primary` et pour réduire les redémarrages du domaine `primary`.

- L'assignation d'un périphérique d'extrémité PCIe à un autre domaine ou sa suppression est uniquement autorisée lorsque ce domaine est arrêté ou inactif.

Planification de la configuration des périphériques d'extrémité PCIe

Planifiez soigneusement à l'avance l'assignation ou la suppression des périphériques d'extrémité PCIe pour éviter les indisponibilités du domaine `primary`. Le redémarrage du domaine `primary` a non seulement une incidence sur les services disponibles sur le domaine `primary` lui-même, mais affecte également les domaines d'E/S ayant des périphériques d'extrémité PCIe assignés. Bien que les modifications apportées à chaque domaine d'E/S n'affectent pas les autres domaines, la planification anticipée permet de minimiser les conséquences sur les services fournis par ce domaine.

La reconfiguration retardée est initiée la première fois que vous assignez ou supprimez un périphérique. En conséquence, vous pouvez continuer à ajouter ou à supprimer plus de périphériques, puis redémarrer le domaine `primary` une seule fois pour appliquer toutes les modifications.

Pour obtenir un exemple, reportez-vous à la section [“Création d'un domaine d'E/S en assignant un périphérique d'extrémité PCIe”](#) à la page 77.

Voici une descriptions de la procédure générale que vous devez suivre pour planifier et effectuer une configuration de périphérique DIO :

1. Comprenez et enregistrez la configuration matérielle de votre système.

Plus précisément, enregistrez les informations sur les numéros de référence et d'autres détails des cartes PCIe dans le système.

Utilisez les commandes `ldm list-io -l` et `prtdiag -v` pour obtenir les informations et enregistrez-les pour une référence ultérieure.

2. Déterminez quels périphériques d'extrémité PCIe doivent se trouver dans le domaine `primary`.

Par exemple, déterminez les périphériques d'extrémité PCIe fournissant l'accès aux éléments suivants :

- Périphérique de disque d'initialisation
- Périphérique réseau
- Autres périphériques que le domaine `primary` offre comme services

3. Supprimez tous les périphériques d'extrémité PCIe qui vous risquez d'utiliser dans les domaines d'E/S.

Cette étape vous aide à éviter d'effectuer des opérations de redémarrage ultérieures sur le domaine `primary`, car les redémarrages ont une incidence sur les domaines d'E/S.

Utilisez la commande `ldm rm-io` pour supprimer les périphériques d'extrémité PCIe. Utilisez des pseudonymes plutôt que les chemins d'accès au périphérique pour indiquer les périphériques aux sous-commandes `rm-io` et `add-io`.

Remarque – Bien que la première suppression d'un périphérique d'extrémité PCIe risque de démarrer une reconfiguration retardée, vous pouvez poursuivre la suppression des périphériques. Après avoir supprimé tous les périphériques que vous souhaitez, vous devez seulement redémarrer le domaine `primary` une fois pour appliquer toutes les modifications.

4. Enregistrez cette configuration sur le processeur de service (SP).

Utilisez la commande `ldm add-config`.

5. Redémarrez le domaine `primary` pour libérer les périphériques d'extrémité PCIe que vous avez supprimés à l'étape 3.

6. Confirmez que les périphériques d'extrémité sur vous supprimez ne sont plus assignés au domaine `primary`.

Utilisez la commande `ldm list-io -l` pour vérifier que les périphériques que vous avez supprimés apparaissent comme `SUNW, assigned-device` dans la sortie.

7. Assignez le périphérique d'extrémité PCIe disponible à un domaine invité pour fournir un accès direct au périphérique physique.

Après avoir effectué cette assignation, vous ne pouvez plus migrer le domaine invité vers un autre système physique au moyen de la fonction de migration.

8. Ajoutez un périphérique d'extrémité PCIe à un domaine invité ou supprimez-en un.

Utilisez la commande `ldm add-io`.

Minimisez les modifications aux domaines d'E/S en réduisant les opérations de redémarrage et en évitant les indisponibilités des services offerts par ce domaine.

9. (Facultatif) Apportez des modifications au matériel PCIe.

Reportez-vous à la section "[Procédure de modification matérielle PCIe](#)" à la page 76.

Redémarrage du domaine `primary`

Le domaine `primary` est le propriétaire du bus PCIe et est responsable de l'initialisation et de la gestion de ce bus. Le domaine `primary` doit être actif et exécuter une version du SE Oracle Solaris prenant en charge la fonction DIO. L'arrêt, l'interruption ou le redémarrage du domaine

`primary` interrompt l'accès au bus PCIe. Lorsque le bus PCIe est indisponible, les périphériques PCIe sur ce bus sont affectés et risquent de devenir indisponibles.

Le comportement des domaines d'E/S avec des périphériques d'extrémité PCIe est imprévisible lorsque le domaine `primary` est redémarré tandis que ces domaines d'E/S sont en cours d'exécution. Par exemple, les domaines d'E/S avec des périphériques d'extrémité PCIe peuvent paniquer au cours ou après un redémarrage. Lors du redémarrage du domaine `primary`, vous devrez arrêter et démarrer manuellement chaque domaine.

Pour contourner ces problèmes, procédez comme suit :

- Arrêtez manuellement les domaines du système ayant des périphériques d'extrémité PCIe leur étant assignés *avant* d'arrêter le domaine `primary`.

Cette étape garantit que les domaines sont arrêtés proprement avant d'arrêter, suspendre ou redémarrer le domaine `primary`.

Pour rechercher tous les domaines ayant des périphériques d'extrémité PCIe leur étant assignés, exécutez la commande `ldm list -io`. Cette commande vous permet de répertorier les périphériques d'extrémité PCIe ayant été assignés aux domaines sur le système. Utilisez donc ces informations pour vous aider à la planification. Pour obtenir une description détaillée de la sortie de cette commande, reportez-vous à la page de manuel [ldm\(1M\)](#).

Pour chaque domaine trouvé, arrêtez le domaine en exécutant la commande `ldm stop`.

- Configurez une relation de dépendance de domaine entre le domaine `primary` et les domaines ayant des périphériques d'extrémité PCIe leur étant assignés.

Cette relation de dépendance garantit que les domaines avec des périphériques d'extrémité PCIe sont automatiquement redémarrés lorsque le domaine `primary` redémarre pour quelque raison que ce soit.

Notez que cette relation de dépendance réinitialise ces domaines et qu'ils ne peuvent pas s'arrêter correctement. Cependant, la relation de dépendance n'a aucune incidence sur les domaines ayant été manuellement arrêtés.

```
# ldm set-domain failure-policy=reset primary
# ldm set-domain master=primary ldom
```

Procédure de modification matérielle PCIe

La procédure suivante vous aide à éviter la mauvaise configuration des assignations d'extrémité PCIe. Pour obtenir des informations spécifiques à la plate-forme sur l'installation et la suppression de matériel spécifique, reportez-vous à la documentation de votre plate-forme.

- Aucune action n'est nécessaire si vous installez une carte PCIe dans un emplacement vide. La carte PCIe est automatiquement détenue par le domaine propriétaire du bus PCIe.

Pour assigner une nouvelle carte PCIe à un domaine d'E/S, utilisez la commande `ldm rm -io` pour d'abord supprimer la carte du domaine `primary`. Utilisez ensuite la commande `ldm add -io` pour assigner la carte à un domaine d'E/S.

- Aucune action n'est nécessaire si la carte PCIe est supprimée du système et assignée au domaine `primary`.
- Pour supprimer une carte PCIe assignée à un domaine d'E/S, supprimez d'abord le périphérique du domaine d'E/S. Ajoutez ensuite le périphérique au domaine `primary` avant de supprimer physiquement le périphérique du système.
- Pour remplacer une carte PCIe assignée à un domaine d'E/S, vérifiez que la nouvelle carte est prise en charge par la fonction DIO.

Si tel est le cas, aucune action n'est nécessaire pour assigner automatiquement la nouvelle carte au domaine d'E/S actuel.

Sinon, supprimez d'abord la carte PCIe du domaine d'E/S à l'aide de la commande `ldm rm -io`. Utilisez ensuite la commande `ldm add -io` pour réassigner cette carte PCIe au domaine `primary`. Remplacez ensuite physiquement la carte PCIe que vous avez assignée au domaine `primary` par une carte PCIe différente. Ces étapes vous permettent d'éviter une configuration non prise en charge par la fonction DIO.

▼ Création d'un domaine d'E/S en assignant un périphérique d'extrémité PCIe

Planifiez tous les déploiements DIO à l'avance pour réduire les indisponibilités.

Pour obtenir un exemple d'ajout de périphérique d'extrémité PCIe pour créer un domaine d'E/S, reportez-vous à la section [“Planification de la configuration des périphériques d'extrémité PCIe”](#) à la page 74.

1 Identifiez et archivez les périphériques qui sont actuellement installés sur le système.

La sortie de la commande `ldm list -io -l` montre comment les périphériques d'E/S sont actuellement configurés. Vous pouvez obtenir des informations plus détaillées à l'aide de la commande `prtdiag -v`.

Remarque – Après l'assignation des périphériques aux domaines d'E/S, l'identité des périphériques peut uniquement être déterminée dans les domaines d'E/S.

```
# ldm list-io -l
IO                PSEUDONYM        DOMAIN
--                -
pci@400           pci_0             primary
pci@500           pci_1             primary

PCIe              PSEUDONYM        STATUS  DOMAIN
----              -
pci@400/pci@0/pci@c PCIe1             EMP     -
pci@400/pci@0/pci@9 PCIe2             OCC     primary
network@0
```

```

        network@0,1
        network@0,2
        network@0,3
pci@400/pci@0/pci@d  PCIE3      OCC      primary
        SUNW,emlxs/fp/disk
        SUNW,emlxs@0,1/fp/disk
        SUNW,emlxs@0,1/fp@0,0
pci@400/pci@0/pci@8  MB/SASHBA OCC      primary
        scsi@0/tape
        scsi@0/disk
        scsi@0/sd@0,0
        scsi@0/sd@1,0
pci@500/pci@0/pci@9  PCIE0      EMP      -
pci@500/pci@0/pci@d  PCIE4      OCC      primary
        network@0
        network@0,1
pci@500/pci@0/pci@c  PCIE5      OCC      primary
        SUNW,qlc@0/fp/disk
        SUNW,qlc@0/fp@0,0
        SUNW,qlc@0,1/fp/disk
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c605dbab,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c6041434,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c6053652,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c6041b4f,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c605dbb3,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c60413bc,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c604167f,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c6041b3a,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c605dabf,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c60417a4,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c60416a7,0
        SUNW,qlc@0,1/fp@0,0/ssd@w21000011c60417e7,0
        SUNW,qlc@0,1/fp@0,0/ses@w215000c0ff082669,0
pci@500/pci@0/pci@8  MB/NET0    OCC      primary
        network@0
        network@0,1
        network@0,2
        network@0,3
    
```

2 Déterminez le chemin du périphérique du disque d'initialisation qui doit être conservé.

- Pour les systèmes de fichiers UFS, exécutez la commande `df /` pour déterminer le chemin de périphérique du disque d'initialisation.

```

primary# df /
/                (/dev/dsk/c0t1d0s0) : 1309384 blocks  457028 files
    
```

- Pour les systèmes de fichiers ZFS, exécutez d'abord la commande `df /` pour déterminer le nom du pool, puis exécutez la commande `zpool status` pour déterminer le chemin de périphérique du disque d'initialisation.

```

primary# df /
/                (rpool/ROOT/s10s_u8wos_08a):245176332 blocks 245176332 files
primary# zpool status rpool
zpool status rpool
  pool: rpool
  state: ONLINE
    
```

```
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
c0t1d0s0	ONLINE	0	0	0

3 Déterminez le périphérique physique auquel le périphérique en mode bloc est connecté.

L'exemple suivant utilise le périphérique en mode bloc `c0t1d0s0` :

```
primary# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx  1 root  root   49 Jul 20 22:17 /dev/dsk/c0t1d0s0 ->
../..../devices/pci@400/pci@0/pci@8/scsi@0/sd@0,0:a
```

Dans cet exemple, le périphérique physique du disque d'initialisation du domaine `primary` est connecté au périphérique d'extrémité PCIe (`pci@400/pci@0/pci@8`), qui correspond à la liste de MB/SASHBA à l'étape 1. La suppression de ce périphérique empêchera le redémarrage du domaine `primary`. Par conséquent, *ne* supprimez pas ce périphérique du domaine `primary`.

4 Déterminez l'interface réseau qui est utilisée par le système.

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
nxge0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 2
    inet 10.6.212.149 netmask fffffe00 broadcast 10.6.213.255
    ether 0:21:28:4:27:cc
```

Dans cet exemple, l'interface `nxge0` est utilisées comme interface réseau pour le domaine `primary`

5 Déterminez le périphérique physique auquel l'interface réseau est connectée.

La commande suivante utilise l'interface réseau `nxge0` :

```
primary# ls -l /dev/nxge0
lrwxrwxrwx  1 root  root   46 Jul 30 17:29 /dev/nxge0 ->
../devices/pci@500/pci@0/pci@8/network@0:nxge0
```

Dans cet exemple, le périphérique physique de l'interface réseau utilisée par le domaine `primary` est connecté au périphérique d'extrémité PCIe (`pci@500/pci@8`), qui correspond à la liste de MB/NET0 à l'étape 1. Par conséquent, vous *ne* voulez pas supprimer ce périphérique du domaine `primary`. Vous pouvez assigner en toute sécurité tous les autres périphériques PCIe aux autres domaines, car ils ne sont pas utilisés par le domaine `primary`.

Si l'interface réseau utilisée par le domaine `primary` se trouve sur un bus que vous voulez assigner à un autre domaine, le domaine `primary` doit être reconfiguré pour utiliser une interface réseau différente.

6 Supprimez les périphériques d'extrémité PCIe que vous risquez d'utiliser dans les domaines d'E/S.

Dans cet exemple, vous pouvez supprimer les périphériques d'extrémité PCIe2, PCIe3, PCIe4 et PCIe5, car ils ne sont pas utilisés par le domaine primary.

a. Supprimez les périphériques d'extrémité PCIe.



Attention – Ne supprimez pas les périphériques utilisés par le domaine primary.

Si vous supprimez par erreur les mauvais périphériques, utilisez la commande `ldm cancel -op reconf primary` pour annuler la reconfiguration retardée sur le domaine primary.

Vous pouvez supprimer plusieurs périphériques à la fois pour éviter plusieurs redémarrages.

```
# ldm rm-io PCIe2 primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
# ldm rm-io PCIe3 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm rm-io PCIe4 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
# ldm rm-io PCIe5 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

b. Enregistrez la nouvelle configuration sur le processeur de service (SP).

La commande suivante enregistre la configuration dans un fichier nommé `dio` :

```
# ldm add-config dio
```

c. Redémarrez le système pour prendre en compte la suppression des périphériques d'extrémité PCIe.

```
# reboot -- -r
```

7 Connectez-vous au domaine primary et vérifiez que les périphériques d'extrémité PCIe ne sont plus assignés au domaine.

```
# ldm list-io
IO                PSEUDONYM        DOMAIN
```

```

--
pci@400      -----
pci@500      pci_0      primary
pci@500      pci_1      primary

PCIe                PSEUDONYM  STATUS  DOMAIN
-----
pci@400/pci@0/pci@c PCIE1     EMP     -
pci@400/pci@0/pci@9 PCIE2     OCC
pci@400/pci@0/pci@d PCIE3     OCC
pci@400/pci@0/pci@8 MB/SASHBA OCC     primary
pci@500/pci@0/pci@9 PCIE0     EMP     -
pci@500/pci@0/pci@d PCIE4     OCC
pci@500/pci@0/pci@c PCIE5     OCC
pci@500/pci@0/pci@8 MB/NET0   OCC     primary
    
```

Remarque – La sortie `ldm list -io -l` peut indiquer `SUNW, assigned-device` pour les périphériques d'extrémité PCIe supprimés. Les informations actuelles ne sont plus disponibles à partir du domaine `primary`, mais le domaine auquel le périphérique est assigné a ces informations.

8 Assignez un périphérique d'extrémité PCIe à un domaine.

a. Ajoutez le périphérique PCI2 au domaine `ldg1`.

```
# ldm add-io PCIE2 ldg1
```

b. Associez et démarrez le domaine `ldg1`.

```
# ldm bind ldg1
# ldm start ldg1
LDom ldg1 started
```

9 Connectez-vous au domaine `ldg1` et vérifiez que le périphérique est disponible à l'utilisation.

Utilisez la commande `dladm show-dev` pour vérifier que le périphérique réseau est disponible. Configurez ensuite le périphérique réseau pour une utilisation dans le domaine.

```
# dladm show-dev
vnet0      link: up      speed: 0      Mbps      duplex: unknown
nxge0      link: unknown speed: 0      Mbps      duplex: unknown
nxge1      link: unknown speed: 0      Mbps      duplex: unknown
nxge2      link: unknown speed: 0      Mbps      duplex: unknown
nxge3      link: unknown speed: 0      Mbps      duplex: unknown
```


Utilisation des disques virtuels

Ce chapitre décrit comment utiliser les disques virtuels avec le logiciel Oracle VM Server for SPARC.

Ce chapitre aborde les sujets suivants :

- “Présentation des disques virtuels” à la page 83
- “Gestion des disques virtuels” à la page 84
- “Identificateur de disque virtuel et nom de périphérique” à la page 87
- “Apparence du disque virtuel” à la page 87
- “Options de moteur de traitement du disque virtuel” à la page 88
- “Arrière-plan du disque virtuel” à la page 90
- “Configuration du multivoie d'un disque virtuel” à la page 96
- “CD, DVD et images ISO” à la page 99
- “Délai d'attente du disque virtuel” à la page 102
- “Disque virtuel et SCSI” à la page 103
- “Disque virtuel et commande format(1M)” à la page 104
- “Utilisation de ZFS avec les disques virtuels” à la page 104
- “Utilisation du gestionnaire de volumes dans un environnement de domaines logiques” à la page 108

Présentation des disques virtuels

Un disque virtuel contient deux composants : le disque virtuel lui-même tel qu'il apparaît dans un domaine invité et l'arrière-plan du disque virtuel, qui est l'endroit où les données sont stockées et où l'E/S virtuelle se termine. L'arrière-plan du disque virtuel est exporté à partir d'un domaine de service vers le pilote du serveur de disque virtuel (vds). Le pilote vds communique avec le pilote du client de disque virtuel (vdc) dans le domaine invité via l'hyperviseur à l'aide d'un canal de domaine logique (LDC). Enfin, un disque virtuel apparaît en tant que périphériques `/dev/[r]dsk/cXdYsZ` dans le domaine invité.

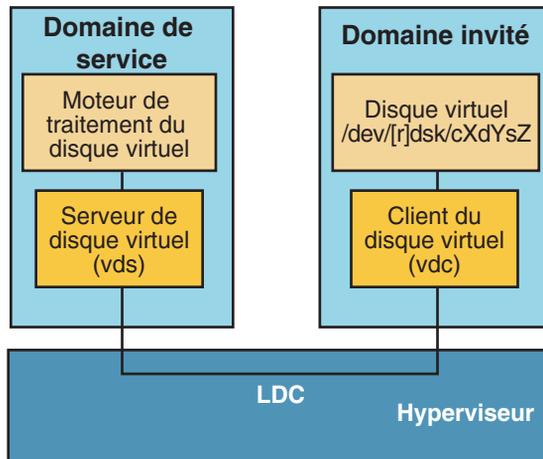
L'arrière-plan du disque virtuel peut être physique ou logique. Les périphériques physiques peuvent comprendre :

- Un disque physique ou un numéro d'unité logique (LUN)
- Un segment de disque physique

Les périphériques logiques peuvent être les suivants :

- Fichier sur un système de fichiers, notamment ZFS ou UFS
- Volume logique à partir d'un gestionnaire de volumes, notamment ZFS, VxVM ou Solaris Volume Manager
- Tout pseudo-périphérique de disque accessible à partir du domaine de service

FIGURE 7-1 Disques virtuels avec domaines logiques



Gestion des disques virtuels

Cette section décrit l'ajout d'un disque virtuel à un domaine invité, la modification des options d'un disque virtuel et de délai d'attente et la suppression d'un disque virtuel d'un domaine invité. Reportez-vous à la section [“Options de moteur de traitement du disque virtuel”](#) à la page 88 pour obtenir une description des options du disque virtuel. Reportez-vous à la section [“Délai d'attente du disque virtuel”](#) à la page 102 pour obtenir une description du délai d'attente du disque virtuel.

▼ Ajout d'un disque virtuel

- 1 Exportez l'arrière-plan du disque virtuel à partir d'un domaine de service.

```
# ldm add-vdsdev [options={ro,slice,excl}] [mpgroup=mpgroup] \  
backend volume-name@service-name
```

- 2 Assignez l'arrière-plan à un domaine invité.

```
# ldm add-vdisk [timeout=seconds] [id=disk-id] disk-name volume-name@service-name ldom
```

Vous pouvez indiquer un ID pour le nouveau périphérique de disque virtuel en définissant la propriété `id`. Par défaut, les valeurs d'ID sont générées automatiquement. Par conséquent, définissez cette propriété si vous devez faire correspondre un nom de périphérique existant dans le SE. Reportez-vous à la section “[Identificateur de disque virtuel et nom de périphérique](#)” à la page 87.

Remarque – Un arrière-plan est généralement exporté à partir du domaine de service et assigné au domaine invité lorsque ce dernier (*ldom*) est associé.

▼ Exportation multiples de l'arrière-plan du disque virtuel

L'arrière-plan d'un disque virtuel peut être exporté plusieurs fois par le même serveur de disque virtuel ou par des serveurs différents. Chaque instance exportée de l'arrière-plan du disque virtuel peut ensuite être assignée au même domaine invité ou à des domaines invités différents.

Lorsque l'arrière-plan d'un disque virtuel est exporté plusieurs fois, il ne doit pas être exporté avec l'option exclusive (`excl`). La définition de l'option `excl` ne permettra l'exportation de l'arrière-plan qu'une seule fois. L'arrière-plan peut être exporté en toute sécurité plusieurs fois en tant que périphérique en lecture seule avec l'option `ro`.



Attention – Lorsque l'arrière-plan d'un disque virtuel est exporté plusieurs fois, les applications s'exécutant sur les domaines invités et utilisant ce disque virtuel sont responsables de la coordination et de la synchronisation de l'accès simultané en écriture afin de garantir la cohérence des données.

L'exemple suivant décrit comment ajouter le même disque virtuel à deux domaines invités différents via le même serveur de disque virtuel.

- 1 Exportez l'arrière-plan du disque virtuel deux fois à partir d'un domaine de service à l'aide des commandes suivantes.

```
# ldm add-vdsdev [options={ro,slice}] backend volume1@service-name  
# ldm add-vdsdev -f [options={ro,slice}] backend volume2@service-name
```

Notez que la seconde commande `ldm add-vdsdev` utilise l'option `-f` pour forcer la seconde exportation de l'arrière-plan. Utilisez cette option lorsque vous utilisez le même chemin d'accès au moteur de traitement pour les deux commandes et lorsque les serveurs de disque virtuel sont situés sur le même domaine de service.

- 2 **Assignez l'arrière-plan exporté à chaque domaine invité à l'aide des commandes suivantes.**

disk-name peut être différent pour `ldom1` et `ldom2`.

```
# ldm add-vdisk [timeout=seconds] disk-name volume1@service-name ldom1
# ldm add-vdisk [timeout=seconds] disk-name volume2@service-name ldom2
```

▼ Modification des options du disque virtuel

Pour plus d'informations sur les options du disque virtuel, reportez-vous à la section “[Options de moteur de traitement du disque virtuel](#)” à la page 88.

- Après l'exportation d'un arrière-plan à partir du domaine de service, vous pouvez modifier les options du disque virtuel à l'aide de la commande suivante.

```
# ldm set-vdsdev options=[{ro,slice,excl}] volume-name@service-name
```

▼ Modification de l'option de délai d'attente

Pour plus d'informations sur les options du disque virtuel, reportez-vous à la section “[Options de moteur de traitement du disque virtuel](#)” à la page 88.

- Après l'assignation d'un disque virtuel à un domaine invité, vous pouvez modifier le délai d'attente du disque virtuel à l'aide de la commande suivante.

```
# ldm set-vdisk timeout=seconds disk-name ldom
```

▼ Suppression d'un disque virtuel

- 1 **Supprimez un disque virtuel d'un domaine invité à l'aide de la commande suivante.**

```
# ldm rm-vdisk disk-name ldom
```

- 2 **Arrêtez l'exportation de l'arrière-plan correspondant à partir du domaine de service à l'aide de la commande suivante.**

```
# ldm rm-vdsdev volume-name@service-name
```

Identificateur de disque virtuel et nom de périphérique

Lorsque vous utilisez la commande `ldm add-vdisk` pour ajouter un disque virtuel à un domaine, vous pouvez définir son numéro de périphérique en définissant la propriété `id`.

```
# ldm add-vdisk [id=disk-id] disk-name volume-name@service-name ldom
```

Chaque disque virtuel d'un domaine a un numéro de périphérique unique qui est assigné lorsque le domaine est associé. Si un disque virtuel a été ajouté avec un numéro de périphérique explicite (en définissant la propriété `id`), le numéro de périphérique défini est utilisé. Sinon, le système assigne automatiquement le numéro de périphérique le plus petit disponible. Dans ce cas, le numéro de périphérique assigné dépend de la manière dont les disques virtuels ont été ajoutés au domaine. Le numéro de périphérique éventuellement assigné à un disque virtuel est visible dans la sortie de la commande `ldm list-bindings` lorsqu'un domaine est associé.

Lorsqu'un domaine avec des disques virtuels exécute le SE Oracle Solaris, chaque disque virtuel apparaît dans le domaine comme un périphérique de disque `c0dn`, où `n` est le numéro de périphérique du disque virtuel.

Dans l'exemple suivant, le domaine `ldg1` a deux disques virtuels : `rootdisk` et `pdisk`. `rootdisk` a le numéro de périphérique `0` (`disk@0`) et apparaît dans le domaine en tant que périphérique de disque `c0d0`. `pdisk` a le numéro de périphérique `1` (`disk@1`) et apparaît dans le domaine en tant que périphérique de disque `c0d1`.

```
primary# ldm list-bindings ldg1
...
DISK
  NAME          VOLUME                TOUT DEVICE  SERVER      MPGROUP
  rootdisk      dsk_nevada@primary-vds0  disk@0  primary
  pdisk         c3t40d1@primary-vds0    disk@1  primary
...
```



Attention – Si un numéro de périphérique n'est pas assigné explicitement à un disque virtuel, son numéro de périphérique peut changer lorsque le domaine est dissocié, puis est réassocié ultérieurement. Dans ce cas, le nom de périphérique assigné par le SE s'exécutant dans le domaine peut également changer et rompre la configuration existante du système. Cela peut se produire, par exemple, lorsqu'un disque virtuel est supprimé de la configuration du domaine.

Apparence du disque virtuel

Lorsqu'un arrière-plan est exporté en tant que disque virtuel, il peut apparaître dans le domaine invité en tant que disque complet ou sous forme de disque à un seul segment. La manière dont il apparaît dépend du type de moteur de traitement et des options utilisées pour l'exportation.

Disque complet

Lorsqu'un arrière-plan est exporté sur un domaine en tant que disque complet, il apparaît dans ce domaine comme un disque normal avec 8 segments (s0 à s7). Un tel disque est visible avec la commande `format(1M)`. La table de partition du disque peut être modifiée à l'aide de la commande `fmthard(1M)` ou `format(1M)`.

Un disque complet est également visible au logiciel d'installation du SE et peut être sélectionné en tant que disque sur lequel le SE peut être installé.

Tout arrière-plan peut être exporté en tant que disque complet sauf pour les segments de disque physique qui peuvent uniquement être exportés comme des disques à segment unique.

Disque à segment unique

Lorsqu'un arrière-plan est exporté sur un domaine en tant que disque à segment unique, il apparaît dans ce domaine comme un disque normal avec 8 segments (s0 à s7). Cependant, seul le premier segment (s0) est utilisable. Un tel disque est visible avec la commande `format(1M)`, mais la table de partition du disque ne peut pas être modifiée.

Un disque à segment unique est également visible à partir du logiciel d'installation du SE et peut être sélectionné comme disque sur lequel vous pouvez installer le SE. Dans ce cas, si vous installez le SE à l'aide du système de fichiers UNIX (UFS), seule la partition root (/) doit être définie, et cette partition doit utiliser tout l'espace disque.

Tout arrière-plan peut être exporté en tant que disque à segment unique sauf pour les disques physiques qui peuvent uniquement être exportés comme des disques complets.

Remarque – Avant la version du SE Oracle Solaris 10 10/08, un disque à segment unique apparaissait comme un disque avec une seule partition (s0). Un tel disque n'était pas visible avec la commande `format(1M)`. Le disque n'était également pas visible à partir du logiciel d'installation du SE et ne pouvait pas être sélectionné comme périphérique de disque sur lequel installer le SE.

Options de moteur de traitement du disque virtuel

Différentes options peuvent être définies lors de l'exportation de l'arrière-plan d'un disque virtuel. Ces options sont indiquées dans l'argument `options=` de la commande `ldm add -vdsdev` sous forme de liste séparée par des virgules. Les options valides sont : `ro`, `slice` et `excl`.

Option de lecture seule `ro`

L'option de lecture seule (`ro`) indique que l'arrière-plan doit être exporté comme un périphérique en lecture seule. Dans ce cas, le disque virtuel assigné au domaine invité est uniquement accessible pour les opérations de lecture et toute opération d'écriture sur le disque virtuel échouera.

Option exclusive `excl`

L'option exclusive (`excl`) indique que l'arrière-plan du domaine de service doit être ouvert exclusivement par le serveur de disque virtuel lorsqu'il est exporté en tant que disque virtuel vers un autre domaine. Lorsqu'un arrière-plan est ouvert en exclusivité, il n'est pas accessible par les autres applications du domaine de service. Cela empêche les applications s'exécutant sur le domaine de service d'utiliser par inadvertance l'arrière-plan étant également utilisé par un domaine invité.

Remarque – Certains pilotes ne remplissent pas l'option `excl` et n'autoriseront pas certains moteurs de traitement de disque virtuel à s'ouvrir en exclusivité. L'option `excl` est connue pour fonctionner avec les disques physiques et les segments, mais l'option ne fonctionne pas sur les fichiers. Elle peut fonctionner ou non avec les pseudo-périphériques, tels que les volumes de disque. Si le pilote de l'arrière-plan n'effectue pas l'ouverture exclusive, l'option `excl` de l'arrière-plan est ignorée et celui-ci n'est pas ouvert en exclusivité.

Comme l'option `excl` empêche les applications s'exécutant dans le domaine de service d'accéder à une arrière-plan exporté sur un domaine invité, ne définissez pas l'option `excl` dans les cas suivants :

- Lorsque les domaines invités sont en cours d'exécution, si vous souhaitez utiliser les commandes telles que `format(1M)` ou `luxadm(1M)` pour gérer les disques physiques, n'exportez pas ces disques avec l'option `excl`.
- Lorsque vous exportez un volume Solaris Volume Manager, tel qu'un volume RAID ou en miroir, ne définissez pas l'option `excl`. Sinon, cela peut empêcher Solaris Volume Manager de démarrer certaines opérations de récupération si un composant du volume RAID ou en miroir échoue. Reportez-vous à la section "[Utilisation de disques virtuels au sommet de Solaris Virtual Manager](#)" à la page 109 pour plus d'informations.
- Si Veritas Volume Manager (VxVM) est installé dans le domaine de service et que Veritas Dynamic Multipathing (VxDMP) est activé pour les disques physiques, ceux-ci doivent être exportés sans l'option `excl` (n'est pas la valeur par défaut). Sinon, l'exportation échoue, car le serveur de disque virtuel (`vds`) est inutilisable pour ouvrir le périphérique de disque virtuel. Reportez-vous à la section "[Utilisation des disques virtuels lorsque VxVM est installé](#)" à la page 110 pour plus d'informations.

- Si vous exportez le même arrière-plan de disque virtuel plusieurs fois à partir du même service de disque virtuel, reportez-vous à la section [“Exportation multiples de l’arrière-plan du disque virtuel”](#) à la page 85 pour plus d’informations.

Par défaut, l’arrière-plan n’est pas ouvert en exclusivité. De cette manière, l’arrière-plan peut toujours être utilisé par les applications s’exécutant dans le domaine de service lorsqu’il est exporté dans un autre domaine. Notez qu’il s’agit d’un comportement nouveau à partir de la version du SE Oracle Solaris 10 5/08. Avant la version du SE Oracle Solaris 105/08, les moteurs de traitement de disque s’ouvraient toujours en exclusivité, et il était impossible de les ouvrir autrement.

Option de segment (s*l*ice)

Un arrière-plan est généralement exporté comme un disque complet ou un disque à segment unique en fonction de son type. Si l’option `slice` est indiquée, l’arrière-plan est exporté de force comme un disque à segment unique.

Cette option est utile lorsque vous voulez exporter le contenu brut d’un arrière-plan. Par exemple, si vous avez un volume ZFS ou Solaris Volume Manager sur lequel vous avez déjà stocké des données et que vous voulez que votre domaine invité accède à ces données, vous devez exporter le volume ZFS ou Solaris Volume Manager à l’aide de l’option `slice`.

Pour plus d’informations sur cette option, reportez-vous à la section [“Arrière-plan du disque virtuel”](#) à la page 90.

Arrière-plan du disque virtuel

L’arrière-plan du disque virtuel est l’emplacement où les données d’un disque virtuel sont stockées. L’arrière-plan peut être un disque, un segment de disque, un fichier ou un volume tel que ZFS, Solaris Volume Manager ou VxVM. Un arrière-plan apparaît dans un domaine invité comme un disque complet ou un disque à segment unique, selon que l’option `slice` a été définie ou non lorsque l’arrière-plan a été exporté à partir du domaine de service. Par défaut, un arrière-plan de disque virtuel est exporté de manière non exclusive en tant que disque complet accessible en lecture et en écriture.

Disque physique ou LUN de disque

Un disque physique ou un LUN de disque est toujours exporté comme un disque complet. Dans ce cas, les pilotes de disque virtuel (`vds` et `vdc`) transmettent des E/S du disque virtuel et agissent comme une intercommunication vers le disque virtuel et le LUN de disque.

Un disque physique ou un LUN de disque est exporté à partir d'un domaine de service en exportant le périphérique qui correspond au segment 2 (s2) de ce disque sans définir l'option `slice`. Si vous exportez le segment 2 d'un disque avec l'option `slice`, seul ce segment est exporté et non pas le disque entier.

▼ Exportation d'un disque physique en tant que disque virtuel

1 Exportez un disque physique en tant que disque virtuel.

Par exemple, pour exporter le disque physique `c1t48d0` en tant que disque virtuel, vous devez exporter le segment 2 de ce disque (`c1t48d0s2`).

```
primary# ldm add-vdsdev /dev/dsk/c1t48d0s2 c1t48d0@primary-vds0
```

2 Assignez le disque à un domaine invité.

Par exemple, assignez le disque (`pdisk`) au domaine invité `ldg1`.

```
primary# ldm add-vdisk pdisk c1t48d0@primary-vds0 ldg1
```

3 Après que le domaine invité est démarré et exécute le SE Oracle Solaris, vérifiez que le disque est accessible et est un disque complet.

Un disque complet est un disque normal qui a huit (8) segments.

Par exemple, le disque en cours de vérification est `c0d1`.

```
ldg1# ls -l /dev/dsk/c0d1s*
/dev/dsk/c0d1s0
/dev/dsk/c0d1s1
/dev/dsk/c0d1s2
/dev/dsk/c0d1s3
/dev/dsk/c0d1s4
/dev/dsk/c0d1s5
/dev/dsk/c0d1s6
/dev/dsk/c0d1s7
```

Segment de disque physique

Un segment de disque physique est toujours exporté comme un disque à segment unique. Dans ce cas, les pilotes de disque virtuel (`vds` et `vdc`) transmettent des E/S du disque virtuel et agissent comme une intercommunication vers le segment de disque virtuel.

Un segment de disque physique est exporté d'un domaine de service en exportant le périphérique de segment correspondant. Si le périphérique est différent du segment 2, il est automatiquement exporté comme un disque à segment unique que vous indiquez ou non l'option `slice`. Si le périphérique est le segment 2 du disque, vous devez définir l'option `slice` pour exporter uniquement le segment 2 en tant que disque à segment unique. Sinon, le disque entier est exporté en tant que disque complet.

▼ Exportation d'un segment de disque physique en tant que disque virtuel

1 Exportez un segment de disque physique en tant que disque virtuel.

Par exemple, pour exporter le segment 0 d'un disque physique `c1t57d0` en tant que disque virtuel, vous devez exporter le périphérique qui correspond à ce segment (`c1t57d0s0`) comme suit.

```
primary# ldm add-vdsdev /dev/dsk/c1t57d0s0 c1t57d0s0@primary-vds0
```

Il est inutile de spécifier l'option `slice`, car un segment est toujours exporté comme un disque à segment unique.

2 Assignez le disque à un domaine invité.

Par exemple, assignez le disque (`pslice`) au domaine invité `ldg1`.

```
primary# ldm add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

3 Après que le domaine invité est démarré et exécute le SE Oracle Solaris, vous pouvez répertorier le disque (`c0d13`, par exemple) et voir que le disque est accessible.

```
ldg1# ls -l /dev/dsk/c0d13s*
/dev/dsk/c0d13s0
/dev/dsk/c0d13s1
/dev/dsk/c0d13s2
/dev/dsk/c0d13s3
/dev/dsk/c0d13s4
/dev/dsk/c0d13s5
/dev/dsk/c0d13s6
/dev/dsk/c0d13s7
```

Bien qu'il ait 8 périphériques, car le disque est un disque à segment unique, seul le premier segment (`s0`) est utilisable.

▼ Exportation du segment 2

- Pour exporter le segment 2 (disque `c1t57d0s2`, par exemple), vous devez spécifier l'option `slice`. Sinon, tout le disque est exporté.

```
# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

Fichier et volume

Un fichier ou un volume (par exemple provenant de ZFS ou Solaris Volume Manager) est exporté soit comme un disque complet, soit comme un disque à segment unique selon que vous définissez ou non l'option `slice`.

Fichier ou volume exporté en tant que disque complet

Si vous ne définissez pas l'option `slice`, un fichier ou un volume est exporté en tant que disque complet. Dans ce cas, les pilotes de disque virtuel (`vds` et `vdc`) transmettent les E/S à partir du disque virtuel et gèrent le partitionnement du disque virtuel. Le fichier ou le volume devient une image de disque contenant les données de tous les segments du disque virtuel et les métadonnées utilisées pour gérer le partitionnement et la structure du disque.

Si un fichier ou un volume vide est exporté en tant que disque complet, il apparaît dans le domaine invité comme un disque non formaté, c'est-à-dire un disque sans partition. Vous devez ensuite exécuter la commande `format(1M)` dans le domaine invité pour définir les partitions utilisables et pour écrire une étiquette de disque valide. Toutes les E/S vers le disque virtuel échouent lorsque le disque n'est pas formaté.

Remarque – Avant la version du SE Oracle Solaris 5/08, si un fichier vide était exporté en tant que disque virtuel, le système écrivait une étiquette de disque par défaut et créait un partitionnement par défaut. Ce n'est plus le cas avec la version du SE Oracle Solaris 5/08, et vous devez exécuter `format(1M)` dans le domaine invité pour créer les partitions.

▼ Exportation d'un fichier en tant que disque complet

- 1 À partir du domaine de service, créez un fichier (`fdisk0` par exemple) à utiliser comme disque virtuel.

```
service# mkfile 100m /ldoms/domain/test/fdisk0
```

La taille du fichier définit la taille du disque virtuel. Cet exemple crée un fichier vide de 100 méga-octets pour obtenir un disque virtuel de 100 méga-octets.

- 2 À partir du domaine de contrôle, exportez le fichier en tant que disque virtuel.

```
primary# ldm add-vdsdev /ldoms/domain/test/fdisk0 fdisk0@primary-vds0
```

Dans cet exemple, l'option `slice` n'est pas définie. Par conséquent, le fichier est exporté comme un disque complet.

- 3 À partir du domaine de contrôle, assignez le disque à un domaine invité.

Par exemple, assignez le disque (`fdisk`) au domaine invité `ldg1`.

```
primary# ldm add-vdisk fdisk fdisk0@primary-vds0 ldg1
```

- 4 Après que le domaine invité est démarré et exécute le SE Oracle Solaris, vérifiez que le disque est accessible et est un disque complet.

Un disque complet est un disque normal avec 8 segments.

L'exemple suivant montre comment répertorier le disque, `c0d5` et vérifier qu'il est accessible et est un disque complet.

```
ldg1# ls -l /dev/dsk/c0d5s*
/dev/dsk/c0d5s0
/dev/dsk/c0d5s1
/dev/dsk/c0d5s2
/dev/dsk/c0d5s3
/dev/dsk/c0d5s4
/dev/dsk/c0d5s5
/dev/dsk/c0d5s6
/dev/dsk/c0d5s7
```

Fichier ou volume exporté en tant que disque à segment unique

Si l'option `slice` est définie, le fichier ou le volume est exporté en tant que disque à segment unique. Dans ce cas, le disque virtuel n'a qu'une seule partition (`s0`), qui est directement mappée sur l'arrière-plan du fichier ou du volume. Le fichier ou le volume ne contient que des données écrites sur le disque virtuel avec données supplémentaires telles que des informations de partitionnement ou la structure du disque.

Si un fichier ou un volume est exporté en tant que disque à segment unique, le système simule un faux partitionnement de disque qui fait apparaître ce fichier ou ce volume en tant que segment de disque. Comme le partitionnement de disque est simulé, vous ne créez pas de partitionnement pour ce disque.

▼ Exportation d'un volume ZFS en tant que disque à segment unique

1 Créez un volume ZFS à utiliser en tant que disque à segment unique.

L'exemple suivant montre comment créer un volume ZFS, `zdisk0`, à utiliser comme disque à segment unique.

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

La taille du volume définit la taille du disque virtuel. Cet exemple crée un volume vide de 100 méga-octets pour obtenir un disque virtuel de 100 méga-octets.

2 À partir du domaine de contrôle, exportez le périphérique correspondant sur ce volume ZFS et définissez l'option `slice` afin que le volume soit exporté comme un disque à segment unique.

```
primary# ldm add-vdsdev options=slice /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

3 À partir du domaine de contrôle, assignez le volume à un domaine invité.

Voici comment assigner le volume, `zdisk0`, au domaine invité `ldg1`.

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

- 4 Après que le domaine invité est démarré et exécute le SE Oracle Solaris, vous pouvez répertorier le disque (c0d9, par exemple) et voir que le disque est accessible et est un disque à segment unique (s0).

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7
```

Exportation de volumes et rétrocompatibilité

Avant la version du SE Oracle Solaris 10 5/08, l'option `slice` n'existait pas et les volumes étaient exportés en tant que disques à segment unique. Si vous avez une configuration exportant les volumes en tant que disques virtuels et si vous mettez à niveau le système vers le SE Oracle Solaris 10 5/08, les volumes sont maintenant exportés comme des disques complets plutôt que des disques à segment unique. Pour préserver l'ancien comportement et que vos volumes soient exportés comme des disques à segment unique, vous devez effectuer l'une des opérations suivantes :

- Utilisez la commande `ldm set -vdsdev` dans le logiciel Oracle VM Server for SPARC 2.0 et définissez l'option `slice` pour tous les volumes que vous voulez exporter en tant que disques à segment unique. Reportez-vous à la page de manuel [ldm\(1M\)](#) pour plus d'informations sur cette commande.
- Ajoutez la ligne suivante au fichier `/etc/system` sur le domaine de service.

```
set vds:vd_volume_force_slice = 1
```

Remarque – Ce paramétrage force l'exportation de tous les volumes en tant que disques à segment unique et vous ne pouvez pas exporter de volume en tant que disque complet.

Récapitulatif des méthodes d'exportation des différents types de moteurs de traitement

Arrière-plan	Sans option de segmentation	Option de segmentation définie
Disque (segment de disque 2)	Disque complet ¹	Disque à segment unique ²
Segment de disque (pas le segment 2)	Disque à segment unique ³	Disque à segment unique
Fichier	Disque complet	Disque à segment unique

¹ Exportez tout le disque.

² Exportez uniquement le segment 2.

³ Un segment est toujours exporté en tant que disque à segment unique.

Arrière-plan	Sans option de segmentation	Option de segmentation définie
Volume, y compris ZFS, Solaris Volume Manager ou VxVM	Disque complet	Disque à segment unique

Consignes d'exportation des fichiers et des segments de fichiers en tant que disques virtuels

Cette section comprend des consignes pour l'exportation d'un fichier et d'un segment de disque en tant que disque virtuel.

Utilisation du pilote de fichier loopback (lofi)

Il est possible d'utiliser le pilote de fichier loopback (`lofi`) pour exporter un fichier en tant que disque virtuel. Cependant, cette opération ajoute une couche de pilote supplémentaire et a une incidence sur les performances du disque virtuel. Au lieu de cela, vous pouvez exporter directement un fichier en tant que disque complet ou en tant que disque à fragment unique. Reportez-vous à la section “Fichier et volume” à la page 92.

Exportation directe ou indirecte d'un segment de disque

Pour exporter un segment en tant que disque virtuel directement ou indirectement (par exemple via un volume Solaris Volume Manager), vérifiez que le segment ne commence pas au premier bloc (bloc 0) du disque physique à l'aide de la commande `prtvtoc(1M)`.

Si vous exportez directement ou indirectement un segment de disque qui commence sur le premier bloc d'un disque physique, vous risquez d'écraser la table des partitions du disque physique et de rendre toutes les partitions de ce disque inaccessibles.

Configuration du multivoie d'un disque virtuel

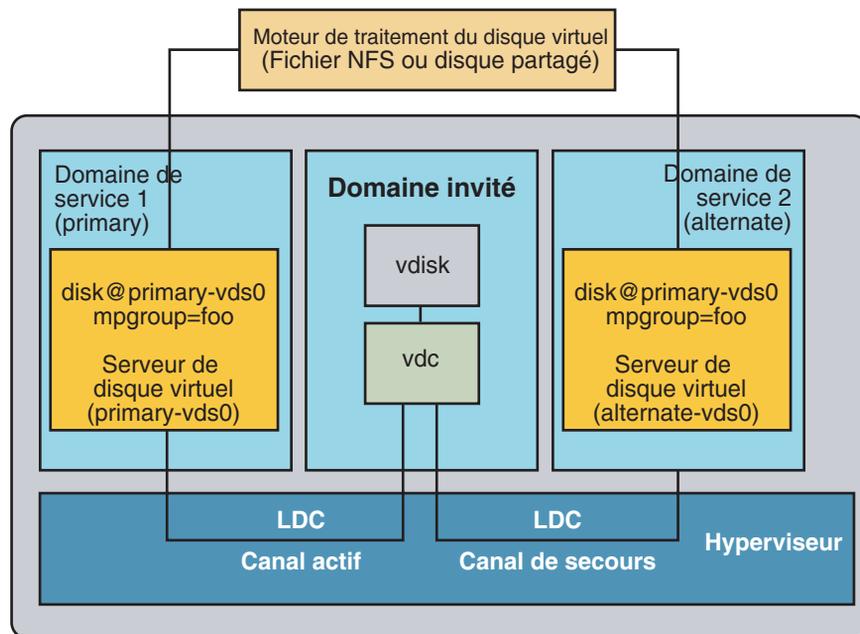
Le multivoie de disque virtuel vous permet de configurer un disque virtuel sur un domaine invité pour accéder à son stockage de moteur de traitement par plusieurs chemins. Les chemins traversent différents domaines de service qui fournissent l'accès au même stockage de moteur de traitement, notamment un LUN de disque. Cette fonction permet à un disque virtuel dans un domaine invité de rester accessible même si l'un des domaines de service est en panne. Par exemple, vous pouvez définir une configuration multivoie de disque virtuel pour accéder à un fichier ou à un serveur de système de fichier réseau (NFS). Vous pouvez également utiliser cette configuration pour accéder à un LUN à partir d'un stockage partagé qui est connecté à plusieurs domaines de service. Par conséquent, lorsque le domaine invité accède au disque virtuel, le pilote du disque virtuel traverse l'un des domaines de service pour accéder au stockage de moteur de traitement. Si le pilote du disque virtuel peut se connecter au domaine de service, le disque virtuel tente d'atteindre le stockage de moteur de traitement via un domaine de service différent.

Remarque – À partir de la version Oracle VM Server for SPARC 2.0, la fonction multivoie du disque virtuel peut détecter si un domaine de service ne peut pas accéder au stockage de moteur de traitement. Dans ce cas, le pilote du disque virtuel tente d'accéder au stockage de moteur de traitement par un autre chemin.

Pour permettre le multivoie au disque virtuel, vous devez exporter un arrière-plan de disque virtuel à partir de chaque domaine de service et ajouter le disque virtuel au même groupe multivoie (mpgroup). mpgroup est identifié par un nom et configuré lorsque vous exportez l'arrière-plan du disque virtuel.

La figure suivante représente une configuration multivoie de disque virtuel qui est utilisée comme un exemple dans la procédure [“Configuration du multivoie d'un disque virtuel” à la page 98](#). Dans cet exemple, un groupe multivoie nommé foo est utilisé pour créer un disque virtuel, dont l'arrière-plan est accessible à partir de deux domaines de service : primary et alternate.

FIGURE 7-2 Configuration du multivoie d'un disque virtuel



▼ Configuration du multivoie d'un disque virtuel

- 1 Exportez l'arrière-plan du disque virtuel à partir du domaine de service `primary`.

```
# ldm add-vdsdev mpgroup=foo backend-path1 volume@primary-vds0
```

où `backend-path1` est le chemin vers l'arrière-plan du disque virtuel à partir du domaine `primary`.

- 2 Exportez le même arrière-plan du disque virtuel à partir du domaine de service `alternate`.

```
# ldm add-vdsdev mpgroup=foo backend-path2 volume@alternate-vds0
```

où `backend-path2` est le chemin vers l'arrière-plan du disque virtuel à partir du domaine `alternate`.

Remarque – `backend-path1` et `backend-path2` sont les chemins vers le même arrière-plan de disque virtuel, mais à partir de deux domaines différents (`primary` et `alternate`). Ces chemins peuvent être identiques ou différents en fonction de la configuration des domaines `primary` et `alternate`. Le nom `volume` est un choix de l'utilisateur. Il peut être identique ou différent pour les deux commandes.

- 3 Exportez le disque virtuel sur le domaine invité.

```
# ldm add-vdisk disk-name volume@primary-vds0 ldom
```

Remarque – Bien que l'arrière-plan du disque virtuel soit exporté plusieurs fois via différents domaines de service, vous n'assignez qu'un seul disque virtuel au domaine invité et l'associez avec l'arrière-plan du disque virtuel via l'un des domaines de service.

Informations supplémentaires

Résultat du multivoie de disque virtuel

Après avoir configuré le disque virtuel avec le multivoie et démarré le domaine invité, le disque virtuel accède à son arrière-plan via le domaine de service auquel il a été associé (le domaine `primary` dans cet exemple). Si ce domaine de service devient indisponible, le disque virtuel essaie d'accéder à son arrière-plan via un domaine de service différent faisant partie du même groupe multivoie.



Attention – Lors de la définition d'un groupe multivoie (`mpgroup`), vérifiez que les moteurs de traitement de disque virtuel faisant partie du même `mpgroup` sont effectivement le même arrière-plan de disque virtuel. Si vous ajoutez des moteurs de traitement différents dans le même `mpgroup`, vous risquez de voir certains comportements inattendus et vous pouvez éventuellement perdre ou corrompre les données stockées sur les moteurs de traitement.

CD, DVD et images ISO

Vous pouvez exporter un CD ou un DVD de la même manière que vous exportez un disque normal. Pour exporter un CD ou un DVD vers un domaine invité, exportez le segment 2 du périphérique CD ou DVD en tant que disque complet, c'est-à-dire sans l'option `slice`.

Remarque – Vous ne pouvez pas exporter le lecteur de CD ou de DVD lui-même. Vous pouvez uniquement exporter le CD ou le DVD se trouvant dans le lecteur. Par conséquent, un CD ou un DVD doit se trouver dans le lecteur avant que vous ne puissiez l'exporter. Pour pouvoir exporter un CD ou un DVD, ceux-ci ne doivent pas être en cours d'utilisation dans le domaine de service. Plus précisément, le service du système de fichiers Volume Management, `volfs(7FS)` ne doit pas utiliser le CD ou le DVD. Reportez-vous à la section “[Exportation d'un CD ou d'un DVD à partir du domaine de service vers le domaine invité](#)” à la page 100 pour obtenir des instructions sur la procédure pour empêcher l'utilisation des périphériques par `volfs`.

Si vous disposez d'une image ISO (International Organisation for Standardization) d'un CD ou d'un DVD stockée dans un fichier ou sur un volume et que vous exportez ce fichier ou volume en tant que disque complet, il apparaît comme un CD ou un DVD dans le domaine invité.

Lorsque vous exportez un CD, un DVD ou une image ISO, il apparaît automatiquement comme un périphérique en lecture seule sur le domaine invité. Cependant, vous ne pouvez pas effectuer d'opérations de contrôle du CD à partir du domaine invité, c'est-à-dire que vous ne pouvez pas démarrer, arrêter ou éjecter le CD du domaine invité. Si l'image ISO, le CD ou le DVD exporté est initialisable, le domaine invité peut être initialisé sur le disque virtuel correspondant.

Par exemple, si vous exportez le DVD d'installation du SE Oracle Solaris, vous pouvez démarrer le domaine invité sur le disque virtuel qui correspond à ce DVD et installer le domaine invité à partir de ce DVD. Pour ce faire, lorsque le domaine invité atteint l'invite `ok`, utilisez la commande suivante.

```
ok boot /virtual-devices@100/channel-devices@200/disk@n:f
```

Où `n` est l'index du disque virtuel représentant le DVD exporté.

Remarque – Si vous exportez un DVD d'installation du SE Oracle Solaris et démarrez un domaine invité sur le disque virtuel qui correspond à ce DVD pour installer le domaine invité; vous ne pouvez pas changer le DVD au cours de l'installation. Vous devrez peut-être sauter les étapes d'installation demandant un CD/DVD différent ou vous devrez fournir un autre chemin pour accéder au support demandé.

▼ Exportation d'un CD ou d'un DVD à partir du domaine de service vers le domaine invité

- 1 À partir du domaine de service, vérifiez si le démon de gestion des volumes `volD(1M)`, est en cours d'exécution et en ligne.

```
service# svcs volfs
STATE          STIME      FMRI
online         12:28:12  svc:/system/filesystem/volfs:default
```

- 2 Effectuez l'une des opérations suivantes.

- Si le volume de gestion des volumes n'est pas en cours d'exécution ou en ligne, passez à l'étape 3.
- Si le démon de gestion des volumes est en cours d'exécution et en ligne, comme dans l'exemple de l'étape 1, procédez comme suit :

- a. Éditez le fichier `/etc/vold.conf` et commentez la ligne commençant par les mots suivants.

```
use cdrom drive...
```

Reportez-vous à la page de manuel `vold.conf(4)`.

- b. Insérez le CD ou le DVD dans le lecteur de CD ou de DVD.

- c. À partir du domaine de service, redémarrez le service du système de fichiers de gestion des volumes.

```
service# svcadm refresh volfs
service# svcadm restart volfs
```

- 3 À partir du domaine de service, recherchez le chemin de disque pour le périphérique CD-ROM.

```
service# cdrw -l
Looking for CD devices...
  Node                               Connected Device                               Device type
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
/dev/rdisk/c1t0d0s2 | MATSHITA CD-RW CW-8124 DZ13 | CD Reader/Writer
```

- 4 Exportez le périphérique de CD ou de DVD comme un disque complet.

```
primary# ldm add-vdsdev /dev/dsk/c1t0d0s2 cdrom@primary-vds0
```

- 5 Assignez le CD ou le DVD exporté au domaine invité.

Voici comment assigner le CD ou le DVD exporté au domaine `ldg1` :

```
primary# ldm add-vdisk cdrom cdrom@primary-vds0 ldg1
```

**Informations
supplémentaires****Exportation multiple d'un CD ou d'un DVD**

Un CD ou un DVD peut être exporté plusieurs fois et assigné à des domaines invités différents. Reportez-vous à la section “[Exportation multiples de l'arrière-plan du disque virtuel](#)” à la page 85.

▼ Exportation d'une image ISO à partir du domaine primary pour installation sur un domaine invité

Cette procédure montre comment exporter une image ISO à partir du domaine primary et l'installer sur un domaine invité. Cette procédure suppose que le domaine primary et le domaine invité soient configurés.

Par exemple, `ldm list` montre que les deux domaines primary et `ldg1` sont configurés :

```
# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    4     4G      0.3%  15m
ldom1        active -t---  5000  4     1G      25%   8m
```

1 Ajoutez un périphérique de serveur de disque virtuel pour exporter l'image ISO.

Dans cet exemple, l'image ISO est `/export/images/so1-10-u8-ga-sparc-dvd.iso`.

```
# ldm add-vdsdev /export/images/so1-10-u8-ga-sparc-dvd.iso dvd-iso@primary-vds0
```

2 Arrêtez le domaine invité.

Dans cet exemple, le domaine logique est `ldom1`.

```
# ldm stop-domain ldom1
LDom ldom1 stopped
```

3 Ajoutez le disque virtuel pour l'image ISO sur le domaine logique.

Dans cet exemple, le domaine logique est `ldom1`.

```
# ldm add-vdisk s10-dvd dvd-iso@primary-vds0 ldom1
```

4 Redémarrez le domaine invité.

Dans cet exemple, le domaine logique est `ldom1`.

```
# ldm start-domain ldom1
LDom ldom1 started
# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    4     4G      0.4%  25m
ldom1        active -t---  5000  4     1G      0.0%  0s
```

Dans cet exemple, la commande `ldm list` affiche que le domaine `ldom1` vient d'être démarré.

5 Connectez-vous au domaine invité.

```
# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

```
Connecting to console "ldom1" in group "ldom1" ....
Press ~? for control options ..
```

6 Vérifiez l'existence de l'image ISO en tant que disque virtuel.

```
{0} ok show-disks
a) /virtual-devices@100/channel-devices@200/disk@1
b) /virtual-devices@100/channel-devices@200/disk@0
q) NO SELECTION
Enter Selection, q to quit: q
```

Dans cet exemple, le périphérique nouvellement ajouté est
/virtual-devices@100/channel-devices@200/disk@1.

7 Démarrez le domaine invité pour effectuer l'installation à partir de l'image ISO.

Dans cet exemple, démarrez à partir du segment f du disque
/virtual-devices@100/channel-devices@200/disk@1.

```
{0} ok boot /virtual-devices@100/channel-devices@200/disk@1:f
```

Délai d'attente du disque virtuel

Par défaut, si le domaine de service fournissant l'accès à l'arrière-plan d'un disque virtuel est en panne, toutes les E/S du domaine invité vers le disque virtuel correspondant sont bloquées. Les E/S reprennent automatiquement lorsque le domaine de service est opérationnel et traite les demandes d'E/S vers l'arrière-plan du disque virtuel.

Cependant, il existent certains cas dans lesquels les systèmes de fichiers ou les applications peuvent ne pas vouloir que l'opération d'E/S se bloque, mais échoue et signale une erreur lorsque le domaine de service est en panne pour une longue période. Il est maintenant possible de définir une période de délai d'attente de connexion pour chaque disque virtuel, qui peut ensuite être utilisée pour établir une connexion entre le client de disque virtuel sur un domaine invité et le serveur de disque virtuel sur le domaine de service. Lorsque cette période de délai d'attente est atteinte, toutes les E/S en attente ou toutes les nouvelles E/S échoueront tant que le domaine de service est en panne et que la connexion entre le client de disque virtuel et le serveur n'est pas rétablie.

Ce délai d'attente peut être défini en procédant comme suit :

- À l'aide de la commande `ldm add-vdisk`.


```
ldm add-vdisk timeout=seconds disk-name volume-name@service-name ldom
```
- À l'aide de la commande `ldm set-vdisk`.

```
ldm set-vdisk timeout=seconds disk-name ldom
```

Indiquez le délai d'attente en secondes. Si le délai d'attente est défini sur 0, il est désactivé et les E/S sont bloquées pendant la panne du domaine de service (il s'agit du paramètre et du comportement par défaut).

Sinon, le délai d'attente peut être défini en ajoutant la ligne suivante au fichier `/etc/system` sur le domaine invité.

```
set vdc:vdc_timeout=seconds
```

Remarque – Si cette option est définie, elle remplace le paramètre de délai d'attente défini à l'aide de la CLI `ldm`. Cette option définit également le délai d'attente pour tous les disques virtuels dans le domaine invité.

Disque virtuel et SCSI

Si un disque SCSI physique ou un LUN est exporté en tant que disque complet, le disque virtuel correspondant prend en charge l'interface de commande SCSI utilisateur, `uscsci(7I)` et les opérations de contrôle des risques multihôtes `mhd(7i)`. Les autres disques virtuels, notamment les disques virtuels ayant un fichier ou un volume comme arrière-plan, ne prennent pas en charge ces interfaces.

En conséquence, les applications ou fonctions de produit utilisant les commandes SCSI (notamment `metaset` de Solaris Volume Manager ou `shared devices cluster Oracle Solaris`) peuvent être utilisés dans les domaines invités uniquement avec les disques virtuels ayant un disque SCSI physique comme arrière-plan.

Remarque – Les opérations SCSI sont effectivement exécutées par le domaine de service qui gère le disque SCSI physique ou le LUN utilisé comme moteur de traitement du disque virtuel. Les réservations SCSI sont effectuées par le domaine de service. Par conséquent, les applications s'exécutant dans le domaine de service et dans les domaines invités ne doivent pas émettre de commandes SCSI vers les mêmes disques SCSI physiques. Sinon, cela peut aboutir à un état de disque inattendu.

Disque virtuel et commande `format(1M)`

La commande `format(1M)` reconnaît tous les disques virtuels présents dans un domaine. Cependant, pour les disques virtuels exportés en tant que disques à segment unique, la commande `format` ne peut pas modifier la table des partitions du disque virtuel. Les commandes telles que `label` échoueront à moins que vous ne tentiez d'écrire une étiquette de disque semblable à celle qui est déjà associée au disque virtuel.

Les disques virtuels dont les moteurs de traitement sont des disques SCSI prennent en charge toutes les sous-commandes `format(1M)`. Les disques virtuels dont les moteurs de traitement ne sont pas des disques SCSI ne prennent pas en charge certaines des sous-commandes `format(1M)`, notamment `repair` et `defect`. Dans ce cas, le comportement de `format(1M)` est semblable au comportement des disques Integrated Drive Electronics (IDE).

Utilisation de ZFS avec les disques virtuels

Cette section décrit l'utilisation du système de fichiers Zettabyte (ZFS) pour stocker des moteurs de traitement de disque virtuel exportés sur des domaines invités. ZFS fournit une solution pratique et puissante pour créer et gérer des moteurs de traitement de disque virtuel. ZFS permet :

- de stocker des images de disque dans des volumes ZFS ou des fichiers ZFS ;
- d'utiliser des instantanés pour sauvegarder les images de disque ;
- d'utiliser des clones pour dupliquer les images de disque et mettre à disposition d'autres domaines.

Reportez-vous au [Guide d'administration Oracle Solaris ZFS](#) pour plus d'informations sur l'utilisation de ZFS.

Dans les descriptions et les exemples suivants, le domaine principal est également le domaine de service où les images de disque sont stockées.

Configuration d'un pool ZFS dans un domaine de service

Pour stocker des images de disque, créez d'abord un pool de stockage ZFS dans le domaine de service. Par exemple, cette commande crée le pool de stockage `ldmpool` contenant le disque `c1t50d0` dans le domaine `primary`.

```
primary# zpool create ldmpool c1t50d0
```

Stockage des images de disque avec ZFS

La commande suivante crée une image de disque pour le domaine invité `ldg1`. Un système de fichiers ZFS pour ce domaine invité est créé et toutes les images de disque de ce domaine invité seront stockés sur ce système de fichiers.

```
primary# zfs create ldmpool/ldg1
```

Les images de disque peuvent être stockées sur les volumes ZFS ou les fichiers ZFS. La création d'un volume ZFS, quelle que soit sa taille, est rapide à l'aide de la commande `zfs create -V`. D'autre part, les fichiers ZFS doivent être créés à l'aide de la commande `mkfile`. La commande peut prendre un certain temps pour se terminer, surtout si le fichier à créer est très volumineux, ce qui est souvent le cas lors de la création d'une image de disque.

Les volumes ZFS et les fichiers ZFS peuvent tirer parti des fonctions ZFS telles que les fonctions d'instantané et de clone, mais un volume ZFS est un pseudo-périphérique tandis qu'un fichier ZFS est un fichier normal.

Si l'image de disque doit être utilisée comme disque virtuel sur lequel un SE est installé, celle-ci doit être de taille suffisante pour s'adapter aux contraintes d'installation du SE. La taille dépend de la version du SE et du type d'installation effectuée. Si vous installez le SE Oracle Solaris, vous pouvez utiliser une taille de disque de 20 giga-octets pour accueillir tout type d'installation de n'importe quelle version du SE Oracle Solaris.

Exemples de stockage d'images de disque avec ZFS

Les exemples suivants :

1. Créent une image de 20 giga-octets sur un volume ou un fichier ZFS.
2. Exportent le volume ou le fichier ZFS en tant que disque virtuel. La syntaxe pour exporter un volume ou un fichier ZFS est identique, mais le chemin au moteur de traitement est différent.
3. Assignent le volume ou le fichier ZFS exporté à un domaine invité.

Lorsque le domaine invité est démarré, le volume ou le fichier ZFS apparaît comme un disque virtuel sur lequel le SE Oracle Solaris peut être installé.

▼ Création d'une image de disque à l'aide d'un volume ZFS

- Par exemple, créez une image de disque de 20 giga-octets sur un volume ZFS.

```
primary# zfs create -V 20gb ldmpool/ldg1/disk0
```

▼ Création d'une image de disque à l'aide d'un fichier ZFS

- Par exemple, créez une image de disque de 20 giga-octets sur un volume ZFS.

```
primary# zfs create ldmpool/ldg1/disk0
primary# mkfile 20g /ldmpool/ldg1/disk0/file
```

▼ Exportation du volume ZFS

- Exportez le volume ZFS en tant que disque virtuel.

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldmpool/ldg1/disk0 ldg1_disk0@primary-vds0
```

▼ Exportation du fichier ZFS

- Exportez le fichier ZFS en tant que disque virtuel.

```
primary# ldm add-vdsdev /ldmpool/ldg1/disk0/file ldg1_disk0@primary-vds0
```

▼ Assignation du volume ZFS à un domaine invité

- Assignez le volume ou le fichier ZFS à un domaine invité, dans cet exemple ldg1.

```
primary# ldm add-vdisk disk0 ldg1_disk0@primary-vds0 ldg1
```

Création d'un instantané d'une image de disque

Lorsque votre image de disque est stockée sur un volume ZFS ou un fichier ZFS, vous pouvez créer des instantanés de cette image de disque à l'aide de la commande d'instantané ZFS.

Avant de créer un instantané de l'image de disque, vérifiez que le disque n'est pas actuellement utilisé dans le domaine invité pour garantir que les données actuellement stockées sur l'image de disque sont cohérentes. Il y a plusieurs méthodes pour vérifier qu'un disque n'est pas en cours d'utilisation dans un domaine invité. Vous pouvez :

- Arrêter et dissocier le domaine invité. C'est la solution la plus sûre et c'est la seule solution disponible si vous voulez créer un instantané d'une image de disque utilisée comme disque de démarrage d'un domaine invité.
- Vous pouvez également démonter les segments du disque dont vous souhaitez créer un instantané dans le domaine invité et vérifier qu'aucun segment n'est utilisé dans le domaine invité.

Dans cet exemple, en raison de la disposition ZFS, la commande pour créer un instantané de l'image de disque est identique que l'image de disque soit stockée sur un volume ZFS ou un fichier ZFS.

▼ Création d'un instantané d'une image de disque

- Créez un instantané de l'image de disque qui a été créée pour le domaine `ldg1`, par exemple.

```
primary# zfs snapshot ldmpool/ldg1/disk0@version_1
```

Utilisation du clone pour mettre à disposition un nouveau domaine

Un fois que vous avez créé un instantané de l'image de disque, vous pouvez dupliquer cette image de disque à l'aide de la commande de clonage ZFS. L'image clonée est ensuite assignée à un autre domaine. Le clonage d'une image de disque d'initialisation crée un disque d'initialisation pour un nouveau domaine invité sans avoir à effectuer tout le processus d'installation du SE Oracle Solaris.

Par exemple, si le `disk0` créé était le disque d'initialisation du domaine `ldg1`, procédez comme suit pour cloner ce disque pour créer un disque d'initialisation pour le domaine `ldg2`.

```
primary# zfs create ldmpool/ldg2
primary# zfs clone ldmpool/ldg1/disk0@version_1 ldmpool/ldg2/disk0
```

Ensuite, `ldmpool/ldg2/disk0` peut être exporté comme disque virtuel et assigné au nouveau domaine `ldg2`. Le domaine `ldg2` peut s'initialiser directement à partir de ce disque virtuel sans devoir suivre toute le processus d'installation du SE.

Clonage d'une image de disque d'initialisation

Si une image de disque d'initialisation est clonée, la nouvelle image est totalement identique au disque d'initialisation d'origine, et elle contient toutes les informations ayant été stockées sur le disque d'initialisation avant que l'image ne soit clonée, notamment le nom d'hôte, l'adresse IP, la table du système de fichiers monté ou la configuration et les paramètres du système.

Comme la table du système de fichiers montée est identique sur l'image du disque d'initialisation d'origine et sur l'image de disque clonée, l'image de disque clonée doit être assignée au nouveau domaine dans le même ordre que celui du domaine d'origine. Par exemple, si l'image du disque d'initialisation a été assignée comme premier disque du domaine d'origine, l'image de disque clonée doit être assignée comme premier disque du nouveau domaine. Sinon, le nouveau domaine ne peut pas s'initialiser.

Si le domaine d'origine a été configuré avec une adresse IP statique, un nouveau domaine utilisant l'image clonée comment avec la même adresse IP statique. Dans ce cas, vous pouvez modifier la configuration réseau du nouveau domaine à l'aide de la commande `sys-unconfig(1M)`. Pour éviter ce problème, vous pouvez également créer un instantané d'une image de disque du système non configuré.

Si le domaine d'origine était configuré avec le protocole DHCP (Dynamic Host Configuration Protocol), un nouveau domaine utilisant l'image clonée utilise également le protocole DHCP. Dans ce cas, il est inutile de modifier la configuration réseau du nouveau domaine, car il reçoit automatiquement une adresse IP et sa configuration réseau lorsqu'il s'initialise.

Remarque – L'ID hôte d'un domaine n'est pas stocké sur le disque d'initialisation, mais est assigné par le gestionnaire de domaines logiques lorsque vous créez un domaine. Par conséquent, lorsque vous clonez une image de disque, le nouveau domaine ne conserve pas l'ID hôte du domaine d'origine.

▼ **Création d'un instantané d'une image de disque d'un système non configuré**

- 1 Associez et démarrez le domaine d'origine.
- 2 Exécutez la commande `sys-unconfig`.
- 3 À la fin de la commande `sys-unconfig`, le domaine s'arrête.
- 4 Arrêtez et dissociez le domaine. Ne le redémarrez pas.
- 5 Prenez un instantané de l'image du disque d'initialisation du domaine.

Par exemple :

```
primary# zfs snapshot ldmpool/ldg1/disk0@unconfigured
```

À ce moment, vous avez l'instantané de l'image du disque d'initialisation d'un système non configuré.

- 6 Clonez cette image pour créer un nouveau domaine qui, au premier démarrage, demande la configuration du système.

Utilisation du gestionnaire de volumes dans un environnement de domaines logiques

Cette section décrit l'utilisation des gestionnaires de volumes dans un environnement de domaines logiques.

Utilisation de disques virtuels au sommet de gestionnaires de volumes

Tous les volumes ZFS, Solaris Volume Manager ou Veritas Volume Manager (VxVM) peuvent être exportés d'un domaine de service vers un domaine invité en tant que disque virtuel. Un volume peut être exporté comme un disque à segment unique (si l'option `slice` est définie avec la commande `ldm add-vdsdev`) ou en tant que disque complet.

Remarque – Le reste de cette section utilise un volume Solaris Volume Manager comme exemple. Cependant, la section s'applique également aux volumes ZFS et VxVM.

Les exemples suivants montrent comment exporter un volume en tant que disque à segment unique.

Le disque virtuel du domaine invité (par exemple, `/dev/dsk/c0d2s0`) est directement mappé sur le volume associé (par exemple, `/dev/md/dsk/d0`), et les données stockées sur le disque virtuel du domaine invité sont directement stockées sur le volume associé sans métadonnées supplémentaires. Par conséquent, les données stockées sur le disque virtuel du domaine invité sont accessibles directement à partir du domaine de service par le volume associé.

Exemples

- Si le volume Solaris Volume Manager `d0` est exporté du domaine `primary` sur `domain1`, la configuration de `domain1` nécessite des étapes supplémentaires.

```
primary# metainit d0 3 1 c2t70d0s6 1 c2t80d0s6 1 c2t90d0s6
primary# ldm add-vdsdev options=slice /dev/md/dsk/d0 vol3@primary-vds0
primary# ldm add-vdisk vdisk3 vol3@primary-vds0 domain1
```

- Après que `domain1` a été associé et démarré, le volume exporté apparaît comme `/dev/dsk/c0d2s0`, par exemple, et vous pouvez l'utiliser.

```
domain1# newfs /dev/rdisk/c0d2s0
domain1# mount /dev/dsk/c0d2s0 /mnt
domain1# echo test-domain1 > /mnt/file
```

- Après que `domain1` a été arrêté et dissocié, les données stockées sur le disque virtuel de `domain1` sont accessibles directement à partir du domaine principal par le volume Solaris Volume Manager `d0`.

```
primary# mount /dev/md/dsk/d0 /mnt
primary# cat /mnt/file
test-domain1
```

Utilisation de disques virtuels au sommet de Solaris Virtual Manager

Si un volume RAID ou Solaris Volume Manager en miroir est utilisé comme disque virtuel par un autre domaine, il doit être exporté sans définir l'option d'exclusivité (`excl`). Sinon, s'il y a une panne de l'un des composants du volume Solaris Volume Manager, la récupération du volume

Solaris Volume Manager à l'aide de la commande `metareplace` ou à l'aide d'un disque hot spare ne démarre pas. La commande `metastart` voit le volume comme étant en resynchronisation, mais la resynchronisation ne progresse pas.

Par exemple, `/dev/md/dsk/d0` est un volume RAID Solaris Volume Manager exporté en tant que disque virtuel avec l'option `excl` sur un autre domaine et `d0` est configuré avec des périphériques hot spare. Si un composant de `d0` échoue, Solaris Volume Manager remplace le composant défaillant par un disque hot spare et resynchronise le volume Solaris Volume Manager. Cependant, la resynchronisation ne démarre pas. Le volume est signalé comme étant en resynchronisation, mais la resynchronisation ne progresse pas.

```
# metastat d0
d0: RAID
  State: Resyncing
  Hot spare pool: hsp000
  Interlace: 32 blocks
  Size: 20097600 blocks (9.6 GB)
Original device:
  Size: 20100992 blocks (9.6 GB)
Device                               Start Block  Dbase   State Reloc
c2t2d0s1                               330         No      Okay   Yes
c4t12d0s1                               330         No      Okay   Yes
/dev/dsk/c10t600C0FF00000000000015153295A4B100d0s1 330         No      Resyncing Yes
```

Dans ce cas, le domaine utilisant le volume Solaris Volume Manager en tant que disque virtuel doit être arrêté et dissocié pour terminer la resynchronisation. Ensuite, le volume Solaris Volume Manager peut être resynchronisé à l'aide de la commande `metasync`.

```
# metasync d0
```

Utilisation des disques virtuels lorsque VxVM est installé

Lorsque Veritas Volume Manager (VxVM) est installé sur votre système, et si Veritas Dynamic Multipathing (DMP) est activé sur un disque ou une partition physique que vous voulez exporter en tant que disque virtuel, vous devez exporter ce disque ou cette partition sans définir l'option `excl` (valeur l'étant pas celle par défaut). Sinon, vous recevez une erreur dans `/var/adm/messages` lors de l'association du domaine utilisant un tel disque.

```
vd_setup_vd(): ldi_open_by_name(/dev/dsk/c4t12d0s2) = errno 16
vds_add_vd(): Failed to add vdisk ID 0
```

Vous pouvez vérifier si Veritas DMP est activé en vérifiant les informations sur le multivoie dans la sortie de la commande `vxdisk list`, par exemple :

```
# vxdisk list Disk_3
Device:      Disk_3
devicetag:  Disk_3
type:       auto
info:       format=none
flags:      online ready private autoconfig invalid
```

```
pubpaths: block=/dev/vx/dmp/Disk_3s2 char=/dev/vx/rdmp/Disk_3s2
guid: -
udid: SEAGATE%5FST336753LSUN36G%5FDISKS%5F3032333948303144304E0000
site: -
Multipathing information:
numpaths: 1
c4t12d0s2 state=enabled
```

Si non, si Veritas DMP est activé sur un disque ou un segment que vous voulez exporter en tant que disque virtuel avec l'option `excl` définie, vous pouvez désactiver le DMP à l'aide de la commande `vxmpadm`. Par exemple :

```
# vxmpadm -f disable path=/dev/dsk/c4t12d0s2
```

Utilisation des gestionnaires de volumes au sommet de disques virtuels

Cette section décrit l'utilisation des gestionnaires de volumes au sommet des disques virtuels.

Utilisation de ZFS au sommet de disques virtuels

Tous les disques virtuels peuvent être utilisés avec ZFS. Un pool de stockage ZFS (`zpool`) peut être importé dans un domaine qui voit tous les périphériques de stockage qui font partie de ce `zpool`, que ces domaines voient tous ces périphériques comme des périphériques virtuels ou des périphériques réels.

Utilisation du gestionnaire de volumes au sommet de disques virtuels

Tout disque virtuel peut être utilisé dans l'ensemble de disques locaux Solaris Volume Manager. Par exemple, un disque virtuel peut être utilisé pour stocker la base de données d'état des métapériphériques Solaris Volume Manager, `metadb(1M)`, de l'ensemble de disques locaux ou pour créer les volumes Solaris Volume Manager dans l'ensemble de disques locaux.

Tout disque virtuel dont le moteur de traitement est un disque SCSI peut être utilisé dans un ensemble de disques partagés Solaris Volume Manager, `metaset(1M)`. Les disques virtuels dont les moteurs de traitement ne sont pas des disques SCSI ne peuvent pas être ajoutés dans un ensemble de disques partagés Solaris Volume Manager. Toute tentative pour ajouter un disque virtuel dont le moteur de traitement n'est pas un disque SCSI dans un ensemble de disques partagés Solaris Volume Manager échouera avec une erreur semblable à la suivante.

```
# metaset -s test -a c2d2
metaset: domain1: test: failed to reserve any drives
```

Utilisation de VxVM au sommet de disques virtuels

Pour la prise en charge de VxVM dans les domaines invités, reportez-vous à la documentation VxVM de Symantec.

Utilisation des réseaux virtuels

Ce chapitre décrit comment utiliser un réseau virtuel avec le logiciel Oracle VM Server for SPARC et aborde les rubriques suivantes :

- “Introduction au réseau virtuel” à la page 113
- “Commutateur virtuel” à la page 114
- “Périphérique réseau virtuel” à la page 114
- “Gestion d’un commutateur virtuel” à la page 116
- “Gestion d’un périphérique réseau virtuel” à la page 118
- “Identificateur de périphérique virtuel et nom d’interface réseau” à la page 120
- “Assignation automatique et manuelle des adresses MAC” à la page 122
- “Utilisation des adaptateurs réseau avec les domaines logiques” à la page 125
- “Configuration du commutateur virtuel et du domaine de service pour le NAT et le routage” à la page 126
- “Configuration d’IPMP dans un environnement de domaines logiques” à la page 128
- “Utilisation du balisage VLAN” à la page 135
- “Utilisation des E/S hybrides NIU” à la page 139
- “Utilisation du groupement de liens avec un commutateur virtuel” à la page 142
- “Configuration des cadres Jumbo” à la page 144

Introduction au réseau virtuel

Un réseau virtuel permet aux domaines de communiquer les uns avec les autres sans utiliser de réseaux physiques externes. Un réseau virtuel peut également autoriser les domaines à utiliser la même interface réseau physique pour accéder à un réseau physique et communiquer avec des systèmes distants. Un réseau virtuel est créé en ayant un commutateur virtuel auquel vous pouvez connecter des périphériques réseau virtuels.

Commutateur virtuel

Un commutateur virtuel (*vsw*) est un composant s'exécutant dans un domaine de service et géré par le pilote du commutateur virtuel. Un commutateur virtuel peut être connecté à certains domaines invités pour permettre les communications réseau entre ces domaines. Par ailleurs, si le commutateur virtuel est associé également à une interface réseau physique, les communications réseau entre les domaines invités et le réseau physique sont possibles via l'interface réseau physique. Un commutateur virtuel a également une interface réseau, *vswm*, qui permet au domaine de service de communiquer avec les autres domaines connectés à ce commutateur virtuel. Il peut être utilisé comme une interface réseau classique et configuré avec la commande `ifconfig(1M)`.

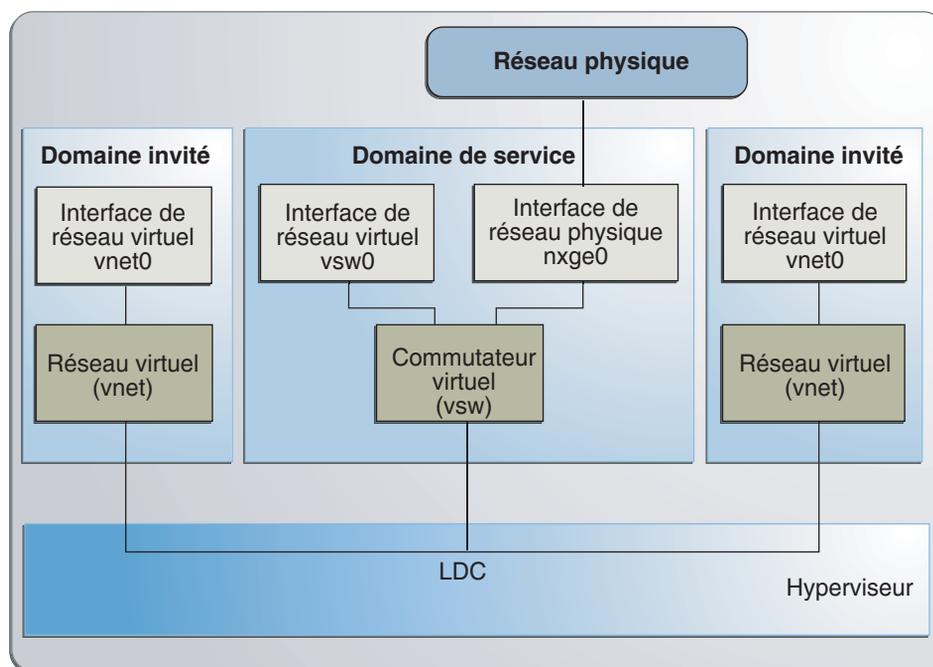
Remarque – Lorsqu'un commutateur virtuel est ajoutée au domaine de service, son interface réseau n'est pas plombée. Par conséquent, le domaine de service est incapable par défaut de communiquer avec les domaines invités connectés à son commutateur virtuel. Pour activer les communications réseau entre les domaines invités et le domaine de service, l'interface réseau du commutateur virtuel associé doit être plombée et configurée dans le domaine de service. Reportez-vous à la section [“Activation de la mise en réseau entre le domaine de contrôle/service et les autres domaines”](#) à la page 53 pour obtenir des instructions.

Périphérique réseau virtuel

Un périphérique réseau virtuel (*vnet*) est un périphérique virtuel qui est défini dans un domaine connecté à un commutateur virtuel. Un périphérique réseau virtuel est géré par le pilote de réseau virtuel et il est connecté à un réseau virtuel par l'hyperviseur à l'aide des canaux de domaines logiques (LDC).

Un périphérique réseau virtuel peut être utilisé en tant qu'interface réseau avec le nom *vnet**n*, qui peut être utilisée comme n'importe quelle interface réseau classique et configurée avec la commande `ifconfig(1M)`.

FIGURE 8-1 Configuration d'un réseau virtuel



Vous trouverez ci-dessous une explication de l'exemple de la [Figure 8-1](#).

- Le commutateur virtuel du domaine de service est connecté aux domaines invités. Cela permet aux domaines invités de communiquer entre eux.
- Le commutateur virtuel est également connecté à l'interface réseau physique `nxge0`. Cela permet aux domaines invités de communiquer avec le réseau physique.
- L'interface réseau du commutateur virtuel `vsw0` est plombée dans le domaine de service de façon à permettre aux deux domaines invités de communiquer avec le domaine de service.
- L'interface réseau du commutateur virtuel `vsw0` dans le domaine de service peut être configurée à l'aide de la commande `ifconfig(1M)`.
- Les interfaces de réseau virtuel `vnet0` des domaines invités peuvent être configurées à l'aide de la commande `ifconfig(1M)`.

À la base, le commutateur virtuel se comporte comme un commutateur de réseau physique classique et commute les paquets du réseau entre les différents systèmes, tels que les domaines invités et le réseau physique auquel il est connecté.

Gestion d'un commutateur virtuel

Cette section décrit l'ajout d'un commutateur virtuel à un domaine, la définition des options pour un commutateur virtuel et la suppression d'un commutateur virtuel.

▼ Ajout d'un commutateur virtuel

- Utilisez la syntaxe de commande suivante pour ajouter un commutateur virtuel.

```
# ldm add-vsw [default-vlan-id=vlan-id] [pvid=[port-vlan-id]] [vid=vlan-id1,vlan-id2,...]
  [linkprop=phys-state] [mac-addr=num] [net-dev=device] [mode=sc] [mtu=size]
  [id=switch-id] vswitch-name ldom
```

où :

- `default-vlan-id=id-vlan` indique le réseau local virtuel (VLAN) par défaut auquel un commutateur virtuel et ses périphériques réseau virtuels associés appartiennent implicitement en mode non balisé. Il sert d'ID VLAN de port par défaut (pvid) du commutateur virtuel et des périphériques réseau virtuels. Sans cette option, la valeur par défaut de cette propriété est 1. Normalement, vous ne devez pas utiliser cette option. Elle est fournie uniquement comme méthode pour modifier la valeur par défaut de 1. Voir [“Utilisation du balisage VLAN” à la page 135](#) pour plus d'informations.
- `pvid=id-vlan-port` définit le VLAN dont le commutateur virtuel doit être membre en mode non balisé. Voir [“Utilisation du balisage VLAN” à la page 135](#) pour plus d'informations.
- `vid=vlan-id` définit un ou plusieurs VLAN dont un commutateur virtuel doit être membre en mode balisé. Voir [“Utilisation du balisage VLAN” à la page 135](#) pour plus d'informations.
- `linkprop=phys-state` indique si le périphérique virtuel signale son état de liaison en fonction du périphérique réseau physique sous-jacent. Lorsque `linkprop=phys-state` est défini dans une ligne de commande, l'état de liaison du périphérique virtuel reflète l'état de liaison physique. Par défaut, l'état de liaison du périphérique virtuel ne reflète pas l'état de liaison physique.

Définissez cette option pour utiliser l'IPMP basé sur la liaison. Voir [“Utilisation de l'IPMP basé sur liaison dans la mise en réseau virtuelle des domaines logiques” à la page 130](#)

- `mac-addr=num` est l'adresse MAC à utiliser pour ce commutateur. Le numéro doit être en notation d'octet standard, par exemple, 80:00:33:55:22:66. Si vous ne définissez d'adresse MAC, une adresse dans la plage d'adresses MAC publiques allouées au gestionnaire de domaines logiques est automatiquement assignée au commutateur. Voir [“Assignation automatique et manuelle des adresses MAC” à la page 122](#) pour plus d'informations.
- `net-dev=périphérique` est le chemin au périphérique réseau sur lequel ce commutateur opère.
- `mode=sc` active la prise en charge de la mise en réseau virtuelle pour le traitement par ordre de priorité des paquets de pulsation du cluster Oracle Solaris dans un environnement de domaines logiques. Les applications telles que le cluster Oracle Solaris doivent garantir que

les paquets de pulsation haute priorité ne sont pas déposés par des périphériques de réseau virtuel et de commutateur saturés. Cette option attribue une priorité aux cadres de pulsation du cluster Oracle Solaris et garanti qu'ils sont transférés de manière fiable.

Vous devez définir cette option lors de l'exécution de l'environnement de domaines logiques et à l'aide des domaines invités en tant que nœuds du cluster Oracle Solaris. Ne définissez *pas* cette option si vous exécutez le logiciel du cluster Oracle Solaris dans des domaines invités, car vous pourriez impacter les performances du réseau virtuel.

- `mtu=taille` définit l'unité de transmission maximale (MTU) d'un périphérique de commutateur virtuel. Les valeurs valides sont comprises entre 1500 et 16000.
- `id=switch-id` est l'ID du nouveau périphérique de commutateur virtuel. Par défaut, les valeurs d'ID sont générées automatiquement. Par conséquent, définissez cette propriété si vous devez faire correspondre un nom de périphérique existant dans le SE. Voir [“Identificateur de périphérique virtuel et nom d'interface réseau” à la page 120](#).
- `vswitch-name` est le nom unique du commutateur qui doit être exporté en tant que service. Les clients (réseau) peuvent se connecter à ce service.
- `ldom` définit le domaine logique dans lequel ajouter un commutateur virtuel.

▼ Définition des options pour un commutateur virtuel existant

- Utilisez la syntaxe de commande suivante pour définir les options d'un commutateur virtuel existant déjà.

```
# ldm set-vsw [pvid=[port-vlan-id]] [vid=[vlan-id1,vlan-id2,...]] [mac-addr=num]
[linkprop=[phys-state]] [net-dev=[device]] [mode=[sc]] [mtu=[size]] vswitch-name
```

où :

- `mode=` (laissé vide) arrête le traitement spécial des paquets de pulsation du cluster Oracle Solaris.
- Sinon, les arguments de la commande sont identiques à ceux décrits dans la section [“Ajout d'un commutateur virtuel” à la page 116](#).

▼ Suppression d'un commutateur virtuel

- Utilisez la syntaxe de commande suivante pour supprimer un commutateur virtuel.

```
# ldm rm-vsw [-f] vswitch-name
```

où :

- -f tente de forcer la suppression d'un commutateur virtuel. La suppression peut échouer.
- *vswitch-name* est le nom du commutateur à supprimer en tant que service.

Gestion d'un périphérique réseau virtuel

Cette section décrit l'ajout d'un périphérique réseau virtuel à un domaine, la définition des options pour un périphérique réseau virtuel existant et la suppression d'un périphérique réseau virtuel.

▼ Ajout d'un périphérique réseau virtuel

- Utilisez la syntaxe de commande suivante pour ajouter un périphérique réseau virtuel.

```
# ldm add-vnet [mac-addr=num] [mode=hybrid] [pvid=[port-vlan-id]]
  [linkprop=phys-state] [vid=vlan-id1,vlan-id2,...] [mtu=size] [id=network-id]
  if-name vswitch-name ldom
```

où :

- *mac-addr=num* est l'adresse MAC de ce périphérique réseau. Le numéro doit être en notation d'octet standard, par exemple, 80:00:33:55:22:66. Voir [“Assignment automatique et manuelle des adresses MAC”](#) à la page 122 pour plus d'informations.
- *mode=hybrid* pour demander au système d'utiliser l'E/S hybride NIU sur ce vnet si possible. Si c'est impossible, le système revient à l'E/S virtuelle. Le mode hybride est considéré comme une reconfiguration retardée s'il est défini sur un vnet actif. Voir [“Utilisation des E/S hybrides NIU”](#) à la page 139 pour plus d'informations.
- *pvid=id-vlan-port* définit le VLAN dont le périphérique réseau virtuel doit être membre en mode non balisé. Voir [“Utilisation du balisage VLAN”](#) à la page 135 pour plus d'informations.
- *linkprop=phys-state* indique si le périphérique réseau virtuel signale son état de liaison en fonction du périphérique réseau physique sous-jacent. Lorsque *linkprop=phys-state* est défini dans une ligne de commande, l'état de liaison du périphérique réseau virtuel reflète l'état de liaison physique. Par défaut, l'état de liaison du périphérique réseau virtuel ne reflète pas l'état de liaison physique.
Définissez cette option pour utiliser l'IPMP basé sur la liaison. Voir [“Utilisation de l'IPMP basé sur liaison dans la mise en réseau virtuelle des domaines logiques”](#) à la page 130
- *vid=vlan-id* définit un ou plusieurs VLAN dont un périphérique réseau virtuel doit être membre en mode balisé. Voir [“Utilisation du balisage VLAN”](#) à la page 135 pour plus d'informations.
- *mtu=taille* définit l'unité de transmission maximale (MTU) d'un périphérique réseau virtuel. Les valeurs valides sont comprises entre 1500 et 16000.

- `id=id-réseau` est l'ID du nouveau périphérique réseau virtuel. Par défaut, les valeurs d'ID sont générées automatiquement. Par conséquent, définissez cette propriété si vous devez faire correspondre un nom de périphérique existant dans le SE. Voir [“Identificateur de périphérique virtuel et nom d'interface réseau”](#) à la page 120.
- `if-name`, nom d'interface, est un nom unique vers le domaine logique, assigné à cette instance de périphérique réseau virtuel pour référence sur les commandes `ldm set -vnet` ou `ldm rm -vnet` ultérieures.
- `vswitch-name` est le nom d'un service de périphérique existant (commutateur virtuel) auquel se connecter.
- `ldom` définit le domaine logique auquel ajouter le périphérique réseau virtuel.

▼ Définition des options pour un périphérique réseau virtuel existant

- Utilisez la syntaxe de commande suivante pour définir les options d'un périphérique réseau virtuel existant déjà.

```
# ldm set -vnet [mac-addr=num] [vswitch=vswitch-name] [mode=[hybrid]]
  [pvid=[port-vlan-id]] [linkprop=[phys-state]] [vid=[vlan-id1,vlan-id2,...]]
  [mtu=[size]] if-name ldom
```

où :

- `mode=` (laissé vide) désactive l'E/S hybride.
- `if-name`, nom d'interface, est le nom unique assigné au périphérique réseau virtuel que vous voulez définir.
- `ldom` définit le domaine logique à partir duquel supprimer le périphérique réseau virtuel.
- Sinon, les arguments de la commande sont identiques à ceux décrits dans la section [“Ajout d'un périphérique réseau virtuel”](#) à la page 118.

▼ Suppression d'un périphérique réseau virtuel

- Utilisez la syntaxe de commande suivante pour supprimer un périphérique réseau virtuel.

```
# ldm rm -vnet [-f] if-name ldom
```

où :

- `-f` tente de forcer la suppression d'un périphérique réseau virtuel dans un domaine logique. La suppression peut échouer.
- `if-name`, nom d'interface, est le nom unique assigné au périphérique réseau virtuel que vous voulez supprimer.

- *ldom* définit le domaine logique à partir duquel supprimer le périphérique réseau virtuel.

Identificateur de périphérique virtuel et nom d'interface réseau

Si vous ajoutez un commutateur virtuel ou un périphérique réseau virtuel à un domaine, vous pouvez indiquer le numéro de périphérique en définissant la propriété `id`.

```
# ldm add-vsw [id=switch-id] vswitch-name ldom
# ldm add-vnet [id=network-id] if-name vswitch-name ldom
```

Chaque commutateur virtuel ou périphérique réseau virtuel d'un domaine a un numéro de périphérique unique qui est assigné lorsque le domaine est associé. Si un commutateur virtuel ou un périphérique réseau virtuel a été ajouté avec un numéro de périphérique explicite (en définissant la propriété `id`), le numéro de périphérique défini est utilisé. Sinon, le système assigne automatiquement le numéro de périphérique le plus petit disponible. Dans ce cas, le numéro de périphérique assigné dépend de la manière dont le commutateur virtuel ou les périphériques réseau virtuels ont été ajoutés au système. Le numéro de périphérique éventuellement assigné à un commutateur virtuel ou à un périphérique réseau virtuel est visible dans la sortie de la commande `ldm list-bindings` lorsqu'un domaine est associé.

L'exemple suivant montre que le domaine `primary` a une commutateur virtuel, `primary-vsw0`. Ce commutateur virtuel a un numéro de périphérique de 0 (`switch@0`).

```
primary# ldm list-bindings primary
...
VSW
  NAME          MAC          NET-DEV DEVICE  DEFAULT-VLAN-ID PVID VID MTU MODE
  primary-vsw0  00:14:4f:fb:54:f2 nxge0  switch@0  1          1  5,6 1500
...
```

L'exemple suivant montre que le domaine `ldg1` a deux périphériques réseau virtuels : `vnet` et `vnet1`. Le périphérique `vnet` a un numéro de périphérique de 0 (`network@0`) et le périphérique `vnet1` a un numéro de périphérique de 1 (`network@1`).

```
primary# ldm list-bindings ldg1
...
NETWORK
  NAME SERVICE          DEVICE  MAC          MODE PVID VID MTU
  vnet  primary-vsw0@primary network@0 00:14:4f:fb:e0:4b hybrid 1 1500
  ...
  vnet1 primary-vsw0@primary network@1 00:14:4f:f8:e1:ea 1 1500
...
```

Si un domaine avec un commutateur virtuel exécute le SE Oracle Solaris, le commutateur virtuel a une interface réseau, `vswN`. Cependant, le numéro d'interface réseau du commutateur virtuel, `N`, n'est pas nécessairement le même que le numéro de périphérique du commutateur virtuel, `n`.

De même, si un domaine avec un périphérique réseau virtuel exécute le SE Oracle Solaris, le périphérique réseau virtuel a une interface réseau, `vnetN`. Cependant, le numéro d'interface réseau du périphérique réseau virtuel, `N`, n'est pas nécessairement le même que le numéro de périphérique réseau virtuel, `n`.



Attention – LE SE Oracle Solaris préserve le mappage entre le nom de l'interface réseau et un commutateur virtuel ou un périphérique réseau virtuel en fonction du numéro de périphérique. Si un numéro de périphérique n'est assigné pas explicitement à un commutateur virtuel ou à un périphérique réseau virtuel, son numéro de périphérique peut changer lorsque le domaine est dissocié, puis est réassocié ultérieurement. Dans ce cas, le nom de l'interface réseau assigné par le SE s'exécutant dans le domaine peut également changer et rompre la configuration existante du système. Cela peut se produire, par exemple, lorsque l'interface d'un commutateur virtuel ou d'un réseau virtuel est supprimée de la configuration du domaine.

Vous ne pouvez pas utiliser les commandes `ldm list -*` pour déterminer directement le nom de l'interface réseau du SE Oracle Solaris qui correspond à un commutateur virtuel ou à un périphérique de réseau virtuel. Cependant, vous pouvez obtenir ces informations à l'aide d'une association de la sortie de la commande `ldm list -l` et des entrées sous `/devices` du SE Oracle Solaris.

▼ Recherche du nom de l'interface réseau du SE Oracle Solaris

Dans cet exemple de procédure, le domaine invité `ldg1` contient deux périphériques de réseau virtuel, `net - a` et `net - c`. Pour rechercher le nom de l'interface réseau du SE Oracle Solaris dans `ldg1` qui correspond à `net - c`, procédez comme suit. Cet exemple montre également les différences si vous recherchez un nom d'interface réseau d'un commutateur virtuel au lieu d'un périphérique de réseau virtuel.

- 1 Utilisez la commande `ldm` pour rechercher le numéro du périphérique de réseau virtuel pour `net - c`.

```
# ldm list -l ldg1
...
NETWORK
NAME          SERVICE          DEVICE          MAC
net-a         primary-vsw0@primary  network@0      00:14:4f:f8:91:4f
net-c         primary-vsw0@primary  network@2      00:14:4f:f8:dd:68
...
```

Le numéro de périphérique de réseau virtuel pour `net - c` est 2 (`network@2`).

Pour déterminer le nom d'interface réseau d'un commutateur virtuel, recherchez le numéro du périphérique de commutateur virtuel, `n` en tant que `switch@n`.

- 2 Pour trouver l'interface réseau correspondante sur `ldg1`, connectez-vous à `ldg1` et recherchez l'entrée pour ce numéro de périphérique sous `/devices`.

```
# uname -n
ldg1
# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
```

Le nom d'interface réseau est la partie de l'entrée après les deux-points, c'est-à-dire `vnet1`.

Pour déterminer le nom d'interface réseau d'un commutateur virtuel, remplacez l'argument de l'option `-name` par `virtual-network-switch@n*`. Recherchez ensuite l'interface réseau avec le nom `vswN`.

- 3 Montez `vnet1` pour voir s'il a l'adresse MAC `00:14:4f:f8:dd:68` comme indiqué dans la sortie `ldm list -l pour net -c` à l'étape 1.

```
# ifconfig vnet1
vnet1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 0.0.0.0 netmask 0
        ether 0:14:4f:f8:dd:68
```

Assignation automatique et manuelle des adresses MAC

Vous devez disposer du nombre suffisant d'adresses de contrôle d'accès aux médias (MAC) pour assigner le numéro des domaines logiques, des commutateurs virtuels et des réseaux virtuels que vous allez utiliser. Le gestionnaire de domaines logiques peut assigner automatiquement les adresses MAC à un domaine logique, un réseau virtuel (`vnet`), un commutateur virtuel (`vsw`) ou vous pouvez assigner manuellement des adresses MAC à partir de votre propre pool d'adresses MAC assignées. Les sous-commandes `ldm` définissant des adresses MAC sont `add-domain`, `add-vsw`, `set-vsw`, `add-vnet` et `set-vnet`. Si vous n'indiquez aucune adresse MAC dans ces sous-commandes, le gestionnaire de domaines logiques en assigne une automatiquement.

L'avantage de faire assigner les adresses MAC par le gestionnaire de domaines logiques est qu'il utilise le bloc d'adresses MAC dédié à une utilisation avec les domaines logiques. Le gestionnaire de domaines logiques détecte et empêche également les collisions d'adresse MAC avec d'autres instances du gestionnaire de domaines logiques du même sous-réseau. Cela vous libère de la gestion manuelle de votre pool d'adresses MAC.

L'assignation d'adresse MAC se produit dès qu'un domaine logique est créé ou qu'un périphérique réseau est configuré dans un domaine. Par ailleurs, l'assignation est persistante jusqu'à ce que le périphérique, ou le domaine logique lui-même, soit supprimé.

Plage d'adresses MAC assignées aux domaines logiques

Le bloc suivant d'adresses MAC est assigné aux domaines logiques :

00:14:4F:F8:00:00 ~ 00:14:4F:FF:FF:FF

Les adresses 256K inférieures sont utilisées par le gestionnaire de domaines logiques pour l'*allocation automatique d'adresse MAC* et vous *ne pouvez pas* demander manuellement une adresse dans cette plage :

00:14:4F:F8:00:00 - 00:14:4F:FB:FF:FF

Vous pouvez utiliser la moitié supérieure de cette plage pour l'*allocation manuelle d'adresse MAC* :

00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF

Algorithme d'assignation automatique

Lorsque vous n'indiquez aucune adresse MAC lors de la création d'un domaine logique ou d'un périphérique réseau, le gestionnaire de domaines logiques alloue automatiquement et assigne une adresse MAC à ce domaine logique ou périphérique réseau. Pour obtenir cette adresse MAC, le gestionnaire de domaines logiques tente itérativement de sélectionner une adresse, puis contrôle les collisions potentielles.

Avant de sélectionner une adresse potentielle, le gestionnaire de domaines logiques recherche d'abord s'il a des adresses assignées automatiquement libérées récemment enregistrées dans une base de données à ces fins (voir [“Adresses MAC libérées” à la page 124](#)). Dans ce cas, le gestionnaire de domaines logiques sélectionne son adresse candidate dans la base de données.

Si aucune adresse libérée récemment n'est disponible, l'adresse MAC est sélectionnée de manière aléatoire dans la plage 256K d'adresses mise de côté dans ce but. L'adresse MAC est sélectionnée de manière aléatoire pour réduire le risque de sélectionner une adresse MAC en doublon comme candidat.

L'adresse sélectionnée est contrôlée par rapport aux autres gestionnaire de domaines logiques sur les autres systèmes pour éviter aux adresses MAC en doublon d'être assignées. L'algorithme utilisé est décrit dans la section [“Détection des adresses MAC en doublon” à la page 124](#). Si l'adresse est déjà assignée, le gestionnaire de domaines logiques itère en choisissant une autre adresse et en vérifiant à nouveau les collisions. Cela continue jusqu'à ce qu'une adresse MAC non déjà allouée soit trouvée ou qu'une limite de temps de 30 secondes se soit écoulée. Si la limite de temps est atteinte, la création du périphérique échoue et un message d'erreur semblable au suivant s'affiche.

```
Automatic MAC allocation failed. Please set the vnet MAC address manually.
```

Détection des adresses MAC en doublon

Pour éviter qu'une adresse MAC identique ne soit allouée à différents périphériques, un gestionnaire de domaines logiques effectue une vérification auprès des autres gestionnaires de domaines logiques sur les autres systèmes en envoyant un message multidiffusion par l'interface réseau par défaut du domaine de contrôle, en incluant l'adresse que le gestionnaire de domaines logiques souhaite assigner au périphérique. Le gestionnaire de domaines logiques tentant d'assigner l'adresse MAC attend une réponse pendant une seconde. Si un périphérique différent sur un autre système de domaines logiques a déjà été associé à cette adresse MAC, le gestionnaire de domaines logiques de ce système renvoie une réponse contenant l'adresse MAC en question. Si le gestionnaire de domaines logiques demandant reçoit une réponse, il sait que l'adresse MAC choisie a déjà été allouée, en choisit une autre, puis itère.

Par défaut, ces messages multidiffusion ne sont envoyés qu'aux autres gestionnaires sur le même sous-réseau : le TTL (time-to-live) par défaut est 1. Le TTL peut être configuré à l'aide de la propriété SMF (Utilitaires de gestion des services) `ldmd/hops`.

Chaque gestionnaire de domaines logiques est responsable :

- De l'écoute des messages multidiffusion
- Du suivi des adresses MAC assignées à ses domaines
- De la recherche des doublons
- De la réponse afin que les doublons ne surviennent pas

Si le gestionnaire de domaines logiques sur un système est arrêté pour quelque raison que ce soit, les adresses MAC en doublon peuvent se produire pendant l'arrêt du gestionnaire de domaines logiques.

L'allocation MAC automatique se produit au moment où le domaine logique ou le périphérique réseau est créé et persiste jusqu'à ce que le périphérique ou le domaine logique soit supprimé.

Remarque – Une vérification de détection des adresses MAC en doublon est effectuée lorsque le domaine logique ou le périphérique réseau est créé et que le domaine logique est démarré.

Adresses MAC libérées

Lorsqu'un domaine logique ou un périphérique associé à une adresse MAC automatique est supprimé, l'adresse MAC est enregistrée dans une base de données des adresses MAC libérées récemment pour une utilisation ultérieure potentielle sur ce système. Les adresses MAC sont enregistrées pour éviter l'épuisement des adresses IP (Internet Protocol) d'un serveur DHCP (Dynamic Host Configuration Protocol). Lorsque les serveurs DHCP allouent des adresses IP, ils effectuent cette opération pour une certaine période (durée de location). La durée de location est souvent configurée comme très longue, généralement des heures ou des jours. Si les périphériques réseau sont créés et supprimés à une vitesse élevée sans que le gestionnaire de

domaines logiques réutilise automatiquement les adresses MAC, le nombre d'adresses MAC allouées peut rapidement submerger un serveur DHCP configuré de manière classique.

Lorsque le gestionnaire de domaines logiques est invité à obtenir automatiquement l'adresse MAC pour un domaine logique ou un périphérique réseau, il recherche d'abord dans la base de données des adresses MAC libérées pour voir s'il y a une adresse MAC assignée précédemment qu'il peut utiliser. S'il y a une adresse MAC disponible dans cette base de données, l'algorithme de détection d'adresse MAC en doublon est exécuté. Si l'adresse MAC n'a pas été assignée à quelqu'un d'autre depuis sa libération, elle sera réutilisée et supprimée de la base de données. Si une collision est détectée, l'adresse est tout simplement supprimée de la base de données. Le gestionnaire de domaines logiques essaye ensuite l'adresse suivante dans la base de données ou s'il n'y en a aucune disponible, sélectionne de manière aléatoire une nouvelle adresse MAC.

Utilisation des adaptateurs réseau avec les domaines logiques

Dans un environnement de domaines logiques, le service de commutateur virtuel s'exécutant sur un domaine de service peut directement interagir avec les adaptateurs réseau compatibles GLDv3. Bien que les adaptateurs réseau non compatibles GLDv3 puissent être utilisés dans ces systèmes, le commutateur virtuel ne peut pas communiquer avec eux directement. Reportez-vous à la section [“Configuration du commutateur virtuel et du domaine de service pour le NAT et le routage”](#) à la page 126 pour plus d'informations sur l'utilisation des adaptateurs réseau non compatibles GLDv3.

Pour plus d'informations sur l'utilisation des groupements de liaisons, reportez-vous à la section [“Utilisation du groupement de liens avec un commutateur virtuel”](#) à la page 142.

▼ Procédure de détermination de la compatibilité GLDv3 d'un adaptateur réseau

- 1 Utilisez la commande `dladm(1M)` du SE Oracle Solaris, où, par exemple, `bge0` est le nom du périphérique réseau.

```
# dladm show-link bge0
bge0                type: non-vlan    mtu: 1500        device: bge0
```

- 2 Recherchez `type` dans la sortie :

- Les pilotes compatibles GLDv3 auront un `type: non-vlan` ou `vlan`.
- Les pilotes non compatibles GLDv3 auront un `type legacy`.

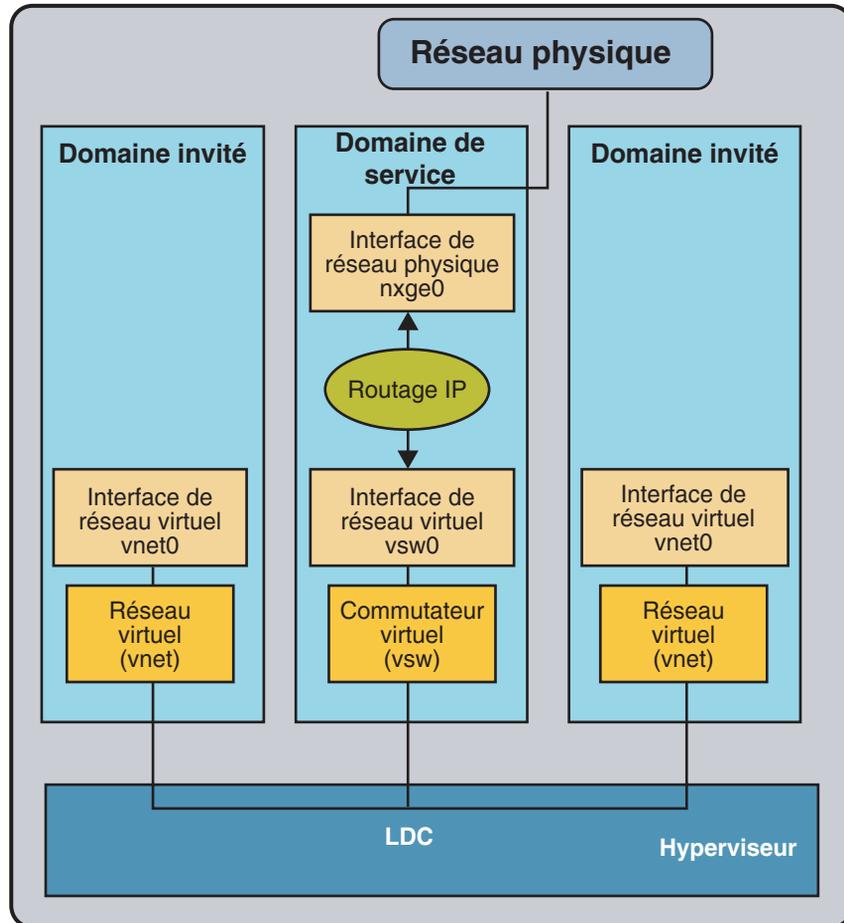
Configuration du commutateur virtuel et du domaine de service pour le NAT et le routage

Le commutateur virtuel (vsw) est un commutateur à 2 couches pouvant également être utilisé comme périphérique réseau dans le domaine de service. Le commutateur virtuel peut être configuré pour agir uniquement comme un commutateur entre le réseau virtuel (vnet) de différents domaines logiques mais sans connectivité à un réseau en dehors de la boîte via un périphérique physique. Dans ce mode, le plombage de vsw en tant que périphérique réseau et l'activation du routage IP sur le domaine de service permet aux réseaux virtuels en dehors de la boîte d'utiliser le domaine de service comme routeur. Ce mode de fonctionnement est essentiel pour fournir une connectivité externe aux domaines lorsque l'adaptateur réseau physique n'est pas compatible GLDv3.

Les avantages de cette configuration sont :

- Le commutateur virtuel ne doit pas utiliser un périphérique physique directement et peut fournir une connectivité externe même si le périphérique sous-jacent n'est pas compatible GLDv3.
- Cette configuration peut tirer parti des fonctionnalités de routage et de filtrage IP du SE Oracle Solaris.

FIGURE 8-2 Routage du réseau virtuel



▼ Configuration d'un commutateur virtuel pour fournir une connectivité externe aux domaines

- 1 Créez un commutateur virtuel sans périphérique physique associé.

Si vous assignez une adresse, vérifiez que le commutateur virtuel a une adresse MAC unique.

```
primary# ldm add-vsw [mac-addr=xx:xx:xx:xx:xx:xx] primary-vsw0 primary
```

- 2 Montez le commutateur virtuel en tant que périphérique réseau sur le périphérique réseau physique utilisé par le domaine.

Reportez-vous à la section [“Configuration du commutateur virtuel en tant qu'interface principale”](#) à la page 53 pour plus d'informations sur le plombage du commutateur virtuel.

3 Configurez le périphérique du commutateur virtuel pour le DHCP, si nécessaire.

Reportez-vous à la section “[Configuration du commutateur virtuel en tant qu'interface principale](#)” à la page 53 pour plus d'informations sur la configuration du périphérique du commutateur virtuel pour le DHCP.

4 Créez le fichier `/etc/dhcp.vsw`, si nécessaire.

5 Configurez le routage IP dans le domaine de service et définissez les tables de routage requises dans tous les domaines.

Pour plus d'informations sur cette opération, reportez-vous à la section “[Transfert et routage de paquets sur des réseaux IPv4](#)” du *Guide d'administration système : services IP*.

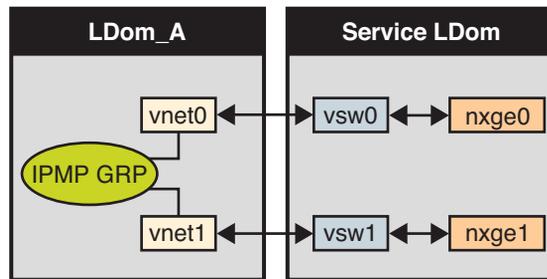
Configuration d'IPMP dans un environnement de domaines logiques

La version 1.3 de domaines logiques a introduit la prise en charge de l'IPMP basé sur les liaisons avec des périphériques de réseau virtuel. Lors de la configuration d'un groupe IPMP avec des périphériques de réseau virtuel, configurez le groupe pour utiliser la détection de liaison. Si vous utilisez des versions antérieures du logiciel Oracle VM Server for SPARC (domaines logiques), vous ne pouvez configurer que la détection par sonde avec les périphériques de réseau virtuel.

Configuration des périphériques de réseau virtuel dans un groupe IPMP dans un domaine

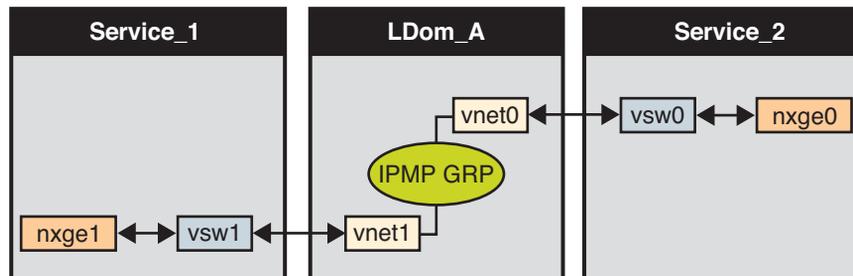
Le schéma suivant représente deux réseaux virtuels (`vnet1` et `vnet2`) connectés à des instances de commutateur virtuel distinctes (`vsw0` et `vsw1`) dans le domaine de service, qui, à son tour, utilise deux interfaces de service différentes (`nxge0` and `nxge1`). En cas de panne d'une liaison physique dans le domaine de service, le périphérique du commutateur virtuel qui est associé à ce périphérique physique détecte la panne de liaison. Ensuite, le périphérique du commutateur virtuel propage la panne au périphérique de réseau virtuel correspondant associé à ce commutateur virtuel. Le périphérique de réseau virtuel envoie une notification de cet événement de liaison à la couche IP dans l'invité `LDom_A`, ce qui aboutit à un basculement sur l'autre périphérique de réseau virtuel dans le groupe IPMP.

FIGURE 8-3 Deux réseaux virtuels connectés à des instances de commutateur virtuel distinctes



On peut obtenir une meilleure fiabilité dans le domaine logique en connectant chaque périphérique de réseau virtuel (`vnet0` et `vnet1`) à des instances de commutateur virtuel dans des domaines de service différents (comme représenté dans le schéma suivant). Dans ce cas, en plus de la panne du réseau physique, `LDom_A` peut détecter une panne du réseau virtuel et déclencher un basculement suite à une panne ou un arrêt du domaine de service.

FIGURE 8-4 Connexion de chaque périphérique de réseau virtuel à des domaines de service différents



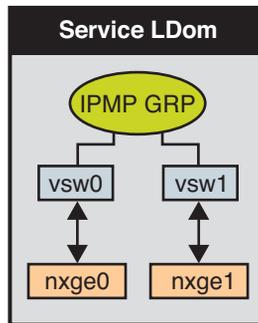
Reportez-vous au [Guide d'administration système : services IP](#) d'Oracle Solaris 10 pour plus d'informations sur la procédure de configuration et l'utilisation des groupes IPMP.

Configuration et utilisation d'IPMP dans le domaine de service

IPMP peut être configuré dans le domaine de service en configurant des interfaces de commutateur virtuel dans un groupe. Le schéma suivant représente deux instances de commutateur virtuel (`vsw0` et `vsw1`) associées à deux périphériques physiques différents. Les deux interfaces de commutateur virtuel peuvent ensuite être plombée et configurée dans un groupe IPMP. En cas de panne d'une liaison physique, le périphérique du commutateur virtuel qui est associé à ce périphérique physique détecte la panne de liaison. Le périphérique du

commutateur virtuel envoie une notification de cet événement de liaison à la couche IP du domaine de service, ce qui aboutit à un basculement sur l'autre périphérique de commutateur virtuel dans le groupe IPMP.

FIGURE 8-5 Configuration de deux interfaces de commutateur virtuel en tant que membre d'un groupe IPMP



Utilisation de l'IPMP basé sur liaison dans la mise en réseau virtuelle des domaines logiques

À partir de la version 1.3 de domaines logiques, les périphériques de réseau virtuel et de commutateur virtuel prennent en charge les mises à jour de l'état de liaison dans la pile réseau. Par défaut, un périphérique de réseau virtuel signale l'état de sa liaison virtuelle (son LDC au commutateur virtuel). Cette configuration est activée par défaut et ne nécessite pas d'autres étapes de configuration de votre part.

Parfois, il peut être nécessaire de détecter les changements d'état de la liaison au réseau physique. Par exemple, si un périphérique physique a été assigné à un commutateur virtuel, la liaison du réseau physique à partir du domaine de service vers le réseau externe peut ne pas fonctionner même si la liaison du périphérique de réseau virtuel à son périphérique de commutateur virtuel fonctionne. Dans ce cas, il peut être nécessaire d'obtenir et de signaler l'état de la liaison physique au périphérique de réseau virtuel et à sa pile.

L'option `linkprop=phys-state` peut être utilisée pour configurer le suivi de l'état de la liaison physique pour les périphériques du réseau virtuel ainsi que pour les périphériques du commutateur virtuel. Lorsque cette option est activée, le périphérique virtuel (réseau virtuel ou commutateur virtuel) signale son état de liaison en fonction de l'état de liaison physique pendant qu'il est plombé en tant qu'interface dans le domaine. Vous pouvez utiliser les commandes d'administration réseau standard d'Oracle Solaris, notamment `dladm(1M)` et `ifconfig(1M)`. Par ailleurs, l'état de liaison est également consigné dans le fichier `/var/adm/messages`.

Remarque – Vous pouvez exécuter les pilotes connaissant et ne connaissant pas l'état de liaison `vnet` et `vsw` simultanément sur un système de domaines logiques. Cependant, si vous avez l'intention de configurer un IPMP basé sur la liaison, vous devez installer un pilote connaissant l'état de liaison. Si vous avez l'intention d'activer les mises à jour de l'état de liaison physique, mettez à niveau les pilotes `vnet` et `vsw` au SE Oracle Solaris 10 9/10, et exécutez au moins la Version 1.3 du gestionnaire de domaines logiques.

▼ Configuration des mises à jour de l'état de liaison

Cette procédure décrit comment activer les mises à jour de l'état de liaison pour les périphériques du réseau virtuel.

Vous pouvez également activer les mises à jour de l'état de liaison physique pour un périphérique du réseau virtuel en suivant une procédure identique et en définissant l'option `linkprop=phys-state` dans les commandes `ldm add-vsw` et `ldm set-vsw`.

Remarque – Vous ne devez utiliser l'option `linkprop=phys-state` que si le périphérique de commutateur virtuel lui-même est plombé en tant qu'interface. Si `linkprop=phys-state` est spécifié et que la liaison physique est interrompue, le périphérique du réseau virtuel signale son état de liaison comme interrompu, même si la connexion au commutateur virtuel fonctionne. Cette situation se produit parce que le SE Oracle Solaris ne fournit actuellement pas d'interface pour signaler deux états de liaison distincts, notamment l'état de liaison virtuelle et l'état de liaison physique.

1 Devenez un superutilisateur ou assumez un rôle équivalent.

Les rôles contiennent des autorisations et des commandes nécessitant des droits d'accès. Pour plus d'informations sur les rôles, reportez-vous à la section "[Configuring RBAC \(Task Map\)](#)" du *System Administration Guide: Security Services*.

2 Activez les mises à jour de l'état de liaison physique pour le périphérique virtuel.

Vous pouvez activer les mises à jour de l'état de liaison physique pour un périphérique de réseau virtuel en procédant comme suit :

- Créez un périphérique de réseau virtuel en indiquant `linkprop=phys-state` lors de l'exécution de la commande `ldm add-vnet`.

La définition de l'option `linkprop=phys-state` configure le périphérique du réseau virtuel pour qu'il obtienne les mises à jour de l'état de liaison physique et les signale à la pile.

Remarque – Si `linkprop=phys-state` est définie et que la liaison physique est interrompue (même si la connexion au commutateur virtuel fonctionne), le périphérique du réseau virtuel signale son état de liaison comme down. Cette situation se produit parce que le SE Oracle Solaris ne fournit actuellement pas d'interface pour signaler deux états de liaison distincts, notamment l'état de liaison virtuelle et l'état de liaison physique.

```
# ldm add-vnet linkprop=phys-state if-name vswitch-name ldom
```

L'exemple suivant active les mises à jour de l'état de liaison physique pour `vnet0` connecté à `primary-vsw0` sur le domaine logique `ldom1` :

```
# ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldom1
```

- Modifiez un périphérique de réseau virtuel existant en indiquant `linkprop=phys-state` lors de l'exécution de la commande `ldm set-vnet`.

```
# ldm set-vnet linkprop=phys-state if-name ldom
```

L'exemple suivant active les mises à jour de l'état de liaison physique pour `vnet0` sur le domaine logique `ldom1` :

```
# ldm set-vnet linkprop=phys-state vnet0 ldom1
```

Pour désactiver les mises à jour de l'état de liaison physique, définissez `linkprop=` en exécutant la commande `ldm set-vnet`.

L'exemple suivant désactive les mises à jour de l'état de liaison physique pour `vnet0` sur le domaine logique `ldom1` :

```
# ldm set-vnet linkprop= vnet0 ldom1
```

Exemple 8-1 Configuration d'un IPMP basé sur liaison

Les exemples suivants montrent comment configurer un IPMP basé sur liaison avec et sans activer les mises à jour de liaison physique :

- L'exemple suivant configure deux périphériques de réseau virtuel sur un domaine. Chaque périphérique de réseau virtuel est connecté à un périphérique de commutateur virtuel distinct sur le domaine de service pour utiliser l'IPMP basé sur liaison.

Remarque – Les adresses de test ne sont pas configurées sur ces périphériques de réseau virtuel. Il est également inutile d'effectuer une configuration supplémentaire lorsque vous utilisez la commande `ldm add-vnet` pour créer ces périphériques de réseau virtuel.

Les commandes suivantes ajoutent les périphériques de réseau virtuel au domaine. Notez que comme `linkprop=phys-state` n'est pas défini, seule la liaison au commutateur virtuel est surveillée pour les changements d'état.

```
# ldm add-vnet vnet0 primary-vsw0 ldom1
# ldm add-vnet vnet1 primary-vsw1 ldom1
```

Les commandes suivantes configurent les périphériques du réseau virtuel sur le domaine invité et les assignent à un groupe IPMP. Notez que les adresses de test ne sont pas configurées sur ces périphériques de réseau virtuel, car la détection des pannes de liaison est en cours d'utilisation.

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 192.168.1.1/24 up
# ifconfig vnet1 192.168.1.2/24 up
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

- L'exemple suivant configure deux périphériques de réseau virtuel sur un domaine. Chaque domaine est connecté à un périphérique de commutateur virtuel distinct sur le domaine de service pour utiliser l'IPMP basé sur liaison. Les périphériques de réseau virtuel sont également configurés pour obtenir les mises à jour de l'état de la liaison physique.

```
# ldm add-vnet linkprop=phys-state vnet0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom1
```

Remarque – Le commutateur virtuel doit disposer d'un périphérique de réseau physique assigné pour que le domaine s'associe correctement. Si le domaine est déjà associé et que le commutateur virtuel ne dispose pas d'un périphérique de réseau physique assigné, la commande `ldm add-vnet` échouera.

Les commandes suivantes plombent les périphériques du réseau virtuel et les assignent à un groupe IPMP :

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 192.168.1.1/24 up
# ifconfig vnet1 192.168.1.2/24 up
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

Configuration et utilisation d'IPMP dans les versions antérieures aux domaines logiques 1.3

Dans les versions des domaines logiques antérieures à la version 1.3, les périphériques de commutateur virtuel et de réseau virtuel ne sont pas capables d'effectuer la détection des pannes de liaison. Dans ces versions, la détection et la récupération des pannes réseau peut être configurée à l'aide d'IPMP basé sur sonde.

Configuration d'IPMP dans le domaine invité

Les périphériques du réseau virtuel dans un domaine invité peuvent être configurés en groupe IPMP comme représenté à la [Figure 8-3](#) et à la [Figure 8-4](#). La seule différence est que la détection des pannes par sonde est utilisée en configurant des adresses de test sur les périphériques de réseau virtuel. Reportez-vous au [Guide d'administration système : services IP](#) pour plus d'informations sur la configuration d'IPMP par sonde.

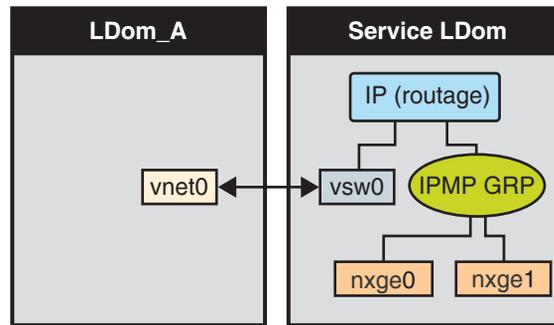
Configuration d'IPMP dans le domaine de service

Dans les version des domaines logiques antérieures à la version 1.3, le périphérique de commutateur virtuel n'est pas capable de détecter les pannes de liaison physique. Dans ce cas, la détection et la récupération des pannes réseau peuvent être définies en configurant les interfaces physiques dans le domaine de service en groupe IPMP. Pour ce faire, configurez le commutateur virtuel du domaine de service sans assigner de périphérique de réseau physique à ce dernier. N'indiquez aucune valeur pour la propriété `net-dev` (`net-dev=`) lorsque vous utilisez la commande `ldm add-vswitch` pour créer le commutateur virtuel. Montez l'interface du commutateur virtuel dans le domaine de service et configurez le domaine de service lui-même pour qu'il agisse comme un routeur IP. Reportez-vous au [Guide d'administration système : services IP](#) d'Oracle Solaris 10 pour plus d'informations sur la configuration du routage IP.

Une fois configuré, le commutateur virtuel envoie tous les paquets provenant des réseaux virtuels (et destinés à une machine externe) à sa couche IP, au lieu d'envoyer les paquets directement au moyen du périphérique physique. En cas de panne de l'interface physique, la couche IP détecte la panne et réachemine automatiquement les paquets via l'interface secondaire.

Comme les interfaces physiques sont directement configurées en groupe IPMP, le groupe peut être configuré pour une détection basée sur la liaison ou sur la sonde. Le schéma suivant représente deux interfaces réseau (`nxge0` et `nxge1`) configurées en tant que membre du groupe IPMP. L'instance de commutateur virtuel (`vsw0`) a été plombée en tant que périphérique réseau pour envoyer des paquets à sa couche IP.

FIGURE 8-6 Configuration de deux interfaces réseau en tant que membre d'un groupe IPMP



▼ Configuration de la route d'un hôte pour l'IPMP basé sur sonde

Remarque – Cette procédure s'applique uniquement aux domaines invités et aux versions antérieures à la version 1.3, où seul l'IPMP basé sur sonde est pris en charge.

Si aucune route explicite n'est configurée pour un routeur dans le réseau correspondant aux interfaces IPMP, une ou plusieurs routes hôtes explicites vers les systèmes cibles doivent être configurées pour que la détection IPMP basée sur sonde fonctionne comme prévu. Sinon, la détection par sonde peut ne pas parvenir à détecter les défaillances réseau.

- **Configurez une route hôte.**

```
# route add -host destination-IP gateway-IP -static
```

Par exemple :

```
# route add -host 192.168.102.1 192.168.102.1 -static
```

Reportez-vous à la section “Configuration de systèmes cible” du *Guide d'administration système : services IP* pour plus d'informations.

Utilisation du balisage VLAN

À partir de la version du SE Oracle Solaris 10 10/08 et du logiciel de domaines logiques version 1.1, la prise en charge du balisage VLAN 802.1Q est disponible dans l'infrastructure réseau des domaines logiques.

Remarque – Les VLAN balisés ne sont pas pris en charge dans les versions précédentes des composants de mise en réseau des domaines logiques.

Les périphériques de commutateur virtuel (vsw) et de réseau virtuel (vnet) prennent en charge la commutation des paquets Ethernet en fonction de l'identificateur du réseau local virtuel (VLAN) et traitent la balise ou le non balisage nécessaire des cadres Ethernet.

Vous pouvez créer plusieurs interfaces VLAN sur un périphérique vnet dans un domaine invité. Vous pouvez utiliser la commande `ifconfig(1M)` du SE Oracle Solaris pour créer une interface VLAN sur un périphérique de réseau virtuel de la même façon qu'elle est utilisée pour configurer une interface VLAN sur tout autre périphérique réseau physique. La contrainte supplémentaire dans un environnement de domaines logiques est que vous devez assigner le vnet aux VLAN correspondants à l'aide des commandes de la CLI du gestionnaire de domaines logiques. Reportez-vous au `ldm(1M)` pour obtenir des informations complètes sur les commandes de la CLI du gestionnaire de domaines logiques.

De même, vous pouvez configurer des interfaces VLAN sur un périphérique de commutateur virtuel dans le domaine de service. Les ID de VLAN 2 à 4094 sont valides, l'ID de VLAN 1 est réservées en tant que `default-vlan-id`.

Lorsque vous créez un périphérique vnet sur un domaine invité, vous devez lui assigner les VLAN requis en définissant l'ID de VLAN du port et zéro ou plusieurs ID de VLAN pour ce vnet à l'aide des arguments `pvid=` et `vid=` de la commande `ldm add-vnet`. Cette opération configure le commutateur virtuel afin qu'il prenne en charge plusieurs VLAN dans le réseau de domaines logiques et commute les paquets à l'aide de l'adresse MAC et des ID de VLAN dans le réseau.

De même, les VLAN auxquels le périphérique vsw lui-même doit appartenir, s'il est plombé en tant qu'interface réseau, doivent être configurés dans le périphérique vsw à l'aide des arguments `pvid=` et `vid=` de la commande `ldm add-vsw`.

Vous pouvez modifier les VLAN auxquels un périphérique appartient à l'aide de la commande `ldm set-vnet` ou `ldm set-vsw`.

ID du VLAN du port (PVID)

Le PVID définit le VLAN dont le périphérique réseau virtuel doit être membre en mode non balisé. Dans ce cas, le périphérique vsw fournit le balisage ou le non balisage nécessaire des cadres pour le périphérique vnet sur le VLAN défini par son PVID. Les cadres de sortie du réseau virtuel non balisés sont balisés avec son PVID par le commutateur virtuel. Les cadres entrants balisés par ce PVID sont non balisés par le commutateur virtuel avant leur envoi au périphérique vnet. Par conséquent, l'assignation d'un PVID à un vnet signifie implicitement que le port de réseau virtuel correspondant sur le commutateur virtuel est marqué comme non balisé pour le VLAN spécifié par le PVID. Vous ne pouvez avoir qu'un seul PVID pour un périphérique vnet.

L'interface de réseau virtuel correspondante, si configurée à l'aide de la commande `ifconfig(1M)` sans ID de VLAN et utilisant uniquement son instance de périphérique, fait que l'interface est implicitement assignée au VLAN spécifié par le PVID du réseau virtuel.

Par exemple, si vous envisagez de monter l'instance `vnet 0`, à l'aide de la commande suivante et si l'argument `pvid=` pour `vnet` a été définie sur `10`, l'interface `vnet0` serait assignée de manière implicite afin d'appartenir au VLAN 10.

```
# ifconfig vnet0 plumb
```

ID de VLAN (VID)

Le VID indique le VLAN dont un périphérique de réseau virtuel ou un commutateur virtuel doit être membre en mode balisé. Le périphérique de réseau virtuel envoie et reçoit des cadres balisés sur les VLAN spécifiés par ses VID. Le commutateur virtuel transmet les cadres balisés avec le VID spécifié entre le périphérique de réseau virtuel et le réseau externe.

▼ Assignation de VLAN à un commutateur virtuel et à un périphérique de réseau virtuel

1 Assignez le commutateur virtuel (vsw) à deux VLAN.

Par exemple, configurez le VLAN 21 comme non balisé et le VLAN 20 comme balisé. Assignez le réseau virtuel (`vnet`) à trois VLAN. Configurez le VLAN 20 comme non balisé et les VLAN 21 et 22 comme balisés.

```
# ldm add-vsw net-dev=nxge0 pvid=21 vid=20 primary-vsw0 primary
# ldm add-vnet pvid=20 vid=21,22 vnet01 primary-vsw0 ldom1
```

2 Montez les interfaces VLAN.

Cet exemple suppose que le numéro d'interface de ces périphériques est `0` dans les domaines et que les VLAN soient mappés sur ces sous-réseaux :

VLAN	Sous-réseau
20	192.168.1.0 (masque de réseau : 255.255.255.0)
21	192.168.2.0 (masque de réseau : 255.255.255.0)
22	192.168.3.0 (masque de réseau : 255.255.255.0)

a. Montez l'interface VLAN dans le domaine de service (primary).

```
primary# ifconfig vsw0 plumb
primary# ifconfig vsw0 192.168.2.100 netmask 0xffffffff broadcast + up
primary# ifconfig vsw20000 plumb
primary# ifconfig vsw20000 192.168.1.100 netmask 0xffffffff broadcast + up
```

b. Montez l'interface VLAN dans le domaine invité (ldom1).

```
ldom1# ifconfig vnet0 plumb
ldom1# ifconfig vnet0 192.168.1.101 netmask 0xffffffff broadcast + up
ldom1# ifconfig vnet21000 plumb
ldom1# ifconfig vnet21000 192.168.2.101 netmask 0xffffffff broadcast + up
ldom1# ifconfig vnet22000 plumb
ldom1# ifconfig vnet22000 192.168.3.101 netmask 0xffffffff broadcast + up
```

Pour plus d'informations sur la procédure de configuration des interfaces VLAN dans le SE Oracle Solaris, reportez-vous à la section “[Administration de réseaux locaux virtuels](#)” du *Guide d'administration système : services IP*.

▼ Installation d'un domaine invité lorsque le serveur d'installation est un VLAN

Faites attention lorsque vous installez un domaine invité sur le réseau (JumpStart) et que le serveur d'installation est un VLAN. Définissez l'ID de VLAN associée au serveur d'installation comme le PVID du périphérique de réseau virtuel et ne configurez pas de VLAN balisés (vid) pour ce périphérique de réseau virtuel. Vous devez effectuer cette opération, car l'OBP ne connaît pas les VLAN et ne peut pas traiter les paquets réseau balisés VLAN. Le commutateur virtuel traite le non balisage et le balisage des paquets de et vers le domaine invité au cours de l'installation réseau. À la fin de l'installation réseau et après la démarrage du SE Oracle Solaris, vous pouvez configurer le périphérique du réseau virtuel pour qu'il soit balisé dans ce VLAN. Vous pouvez ensuite ajouter le périphérique de réseau virtuel à d'autres VLAN en mode balisé.

Pour plus d'informations sur l'utilisation de JumpStart pour effectuer une installation sur un domaine invité, reportez-vous à la section “[Réalisation d'une opération JumpStart sur un domaine invité](#)” à la page 63.

1 Configurez le périphérique réseau en mode non balisé au départ.

Par exemple, si le serveur d'installation est en VLAN 21, configurez le réseau virtuel au départ comme suit :

```
primary# ldm add-vnet pvid=21 vnet01 primary-vsw0 ldom1
```

2 À la fin de l'installation et du démarrage du SE Oracle Solaris, configurez le réseau virtuel en mode balisé.

```
primary# ldm set-vnet pvid= vid=21, 22, 23 vnet01 primary-vsw0 ldom1
```

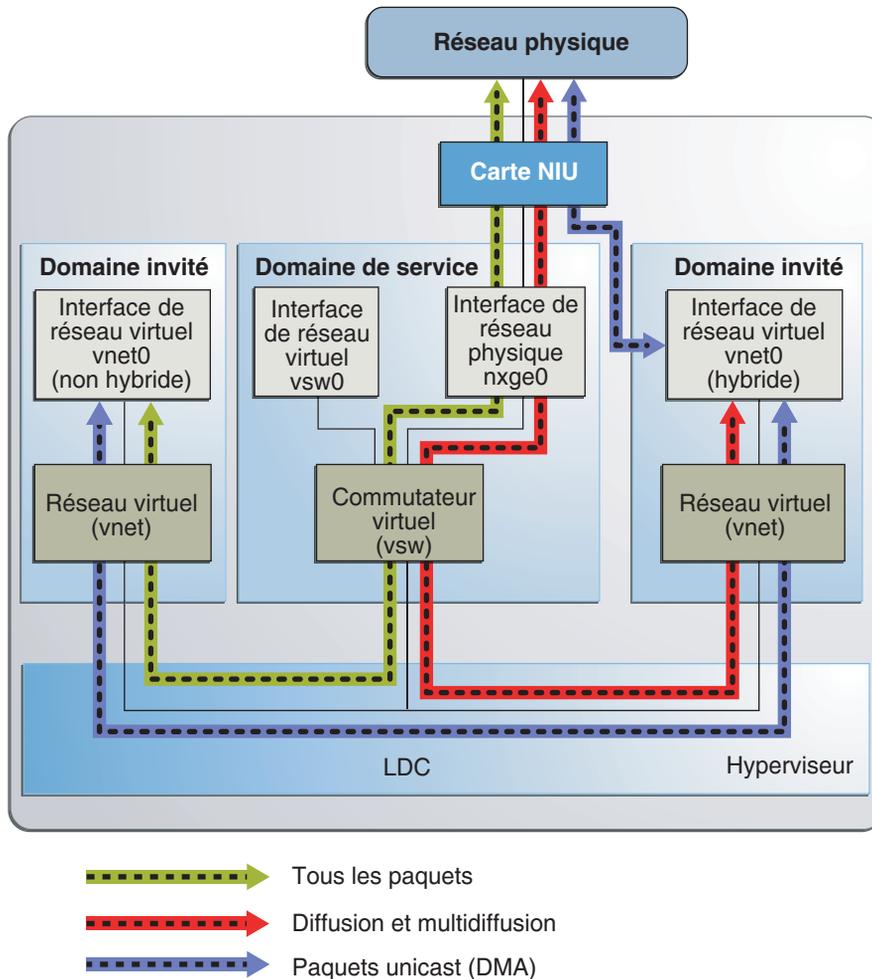
Utilisation des E/S hybrides NIU

Le cadre d'E/S virtuel implémente un modèle d'E/S *hybride* pour optimiser les fonctionnalités et les performances. Le modèle d'E/S hybride associe des E/S directes et virtualisées pour permettre un déploiement flexible des ressources d'E/S sur les machines virtuelles. Cela est particulièrement utilisé lorsque les E/S directes ne fournissent pas une fonctionnalité totale pour la machine virtuelle ou que les E/S directes ne sont disponibles de manière persistante et cohérente pour la machine virtuelle. Cela peut arriver en raison de la disponibilité des ressources ou de la migration de la machine virtuelle. L'architecture d'E/S hybride est bien adaptée pour l'unité d'interface réseau (NIU) sur les plates-formes Sun UltraSPARC T2 et SPARC T3. Une NIU est une interface d'E/S réseau intégrée sur la puce. Cette architecture permet l'assignation dynamique de ressources DMA (Direct Memory Access) aux périphériques de mise en réseau virtuels et, fournit ainsi des performances cohérentes aux applications du domaine.

Les E/S hybrides NIU sont disponibles pour les plates-formes Sun UltraSPARC T2 et SPARC T3. Cette fonction est activée par un mode hybride facultatif qui fournit un périphérique de réseau virtuel (vnet) où les ressources matérielles DMA sont louées à un périphérique vnet dans un domaine invité pour améliorer les performances. En mode hybride, un périphérique vnet dans un domaine invité peut envoyer et recevoir du trafic unicast à partir d'un réseau externe directement dans le domaine invité à l'aide des ressources matérielles DMA. Le trafic de diffusion ou de multidiffusion et le trafic unicast vers les autres domaines invités du même système continuent à être envoyés à l'aide du mécanisme de communication d'E/S.

Remarque – Les E/S hybrides NIU ne sont pas disponibles sur les plates-formes UltraSPARC T2 Plus.

FIGURE 8-7 Mise en réseau virtuelle hybride



Le mode hybride s'applique uniquement aux périphériques vnet associés à un commutateur virtuel (vsw) configuré pour utiliser un périphérique réseau NIU. Comme les ressources matérielles DMA partageables sont limitées, jusqu'à trois périphériques vnet par vsw peuvent se voir assigner des ressources matérielles DMA sur une période donnée. Si plus de trois périphériques vnet ont le mode hybride activé, l'assignation est effectuée sur la base du premier arrivé, premier servi. Comme il y a deux périphériques réseau NIU dans un système, il peut y avoir un total de six périphériques vnet sur deux commutateurs virtuels différents avec des ressources matérielles DMA assignées.

Vous trouverez ci-dessous des points dont vous devez avoir connaissance lors de l'utilisation de cette fonction :

- L'option de mode hybride pour un périphérique vnet est traitée comme une suggestion uniquement. Cela signifie que les ressources DMA sont assignées uniquement lorsqu'elles sont disponibles et que le périphérique est capable de les utiliser.
- Les commandes de la CLI du gestionnaire de domaines logiques ne valident pas l'option de mode hybride, c'est-à-dire qu'il est possible de définir le mode hybride sur un périphérique vnet quelconque ou sur un certain nombre de périphériques vnet.
- Les domaines invités et le domaine de service doivent exécuter le SE Oracle Solaris 10 10/08 au moins.
- Un maximum de trois périphériques vnet par vsw peut disposer des ressources matérielles DMA louées à un moment donné. Comme il y a deux périphériques réseau NIU, il peut y avoir un total de six périphériques vnet avec des ressources matérielles DMA louées.

Remarque – Ne définissez le mode hybride que pour trois périphériques vnet par vsw afin qu'ils soient sûrs d'avoir des ressources matérielles DMA assignées.

- Le mode hybride est désactivé par défaut pour un périphérique vnet. Il doit être explicitement activé avec les commandes de la CLI du gestionnaire de domaines logiques. Reportez-vous à la section [“Activation du mode hybride”](#) à la page 142. (Reportez-vous à la page de manuel [ldm\(1M\)](#) pour plus de détails).
- L'option de mode hybride ne peut pas être modifiée de manière dynamique lorsque le domaine invité est actif.
- Les ressources matérielles DMA ne sont assignées que lorsqu'un périphérique vnet est actif et plombé dans le domaine invité.
- Le pilote Ethernet NIU 10 gigabits (nxge) est utilisé pour la carte NIU. Le même pilote est également utilisé pour les autres cartes réseau 10 gigabits. Cependant, la fonction d'E/S hybride NIU n'est disponible que pour les périphériques réseau NIU.

▼ Configuration d'un commutateur virtuel avec un périphérique réseau NIU

1 Déterminez un périphérique réseau NIU.

L'exemple suivant montre la sortie sur un serveur UltraSPARC T2 :

```
# grep nxge /etc/path_to_inst
"/niu@80/network@0" 0 "nxge"
"/niu@80/network@1" 1 "nxge"
```

L'exemple suivant montre la sortie sur un serveur SPARC T3-1 :

```
# grep nxge /etc/path_to_inst
"/niu@480/network@0" 0 "nxge"
"/niu@480/network@1" 1 "nxge"
```

2 Configurez un commutateur virtuel.

```
# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

▼ Activation du mode hybride

- Par exemple, activez le mode hybride pour un périphérique vnet lors de sa création.

```
# ldm add-vnet mode=hybrid vnet01 primary-vsw0 ldom01
```

▼ Désactivation du mode hybride

- Par exemple, désactivez le mode hybride pour un périphérique vnet.

```
# ldm set-vnet mode= vnet01 ldom01
```

Utilisation du groupement de liens avec un commutateur virtuel

À partir de la version du SE Oracle Solaris 10 10/08 et de la version 1.1 du logiciel des domaines logiques le commutateur virtuel peut être configuré pour utiliser le groupement de liens. Un groupement de liens est utilisé comme périphérique réseau du commutateur virtuel pour se connecter à un réseau physique. Cette configuration permet au commutateur virtuel d'optimiser les fonctions fournies par la norme de groupement de liaisons IEEE 802.3ad. De telles onctions comprennent un bande passante accrue, un équilibrage de la charge et un basculement. Pour plus d'informations sur la procédure de configuration du groupement de liaisons, reportez-vous au *Guide d'administration système : services IP*.

Après avoir créé un groupement de liens, vous pouvez l'assigner à un commutateur virtuel. La réalisation de cette assignation est semblable à l'assignation d'un périphérique de réseau physique à un commutateur virtuel. Utilisez la commande `ldm add-vswitch` ou `ldm set-vswitch` pour définir la propriété `net-dev`.

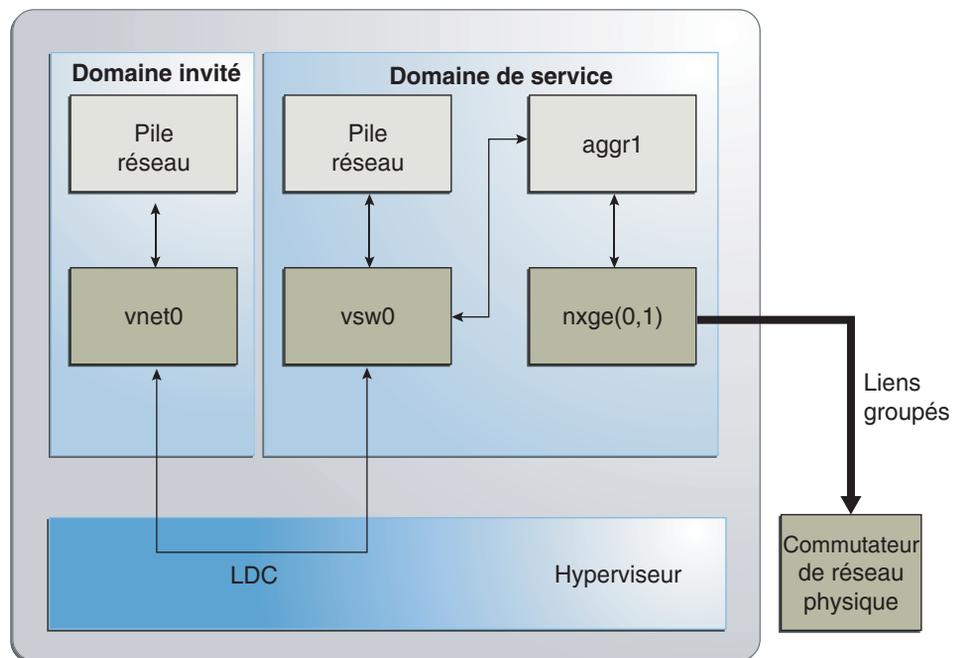
Lorsque le groupement de liens est assigné au commutateur virtuel, le trafic de et vers le réseau physique s'écoule au travers du groupement. Tout équilibrage de charge ou basculement nécessaire est traité de manière transparente par le cadre de groupement sous-jacent. Le groupement de liens est complètement transparent aux périphériques du réseau virtuel (vnet) qui se trouvent sur les domaines invités et sont associés à un commutateur virtuel utilisant un groupement.

Remarque – Vous ne pouvez pas grouper des périphériques de réseau virtuel (vnet et vsw) dans un groupement de liens.

Vous pouvez monter et utiliser le commutateur virtuel configuré pour utiliser un groupement de liens dans le domaine de service. Reportez-vous à la section “[Configuration du commutateur virtuel en tant qu’interface principale](#)” à la page 53.

La figure suivante représente un commutateur virtuel configuré pour utiliser un groupement aggr1 sur les interfaces physiques nxge0 et nxge1.

FIGURE 8-8 Configuration d’un commutateur virtuel pour utiliser un groupement de liens



Configuration des cadres Jumbo

Le commutateur virtuel des domaines logiques (vsw) et les périphériques de réseau virtuel (vnet) prennent maintenant en charge les cadres Ethernet avec des tailles de charge utile supérieures à 1 500 octets. Ce changement a pour conséquence que ces pilotes peuvent augmenter la capacité de traitement du réseau.

▼ Configuration du réseau virtuel et des périphériques de réseau virtuel pour utiliser les cadres Jumbo

Vous activez les cadres Jumbo en définissant l'unité de transmission maximale (MTU) pour le périphérique de commutateur virtuel. Dans ce cas, le périphérique de commutateur virtuel et tous les autres périphériques du réseau virtuel sont associés au périphérique de commutateur virtuel utilisant la valeur MTU définie.

Dans certains cas, vous pouvez définir une valeur MTU directement sur un périphérique de réseau virtuel. Vous pouvez effectuer cette opération si la valeur MTU requise pour le périphérique du réseau virtuel doit être inférieure à celle prise en charge par le commutateur virtuel.

Remarque – Sur le SE Oracle Solaris 10 5/09, la MTU d'un périphérique physique doit être configurée pour correspondre à la MTU du commutateur virtuel. Pour plus d'informations sur la configuration de ces pilotes en particulier, reportez-vous à la page de manuel qui correspond à ce pilote dans la section 7D du manuel de référence Oracle Solaris. Par exemple, pour obtenir des informations sur le pilote `nxge`, reportez-vous à la page de manuel [nxge\(7D\)](#).

1 Connectez-vous au domaine de contrôle.

2 Devenez un superutilisateur ou assumez un rôle équivalent.

Les rôles contiennent des autorisations et des commandes nécessitant des droits d'accès. Pour plus d'informations sur les rôles, reportez-vous à la section “[Configuring RBAC \(Task Map\)](#)” du *System Administration Guide: Security Services*.

3 Déterminez la valeur de MTU que vous voulez utiliser pour ce réseau virtuel.

Vous pouvez indiquer une valeur MTU comprise entre 1 500 et 16 000 octets. La MTU indiquée doit correspondre à la MTU du périphérique du réseau physique qui est assignée au commutateur virtuel.

4 Indiquez la valeur MTU d'un périphérique de commutateur virtuel ou d'un périphérique de réseau virtuel.

Effectuez l'une des opérations suivantes :

- Activez les cadres Jumbo sur un nouveau périphérique de commutateur virtuel dans le domaine de service en définissant sa MTU comme valeur de la propriété `mtu`.

```
# ldm add-vsw mtu=value vswitch-name ldom
```

En plus de configurer le commutateur virtuel, cette commande met à jour la valeur MTU de chaque périphérique de réseau virtuel qui sera associé au commutateur virtuel.

- Activez les cadres Jumbo sur un périphérique de commutateur virtuel existant dans le domaine de service en définissant sa MTU comme valeur de la propriété `mtu`.

```
# ldm set-vsw mtu=value vswitch-name
```

En plus de configurer le commutateur virtuel, cette commande met à jour la valeur MTU de chaque périphérique de réseau virtuel qui sera associé au commutateur virtuel.

Dans de rares cas, vous devrez utiliser la commande `ldm add-vnet` ou `ldm set-vnet` pour spécifier une valeur MTU pour un périphérique de réseau virtuel qui diffère de la valeur MTU du commutateur virtuel. Par exemple, vous pouvez modifier la valeur MTU du périphérique de réseau virtuel si vous configurez des VLAN sur un périphérique de réseau virtuel et la MTU du VLAN la plus grande est inférieure à la valeur MTU sur le commutateur virtuel. Un pilote `vnet` qui prend en charge les cadres Jumbo n'est pas toujours nécessaire pour les domaines dans lesquels seule la valeur MTU par défaut est utilisée. Cependant, si les domaines ont des périphériques de réseau virtuel associés à un commutateur virtuel utilisant des cadres Jumbo, vérifiez que le pilote `vnet` prend en charge les cadres Jumbo.

Si vous utilisez la commande `ldm set-vnet` pour définir une valeur `mtu` sur un périphérique de réseau virtuel, les mises à jour ultérieures de la valeur MTU du périphérique de commutateur virtuel ne sont pas propagées à ce périphérique de commutateur virtuel. Pour réactiver le périphérique de réseau virtuel afin d'obtenir la valeur MTU du périphérique de commutateur virtuel, exécutez la commande suivante :

```
# ldm set-vnet mtu= vnet-name ldom
```

Notez que l'activation des cadres Jumbo pour un périphérique de réseau virtuel active automatiquement les cadres Jumbo pour une ressource d'E/S hybride assignée à un périphérique de réseau virtuel.

Sur le domaine de contrôle, le gestionnaire de domaines logiques met à jour les valeurs MTU initiées par les commandes `ldm set-vsw` et `ldm set-vnet` comme des opérations de reconfiguration retardée. Pour apporter des mises à jour MTU sur des domaines autres que le domaine de contrôle, vous devez arrêter un domaine avant d'exécuter la commande `ldm set-vsw` ou `ldm set-vnet` pour modifier une valeur MTU.

Exemple 8-2 Configuration des cadres Jumbo sur des périphériques de commutateur virtuel et de réseau virtuel

- L'exemple suivant montre comment ajouter un nouveau périphérique de commutateur virtuel utilisant une valeur MTU de 9 000. Cette valeur MTU est propagée du périphérique de commutateur virtuel à tous les périphériques de réseau virtuel du client.

En premier lieu, la commande `ldm add -vsw` crée un périphérique de commutateur virtuel, `primary-vsw0`, avec une valeur MTU de 9000. Notez que l'instance 0 du périphérique réseau `nxge0` est définie comme une valeur de la propriété `net-dev`.

```
# ldm add-vsw net-dev=nxge0 mtu=9000 primary-vsw0 primary
```

Ensuite, la commande `ldm add -vnet` ajoute un périphérique de réseau virtuel client à ce commutateur virtuel, `primary-vsw0`. Notez que la MTU du périphérique de réseau virtuel est assignée implicitement à partir du commutateur virtuel auquel il est associé. En conséquence, la commande `ldm add -vnet` ne nécessite pas que vous définissiez une valeur pour la propriété `mtu`.

```
# ldm add-vnet vnet01 primary-vsw0 ldom1
```

La commande `ifconfig` plombe l'interface du commutateur virtuel dans le domaine de service, `primary`. La sortie de la commande `ifconfig vsw0` montre que la valeur de la propriété `mtu` est 9000.

```
# ifconfig vsw0 plumb
# ifconfig vsw0 192.168.1.100/24 up
# ifconfig vsw0
vsw0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 5
      inet 192.168.1.100 netmask ffffffff broadcast 192.168.1.255
      ether 0:14:4f:fa:0:99
```

La commande `ifconfig` plombe l'interface du réseau virtuel dans le domaine invité, `ldom1`. La sortie de la commande `ifconfig vnet0` montre que la valeur de la propriété `mtu` est 9000.

```
# ifconfig vnet0 plumb
# ifconfig vnet0 192.168.1.101/24 up
# ifconfig vnet0
vnet0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 4
      inet 192.168.1.101 netmask ffffffff broadcast 192.168.1.255
      ether 0:14:4f:f9:c4:13
```

- L'exemple suivant montre comment utiliser la commande `ifconfig` pour modifier la MTU de l'interface à 4 000.

Notez que la MTU d'une interface peut uniquement être remplacée par une valeur inférieure à la MTU du périphérique qui est assigné par le gestionnaire de domaines logiques. Cette méthode est utile lorsque les VLAN sont configurés et que chaque interface de VLAN a besoin d'une MTU différente.

```
# ifconfig vnet0 mtu 4000
# ifconfig vnet0
vnet0: flags=1201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS,FIXEDMTU>
```

```
mtu 4000 index 4
  inet 192.168.1.101 netmask ffffffff broadcast 192.168.1.255
  ether 0:14:4f:f9:c4:13
```

Compatibilité avec des versions antérieures (non connaissant pas Jumbo) des pilotes vnet et vsw

Les pilotes prenant en charge les cadres Jumbo peuvent interopérer avec les pilotes ne prenant pas en charge les cadres Jumbo sur le même système. Cette interopérabilité est possible tant que la prise en charge des cadres Jumbo n'est pas activée lorsque vous créez le commutateur virtuel.

Remarque – Ne définissez pas la propriété `mtu` si des domaines invités ou de service associés au commutateur virtuel n'utilisent pas les pilotes des domaines logiques prenant en charge les cadres Jumbo.

Les cadres Jumbo peuvent être activés en modifiant la propriété `mtu` d'un commutateur virtuel de la valeur par défaut à 1 500. Dans cette instance, des versions antérieures des pilotes ignorent le paramètre `mtu` et continuent à utiliser la valeur par défaut. Notez que la sortie `ldm list` affiche la valeur MTU que vous définissez et non pas la valeur par défaut. Les cadres plus grands que la valeur MTU par défaut ne sont pas envoyés à ces périphériques et sont supprimés par les nouveaux pilotes. Cette situation peut provoquer un comportement incohérent du réseau sur ces invités qui utilisent toujours les anciens pilotes. Cela s'applique aux domaines invités clients et au domaine de service.

Par conséquent, lorsque les cadres Jumbo sont activés, vérifiez que tous les périphériques virtuels dans le réseau de domaines logiques sont mis à niveau pour utiliser les nouveaux pilotes prenant en charge les cadres Jumbo. Vérifiez également d'avoir effectué la mise à niveau à la version 1.2 au moins des domaines logiques afin de pouvoir configurer les cadres Jumbo.

Migration des domaines

Ce chapitre décrit la procédure de migration des domaines d'une machine hôte à une autre.

Ce chapitre aborde les sujets suivants :

- “Introduction à la migration de domaines” à la page 149
- “Présentation d'une opération de migration” à la page 150
- “Compatibilité logicielle” à la page 150
- “Authentification pour les opérations de migration” à la page 151
- “Migration d'un domaine” à la page 151
- “Migration d'un domaine actif” à la page 152
- “Migration de domaines associés ou inactifs” à la page 157
- “Réalisation d'une simulation” à la page 151
- “Surveillance d'une migration en cours” à la page 158
- “Annulation d'une migration en cours” à la page 159
- “Récupération sur un échec de migration” à la page 159
- “Réalisation de migrations non interactives” à la page 151
- “Exemples de migration” à la page 160

Introduction à la migration de domaines

La migration de domaine permet de migrer un domaine logique d'une machine hôte vers une autre. L'hôte sur lequel la migration est lancée est appelée machine source et l'hôte sur lequel le domaine est migré est appelé la machine cible. De même, une fois que la migration est démarrée, le domaine à migrer est appelé le domaine source et le shell d'un domaine créé sur la machine cible est appelé le domaine cible lorsque la migration est en cours.

Présentation d'une opération de migration

Le gestionnaire de domaines logiques sur la machine source accepte les demandes de migration d'un domaine et établit une connexion réseau sécurisée avec le gestionnaire de domaines logiques s'exécutant que la machine cible. Une fois cette connexion établie, la migration a lieu. La migration elle-même peut être décomposée en différentes phases.

Phase 1 : après la connexion au gestionnaire de domaines logiques s'exécutant sur l'hôte cible, les informations sur la machine source et le domaine sont transférées sur l'hôte cible. Ces informations sont utilisées pour effectuer une série de contrôles pour déterminer si une migration est possible. Les contrôles diffèrent en fonction de l'état du domaine source. Par exemple, si le domaine source est actif, un jeu de contrôles différent de celui pour un domaine associé ou inactif est réalisé.

Phase 2 : lorsque tous les contrôles de la Phase 1 ont aboutis, les machines source et cible se préparent à la migration et le domaine source est mis en attente. Sur la machine cible, un domaine est créé pour recevoir le domaine source.

Phase 3 : pour un domaine actif, la phase suivante est le transfert de toutes les informations sur l'état d'exécution du domaine sur la cible. Ces informations sont récupérées à partir de l'hyperviseur. Sur la cible, les informations d'état sont installées dans l'hyperviseur.

Phase 4 : Transfert. Après le transfert de toutes les informations d'état, le transfert a lieu lorsque le domaine cible reprend l'exécution (si la source était active) et que le domaine source est détruit. À partir de ce moment, le domaine cible est la seule version du domaine en cours d'exécution.

Compatibilité logicielle

Pour qu'une migration ait lieu, les machines source et cible doivent exécuter un logiciel compatible, comme suit :

- L'hyperviseur sur les machines source et cible doit disposer d'une version compatible du microprogramme installé. Reportez-vous à la section “Required Software to Enable Oracle VM Server for SPARC 2.0 Features” du *Oracle VM Server for SPARC 2.0 Release Notes* .
- Une version compatible du gestionnaire de domaines logiques doit être en cours d'exécution sur les deux machines.

Remarque – La fonction de migration a été publiée en premier avec le logiciel domaines logiques 1.1 et le microprogramme correspondant. Pour plus d'informations sur le dernier microprogramme de votre plate-forme, reportez-vous au *Oracle VM Server for SPARC 2.0 Release Notes (Notes de version Oracle VM Server for SPARC 2.0)*.

Authentification pour les opérations de migration

Comme l'opération de migration s'exécute sur deux machines, un utilisateur doit être authentifié sur l'hôte source et l'hôte cible. Plus précisément, un utilisateur autre qu'un superutilisateur doit avoir les autorisations `solaris.ldoms.read` et `solaris.ldoms.write`.

L'interface de ligne de commande `ldm` de migration permet à l'utilisateur d'indiquer un autre nom d'utilisateur facultatif pour l'authentification sur l'hôte cible. Si ce n'est pas fait, le nom d'utilisateur de l'utilisateur exécutant la commande de migration est utilisé. Dans les deux cas, l'utilisateur est invité à entrer un mot de passe pour le système cible, à moins que l'option `-p` ne soit utilisée pour initier une migration non interactive.

Migration d'un domaine

Vous pouvez utiliser la commande `ldm migrate-domain` pour démarrer la migration d'un domaine d'un système à un autre.

Pour plus d'informations sur les options et les opérandes de migration, reportez-vous à la page de manuel [ldm\(1M\)](#).

Réalisation d'une simulation

Lorsque vous indiquez l'option `-n` dans la sous-commande `migrate-domain`, des contrôles de migration sont effectués, mais le domaine source n'est pas migré. Toute exigence non remplie est signalée comme une erreur. Cela vous permet de corriger toutes les erreurs de configuration avant de tenter une migration réelle.

Remarque – En raison de la nature dynamique des domaines logiques, il est possible qu'une simulation aboutisse et qu'une migration échoue et inversement.

Réalisation de migrations non interactives

Jusqu'à la version du logiciel domaines logiques 1.3, les migrations étaient des opérations interactives. Lorsque vous démarrez la migration, vous devez indiquer le mot de passe à utiliser pour la machine cible. À partir de la version domaines logiques 1.3, vous pouvez utiliser la commande `ldm migrate-domain -p nomfichier` pour démarrer une opération de migration non interactive.

Le nom de fichier que vous indiquez en tant qu'argument de l'option `-p` doit présenter les propriétés suivantes :

- La première ligne du fichier doit contenir le mot de passe
- Le mot de passe doit être en texte brut
- Le mot de passe ne doit pas dépasser une longueur de 256 caractères

Les caractères de nouvelle ligne à la fin du mot de passe et de toutes les lignes suivantes sont ignorés.

Le fichier dans lequel vous stockez le mot de passe de la machine cible doit être correctement sécurisé. Si vous envisagez de stocker les mots de passe de cette manière, vérifiez que les droits d'accès au fichier sont définis de sorte que seul le propriétaire `root`, ou un utilisateur avec des droits, puisse accéder en lecture ou en écriture au fichier (400 ou 600).

Migration d'un domaine actif

Pour la migration d'un domaine actif avec le logiciel Oracle VM Server for SPARC 2.0, il y a un certain nombre de contraintes et de limitations imposées sur le domaine logique source, la machine source et la machine cible. Les sections suivantes décrivent ces contraintes et limitations pour chacun des types de ressource.

Remarque – L'opération de migration accélère lorsque le domaine `primary` sur les systèmes source et cible ont des unités cryptographiques assignées. À partir de la version 1.3 de &LDoms, vous pouvez accélérer la migration en ajoutant un plus grand nombre de CPU virtuels aux domaines `primary` des systèmes source et cible.

Migration des CPU dans un domaine actif

Vous trouverez ci-dessous les contraintes et les limitations concernant les CPU lorsque vous effectuez une migration :

- Les machines source et cible doivent avoir le même type de processeur s'exécutant à la même fréquence.
- La machine cible doit avoir suffisamment de brins libres pour accueillir le nombre de brins utilisés par le domaine.

D'autres contraintes et limitations s'appliquent dans l'une des conditions suivantes :

- Le système cible n'exécute pas au minimum la version 2.0 du gestionnaire de domaines logiques. Dans ce cas, vous risquez de voir le message suivant au cours de la migration :

```
The target machine is running an older version of the domain
manager that does not support the latest migration functionality.
```

- Le système source n'exécute pas au minimum la version 2.0 du gestionnaire de domaines logiques. Comme le gestionnaire de domaines logiques propriétaire du domaine source est incapable de détecter l'incohérence logicielle, la migration continue sans émettre de message.
- Le domaine source exécute une version du SE Oracle Solaris antérieure au SE Oracle Solaris 10 9/10. Dans ce cas, vous risquez de voir le message suivant au cours de la migration :

```
Domain ldom is not running an operating system that is
compatible with the latest migration functionality.
```

Si l'une des ces conditions s'applique, les contraintes et limitations suivantes concernant les CPU s'appliquent :

- Des serveurs de base entiers doivent être alloués pour le domaine migré. Si le nombre de brins dans le domaine source est inférieur à un serveur de base complet, les brins supplémentaires sont indisponibles pour les domaines tant que le domaine migré n'a pas été redémarré.
- Après une migration, la reconfiguration dynamique (DR) du CPU est désactivée pour le domaine cible tant qu'il n'a pas été redémarré. Après le redémarrage, la DR du CPU devient disponible pour ce domaine.
- Le système cible doit avoir suffisamment de serveurs de base totalement disponibles pour fournir le nombre de brins requis pour le domaine migré. Après la migration, si un serveur de base complet n'est que partiellement utilisé par le domaine migré, tous les brins supplémentaires sont indisponibles pour les domaines tant que le domaine migré n'a pas été redémarré.

Migration des CPU dans un domaine actif

Il doit y avoir suffisamment de mémoire libre sur la machine cible pour permettre la migration de la machine source. Par ailleurs, voici les quelques propriétés qui doivent être conservées pendant la migration :

- Il doit être possible de créer le même nombre de blocs de mémoire de taille identique.
- Les adresses physiques des blocs de mémoire ne doivent pas nécessairement correspondre, mais les mêmes adresses réelles doivent être conservées pendant la migration.

La machine cible doit avoir suffisamment de mémoire libre pour accueillir la migration du domaine source. Par ailleurs, la disposition de la mémoire disponible sur la machine cible doit être compatible avec la disposition de la mémoire sur le domaine source ou la migration échouera.

Plus précisément, si la mémoire de la machine cible est fragmentée en plusieurs petites plages d'adresse, mais que le domaine nécessite une seule plage d'adresse large, la migration échouera. L'exemple suivant illustre ce scénario. Le domaine cible a deux giga-octets de mémoire libre dans deux blocs de mémoire :

```
# ldm list-devices memory
MEMORY
  PA          SIZE
  0x108000000 1G
  0x188000000 1G
```

Le domaine source, `ldg-src`, a également deux giga-octets de mémoire libre, mais il est disposé sur un seul bloc de mémoire :

```
# ldm list -o memory ldg-src
NAME
ldg-src

MEMORY
  RA          PA          SIZE
  0x8000000   0x208000000 2G
```

Étant donné cette disposition de la mémoire, la migration échoue :

```
# ldm migrate-domain ldg-src dt212-239
Target Password:
Unable to bind 2G memory region at real address 0x8000000
Domain Migration of LDom ldg-src failed
```

Remarque – Après une migration, la reconfiguration dynamique (DR) de la mémoire est désactivée pour le domaine cible tant qu'il n'a pas été redémarré. À la fin du redémarrage, la RD de la mémoire est réactivée pour le domaine.

Migration des périphériques d'E/S physiques dans un domaine actif

Les périphériques virtuels qui sont associés à des périphériques physiques peuvent être migrés. Cependant, les domaines ayant un accès direct aux périphériques physiques ne peuvent pas être migrés. Par exemple, vous ne pouvez pas migrer des domaines d'E/S.

Migration des périphériques d'E/S virtuels dans un domaine actif

Tous les services d'E/S virtuels (VIO) utilisés par le domaine source doivent être disponibles sur la machine cible. En d'autres termes, les conditions suivantes doivent être remplies :

- Chaque volume logique utilisé dans le domaine logique source doit être disponible sur l'hôte cible et doit se référer au même stockage.



Attention – Si le volume logique utilisé par la source en tant que périphérique de démarrage existe sur la cible mais ne se réfère pas au même stockage, la migration semble aboutir, mais la machine ne peut pas être utilisée car elle est incapable d'accéder au périphérique de démarrage. Le domaine doit être arrêté, le problème de configuration corrigé et le domaine redémarré. Sinon, le domaine pourrait rester dans un état incohérent.

- Un commutateur de réseau virtuel doit exister sur l'hôte cible pour chaque périphérique de réseau virtuel dans un domaine source, et il doit porter le même nom que le commutateur de réseau virtuel auquel le périphérique est connecté sur l'hôte source.

Par exemple, si `vnet0` dans le domaine source est connecté à un service de commutateur virtuel nommé `switch-y`, il doit y avoir un domaine logique sur l'hôte cible fournissant un service de commutateur virtuel nommé `switch-y`.

Remarque – Les commutateurs ne doivent pas être connectés au même réseau pour que la migration ait lieu, bien que le domaine migré puisse subir des problèmes réseau si les commutateurs ne sont pas connectés au même réseau.

Les adresses MAC utilisées par le domaine source qui se trouvent dans la plage allouée automatiquement doivent être disponibles pour une utilisation sur l'hôte cible.

- Un service de concentrateur de console virtuelle (`vcc`) doit exister sur l'hôte cible et disposer d'au moins trois ports libres. Les contraintes explicites de la console sont ignorées au cours de la migration. La console du domaine cible est créée en utilisant le nom du domaine cible comme groupe de consoles et à l'aide d'un port disponible sur le premier périphérique `vccdu` domaine de contrôle. S'il y a un conflit avec le nom de groupe par défaut, la migration échoue.

Migration d'une entrée/sortie hybride NIU sur un domaine actif

Un domaine utilisant des ressources d'E/S hybrides NIU peut être migré. Une contrainte définissant les ressources d'E/S hybrides NIU n'est pas une contrainte matérielle d'un domaine logique. Si un tel domaine est migré sur une machine ne disposant pas de ressources NIU disponibles, la contrainte est préservée, mais non remplie.

Migration des unités cryptographiques dans un domaine actif

À partir de la version 1.3 des domaines logiques, vous pouvez migrer un domaine invité associé à des unités cryptographiques s'il exécute un système d'exploitation prenant en charge la reconfiguration dynamique (DR) des unités cryptographiques.

Les versions suivantes du SE Oracle Solaris prennent en charge la reconfiguration dynamique des unités cryptographiques :

- Au moins le SE Solaris 10 10/09
- Au moins le SE Oracle Solaris 10 5/08 avec le patch 142245-01

Au début de la migration, le gestionnaire de domaines logiques détermine si le domaine source prend en charge la reconfiguration dynamique des unités cryptographiques. Si tel est le cas, le gestionnaire de domaines logiques tente de supprimer les unités cryptographiques du domaine. À la fin de la migration, les unités cryptographiques sont de nouveau ajoutées au domaine migré.

Remarque – Si les contraintes des unités cryptographiques ne peuvent pas être respectées sur la machine cible, l'opération de migration peut néanmoins aboutir. Dans ce cas, le domaine peut présenter moins d'unités cryptographiques au final qu'avant l'opération de migration.

Reconfiguration retardée dans un domaine actif

Les opérations de reconfiguration retardée active sur les hôtes source et cible empêchent le début de la migration. Les opérations de reconfiguration retardée sont bloquées lorsqu'une migration est en cours.

Migration pendant qu'un domaine actif est en mode élastique

Les migrations de domaine ne sont pas prises en charge pour une machine source ou cible en mode élastique. Si une migration est en cours pendant que le domaine est en mode performance et que la stratégie de gestion de l'alimentation (PM) est définie sur le mode élastique, le changement de stratégie est retardé jusqu'à ce que la migration se termine. La commande de migration renvoie une erreur si la machine source ou cible est en mode élastique et qu'une migration de domaine est tentée.

Opérations sur d'autres domaines

Pendant qu'une migration est en cours sur une machine, toute opération pouvant provoquer la modification de l'état ou la configuration du domaine en cours de migration est bloquée. Toutes les opérations sur le domaine lui-même ainsi que les opérations d'association et d'arrêt sur les autres domaines de la machine sont bloquées.

Migration de domaines associés ou inactifs

Comme un domaine associé ou inactif n'est pas en cours d'exécution au moment de la migration, il y a moins de limitations que lorsque vous migrez un domaine actif.

La migration d'un domaine associé nécessite que la cible soit capable de respecter les contraintes en CPU, mémoire et E/S du domaine source. Sinon, la migration échouera. La migration d'un domaine inactif ne présente pas de telles contraintes. Cependant, la cible doit respecter les contraintes du domaine lorsqu'une association est tentée ultérieurement. Sinon, l'association du domaine échouera.

Migration de CPU dans un domaine associé ou inactif

Vous pouvez migrer un domaine associé ou inactif entre des machines exécutant différents types de processeur et des machines fonctionnant à des fréquences différentes.

L'image du SE Oracle Solaris de l'invité doit prendre en charge le type de processeur de la machine cible.

Migration d'une entrée/sortie virtuelle dans un domaine associé ou inactif

Pour un domaine inactif, il n'y a aucun contrôle effectué en ce qui concerne les contraintes d'entrée/Sortie virtuelle (VIO). Par conséquent, les serveurs n'ont pas besoin d'exister pour que la migration aboutisse. Tout comme avec un domaine inactif, les serveurs VIO doivent exister et être disponibles au moment de l'association du domaine.

Migration de périphériques d'extrémité PCIe dans un domaine associé ou inactif

Vous ne pouvez pas effectuer une migration de domaine sur un domaine d'E/S configuré avec des périphériques d'extrémité PCIe.

Pour plus d'informations sur la fonction d'E/S directe (DIO), reportez-vous à la section “[Assignation des périphériques d'extrémité PCIe](#)” à la page 71.

Surveillance d'une migration en cours

Lorsqu'une migration est en cours, les domaines source et cible sont affichés différemment dans la sortie d'état. La sortie de la commande `ldm list` indique l'état du domaine en migration.

La sixième colonne de la zone `FLAGS` présente l'une des valeurs suivantes :

- Le domaine source présente un `s` pour indiquer qu'il est la source de la migration.
- Le domaine cible présente un `t` pour indiquer qu'il est la cible de la migration.
- Si une erreur se produit et nécessite une intervention de l'utilisateur, un `e` s'affiche.

L'exemple suivant montre que `ldg-src` est le domaine source de la migration :

```
# ldm list ldg-src
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg-src   suspended -n---s    1    1G      0.0%  2h 7m
```

L'exemple suivant montre que `ldg-tgt` est le domaine cible de la migration :

```
# ldm list ldg-tgt
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg-tgt   bound      -----t 5000    1    1G
```

Dans la forme longue de la sortie d'état, des informations supplémentaires sont affichées sur la migration. Sur la source, le pourcentage d'opération terminée est affiché avec le nom de l'hôte cible et du domaine. De même, sur la cible, le pourcentage d'opération terminée est affiché avec le nom de l'hôte source et du domaine.

EXEMPLE 9-1 Surveillance d'une migration en cours

```
# ldm list -o status ldg-src
NAME
ldg-src

STATUS
OPERATION  PROGRESS  TARGET
migration  17%      t5440-sys-2
```

Annulation d'une migration en cours

Une fois qu'une migration est démarrée, si la commande `ldm` est interrompue par un signal `KILL`, la migration est interrompue. Le domaine cible est détruit et le domaine source reprend s'il était actif. Si le shell de contrôle de la commande `ldm` est perdu, la migration continue en arrière-plan.

Une opération de migration peut également être annulée à l'aide de la commande `ldm cancel -operation`. Cela met fin à la migration en cours et le domaine source reprend en tant que domaine actif. La commande `ldm cancel -operation` doit être lancée à partir du système source. Sur un système donné, une commande relative à la migration a une incidence sur l'opération de migration qui a été démarrée à partir de ce système. Un système ne peut pas contrôler une opération de migration lorsqu'il se trouve sur le système cible.

Remarque – Une fois qu'une migration a été démarrée, la suspension du processus `ldm(1M)` ne met pas en pause l'opération, car c'est le démon du gestionnaire de domaines logiques (`ldmd`) sur les machines source et cible qui effectue la migration. Le processus `ldm` attend un signal de `ldmd` indiquant que la migration est terminée avant de revenir.

Récupération sur un échec de migration

Si une connexion réseau est perdue après que la source a terminé d'envoyer toutes les informations d'état d'exécution à la cible, mais avant que la cible n'ait pu indiquer que le domaine a repris, l'opération de migration se termine et la source est placée en état d'erreur. Cela indique qu'une interaction utilisateur est nécessaire pour déterminer si la migration a abouti ou non. Dans ce cas, procédez comme suit.

- Déterminez si le domaine cible a redémarré avec succès. Le domaine cible sera dans l'un des deux états suivants :
 - Si la migration a abouti, le domaine cible est en état normal.
 - Si la migration a échoué, la cible nettoie et détruit le domaine cible.
- Si la cible est redémarrée, le domaine source en état d'erreur peut être détruit en toute sécurité. Si la cible n'est pas présente, le domaine source est encore la version maître du domaine et il doit être récupéré. Pour ce faire, exécutez la commande d'annulation sur la machine source. Cela efface l'état d'erreur et restaure le domaine source à son état d'origine.

Exemples de migration

[Exemple 9-2](#) montre comment un domaine nommé `ldg1`, peut être migré vers une machine appelée `t5440-sys-2`.

EXEMPLE 9-2 Migration d'un domaine invité

```
# ldm migrate-domain ldg1 t5440-sys-2
Target Password:
```

Pour effectuer cette opération sans devoir entrer le mot de passe cible, utilisez la commande suivante :

```
# ldm migrate-domain -p pfile ldg1 t5440-sys-2
```

L'option `-p` prend un nom de fichier comme argument. Le fichier spécifié contient le mot de passe du superutilisateur pour la cible. Dans cet exemple, `pfile` contient le mot de passe de la cible, `t5440-sys-2`.

[Exemple 9-3](#) montre qu'un domaine peut être renommé dans le cadre d'une migration. Dans cet exemple, `ldg-src` est le domaine source renommé en `ldg-tgt` sur la machine cible (`t5440-sys-2`) dans le cadre de la migration. Par ailleurs, le nom d'utilisateur (`root`) sur la machine cible est explicitement défini.

EXEMPLE 9-3 Migration et renommage d'un domaine invité

```
# ldm migrate ldg-src root@t5440-sys-2:ldg-tgt
Target Password:
```

[Exemple 9-4](#) montre un exemple de message d'erreur si le domaine cible ne prend pas en charge la migration, c'est-à-dire si vous exécutez une version des domaines logiques antérieure à la Version 1.1.

EXEMPLE 9-4 Message d'erreur de migration

```
# ldm migrate ldg1 t5440-sys-2
Target Password:
Failed to establish connection with ldmd(1m) on target: t5440-sys-2
Check that the 'ldmd' service is enabled on the target machine and
that the version supports Domain Migration. Check that the 'xmpp_enabled'
and 'incoming_migration_enabled' properties of the 'ldmd' service on
the target machine are set to 'true' using svccfg(1M).
```

[Exemple 9-5](#) montre comment obtenir l'état d'un domaine cible pendant que la migration est en cours. Dans cet exemple, la machine source est `t5440-sys-1`.

EXEMPLE 9-5 Obtention de l'état du domaine cible

```
# ldm list -o status ldg-tgt
NAME
ldg-tgt

STATUS
  OPERATION    PROGRESS    SOURCE
  migration    55%         t5440-sys-1
```

Exemple 9-6 montre comment obtenir l'état analysable d'un domaine source pendant que la migration est en cours. Dans cet exemple, la machine cible est t5440-sys-2.

EXEMPLE 9-6 Obtention de l'état analysable d'un domaine source

```
# ldm list -o status -p ldg-src
VERSION 1.3
DOMAIN|name=ldg-src|
STATUS
|op=migration|progress=42|error=no|target=t5440-sys-2
```


Gestion des ressources

Ce chapitre contient des informations sur la gestion des ressources des systèmes Oracle VM Server for SPARC.

Ce chapitre aborde les sujets suivants :

- “Reconfiguration des ressources” à la page 163
- “Allocation des ressources” à la page 165
- “Allocation de CPU” à la page 165
- “Utilisation de la reconfiguration dynamique de la mémoire” à la page 169
- “Utilisation de la gestion de l'alimentation” à la page 178
- “Utilisation de la gestion dynamique des ressources” à la page 181
- “Liste des ressources du domaine” à la page 184

Reconfiguration des ressources

Une système exécutant le logiciel Oracle VM Server for SPARC peut configurer des ressources, notamment des CPU virtuels, des périphériques d'E/S virtuels, des unités cryptographiques et de la mémoire. Certaines ressources peuvent être configurées de manière dynamique sur un domaine en cours d'exécution, tandis que d'autres doivent être configurées sur un domaine arrêté. Si une ressource ne peut pas être configurée de manière dynamique sur le domaine de contrôle, vous devez d'abord lancer une reconfiguration retardée. La reconfiguration retardée reporte les activités de configuration à après le redémarrage du domaine de contrôle.

Reconfiguration dynamique

La reconfiguration dynamique (DR) permet d'ajouter ou de supprimer des ressources lorsque le système d'exploitation (SE) est en cours d'exécution. La capacité à réaliser une reconfiguration dynamique d'un type de ressource particulier est dépendante du fait que le SE s'exécute sur le domaine logique.

La reconfiguration dynamique est prise en charge pour les ressources suivantes :

- **CPU virtuels** – Pris en charge dans toutes les versions du SE Oracle Solaris 10
- **Périphériques d'E/S virtuels** – Pris en charge au moins dans le SE Solaris 10 10/08
- **Unités cryptographiques** – Pris en charge au moins dans le SE Oracle Solaris 10 9/10
- **Mémoire** – Prise en charge à partir de la version Oracle VM Server for SPARC 2.0 (voir [“Utilisation de la reconfiguration dynamique de la mémoire”](#) à la page 169)
- **Périphériques d'E/S physiques** – Non pris en charge

Pour utiliser la fonction de DR, le démon de DR des domaines logiques `drd` doit être en cours d'exécution dans le domaine que vous souhaitez modifier. Reportez-vous à la page de manuel [`drd\(1M\)`](#).

Reconfiguration retardée

Au contraire des opérations de DR ayant lieu immédiatement, les opérations de reconfiguration retardée ont lieu dans les circonstances suivantes :

- Après le redémarrage du SE
- Après un arrêt ou un démarrage d'un domaine logique

À partir du logiciel 1.2 des domaines logiques, les opérations de reconfiguration retardée sont limitées au domaine de contrôle. Pour tous les autres domaines, vous devez arrêter le domaine pour modifier la configuration à moins que les ressources ne puissent être reconfigurées de manière dynamique.

À partir du logiciel Oracle VM Server for SPARC 2.0, vous devez d'abord démarrer une reconfiguration retardée sur le domaine de contrôle avant d'effectuer des opérations de configuration des ressources. Vous devez redémarrer une reconfiguration retardée à l'aide de la commande `ldm start-reconf primary`.

Lorsqu'une reconfiguration retardée est en cours sur le domaine de contrôle, les autres demandes de reconfiguration du domaine de contrôle sont retardées jusqu'à sa réinitialisation, son arrêt ou son démarrage. Par conséquent, lorsqu'une reconfiguration retardée est en attente pour le domaine de contrôle, les demandes de reconfiguration des autres domaines logiques sont fortement restreintes et échoueront en affichant le message d'erreur correspondant.

La commande `ldm cancel-operation reconf` annule les opérations de reconfiguration retardée sur le domaine de contrôle. Vous pouvez afficher la liste des opérations de reconfiguration à l'aide de la commande `ldm list-domain`. Pour plus d'informations sur l'utilisation de la fonction de reconfiguration retardée, reportez-vous à la page de manuel [`ldm\(1M\)`](#).

Remarque – Vous ne pouvez pas utiliser la commande `ldm cancel-operation reconf` si d'autres commandes `ldm remove-*` ont déjà effectué une opération de reconfiguration retardée sur des périphériques d'E/S virtuels. La commande `ldm cancel-operation reconf` échoue dans ce cas.

Allocation des ressources

À partir de la version Oracle VM Server for SPARC 2.0, le mécanisme d'allocation des ressources utilise les contraintes et les conseils d'allocation des ressources à un domaine au moment de l'association.

Une *contrainte d'allocation de ressource* est une contrainte stricte que le système *doit* respecter lorsque vous assignez une ressource à un domaine. Si la contrainte ne peut pas être respectée, l'allocation de ressource et l'association du domaine échouent.

Un *conseil d'allocation de ressource* est une contrainte légère que le système *tente* de respecter lorsque vous assignez une ressource à un domaine. L'allocation d'une ressource peut toujours aboutir et le domaine peut être associé même si le conseil ne peut pas être totalement respecté. Cette situation peut se produire si le système peut assigner d'une manière qui ne respecte pas nécessairement la contrainte.

Allocation de CPU

Le mécanisme d'allocation de CPU utilise les contraintes et conseils suivants pour les ressources de CPU :

- **Contrainte Whole-core.** Cette contrainte définit que les CPU virtuels sont alloués à un domaine en fonction du nombre indiqué de cœurs de CPU. Le système doit être capable d'allouer le nombre indiqué de cœurs et doit également assigner tous les CPU virtuels à ces serveurs de base alloués au domaine. Si le système ne peut pas allouer le nombre indiqué de cœurs, l'association du domaine échouera.
- **Nombre maximum de contraintes de cœurs.** Cette contrainte définit le nombre maximum de cœurs pouvant être assignés à un domaine associé ou actif. Cette contrainte est automatiquement activée lorsque la contrainte Whole-core est définie sur un domaine. Dans ce cas, le nombre maximum de cœurs est automatiquement défini sur le nombre de cœurs configuré lorsque le domaine est inactif. Actuellement, cette contrainte ne peut pas être activée indépendamment de la contrainte Whole-core, et le nombre maximum de cœurs ne peut pas être défini manuellement.
- **Conseil d'affinité des cœurs.** Ce conseil demande que les CPU virtuels alloués à un domaine viennent des mêmes cœurs de CPU ou d'un nombre inférieur de cœurs de CPU. Le système fait de son mieux pour honorer cette demande. L'association du domaine échoue *uniquement* si un nombre insuffisant de CPU virtuels libres est disponible sur le système.

Le conseil d'affinité des cœurs est activé par défaut et ne peut pas être désactivé.

Remarque – La contrainte Whole-core et le conseil d'affinité des cœurs traitent uniquement de l'emplacement du CPU virtuel sur les cœurs. Ils ne traitent pas de l'emplacement d'un cœur sur les puces ou d'une puce sur des sockets.

Activation de la contrainte Whole-Core

La contrainte whole-core est automatiquement activée lorsque vous définissez le nombre de cœurs à assigner à un domaine. Par défaut, vous définissez les CPU virtuels à assigner à un domaine. Vous ne pouvez activer la contrainte Whole-core que sur un domaine inactif, non pas sur un domaine associé ou actif. Avant d'activer la contrainte whole-core sur le domaine de contrôle, vous devez d'abord démarrer une reconfiguration retardée.

Utilisez la commande `ldm add-vcpu -c number`, `ldm set-vcpu -c number` ou `ldm remove-vcpu -c number` pour assigner des cœurs de CPU à un domaine ou les supprimer. *number* définit le nombre de cœurs de CPU et active la contrainte whole-core. Pour plus d'informations, reportez-vous à la page de manuel [ldm\(1M\)](#).

Vous pouvez utiliser la commande `ldm add-vcpu -c number` ou `ldm remove-vcpu -c number` sur un domaine configuré précédemment avec des CPU virtuels. Dans ce cas, le nombre existant de CPU virtuels est automatiquement converti au nombre correspondant de cœurs. Cette conversion n'est possible que si le nombre existant de CPU virtuels est un multiple du nombre de CPU virtuels par cœur. Sinon, la conversion ne peut pas être effectuée et la commande échoue.

Remarque – Si vous utilisez ces commandes pour activer la contrainte whole-core sur un domaine inactif ou sur le domaine de contrôle dans un mode de reconfiguration retardée, le nombre maximum de cœurs est également défini. Le nombre maximum de cœurs n'est pas affecté lorsque vous utilisez ces commandes sur domaine associé ou actif.

Par exemple, un cœur est constitué de huit CPU virtuels. Si un domaine a sept CPU virtuels assignés, une commande `ldm add-vcpu -c` ou `ldm remove-vcpu -c` ne peut pas respecter la contrainte whole-core. Au lieu de cela, vous devez utiliser la commande `set-vcpu -c` pour indiquer le nombre de cœurs et pour activer la contrainte whole-core.

L'exemple suivant active la contrainte whole-core sur un domaine `ldg1` inactif. La commande `ldm list` vérifie que la contrainte whole-core est activée.

```
primary# ldm add-vcpu -c 1 ldg1
primary# ldm list -o resgmt ldg1
NAME
```

```
ldg1
CONSTRAINT
  whole-core
  max-cores=1
```

Remarque – Lorsque la contrainte `whole-core` est activée sur un domaine, les unités cryptographiques associées à ces cœurs ne sont pas affectées par l'ajout de cœurs. Par conséquent, le système n'ajoute pas automatiquement les unités cryptographiques associées à ce domaine, ni ne les supprime. Vous pouvez également supprimer des cœurs si les unités cryptographiques correspondantes sont assignées au domaine.

Désactivation de la contrainte Whole-Core

Lorsque des CPU virtuels sont assignés à un domaine plutôt que des cœurs, la contrainte `whole-core` est désactivée. Vous ne pouvez désactiver la contrainte `Whole-core` que sur un domaine inactif, non pas sur un domaine associé ou actif. Avant de désactiver la contrainte `whole-core` sur le domaine de contrôle, vous devez d'abord démarrer une reconfiguration retardée.

Utilisez la commande `ldm add-vcnumber`, `ldm set-vcnumber` ou `ldm remove-vcnumber` pour assigner des cœurs de CPU à un domaine ou les supprimer. *number* définit le nombre de CPU virtuels et désactive la contrainte `whole-core`. Pour plus d'informations, reportez-vous à la page de manuel [ldm\(1M\)](#).

Vous pouvez utiliser la commande `ldm add-vcpu number` ou `ldm rm-vcpu number` sur un domaine configuré précédemment avec des CPU virtuels. Dans ce cas, le nombre existant de cœurs de CPU est automatiquement converti au nombre correspondant de CPU virtuels.

Remarque – Lorsque vous désactivez la contrainte `whole-core`, la contrainte de cœurs maximale est automatiquement désactivée.

L'exemple suivant désactive la contrainte `whole-core` sur un domaine `ldg1` inactif :

```
primary# ldm set-vcpu 1 ldg1
```

Allocation de CPU au domaine de contrôle

Pour activer la contrainte `whole-core` sur le domaine de contrôle, ce dernier doit être en mode de reconfiguration retardée. L'activation de la contrainte `whole-core` sur le domaine de contrôle uniquement aboutit si un nombre suffisant de cœurs de CPU est disponible pour respecter la contrainte demandée. C'est-à-dire que les cœurs non utilisés par le domaine de contrôle ou que les cœurs partiellement utilisés par le domaine de contrôle doivent être disponibles. Sinon, l'allocation de CPU sur le domaine de contrôle reste inchangée.

Remarque – Lorsque le domaine de contrôle est en mode de reconfiguration retardée, la contrainte `whole-core` et le paramètre du nombre de cœurs spécifient également le nombre maximum de cœurs.

L'exemple suivant active la contrainte `whole-core` sur le domaine de contrôle (`primary`). En premier lieu, démarrez une reconfiguration retardée sur le domaine de contrôle. Assignez ensuite tout le cœur au domaine de contrôle, puis redémarrez le domaine pour appliquer les modifications.

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the
primary domain reboots, at which time the new configuration for the
primary domain also takes effect.

primary# ldm add-vcpu -c 1 primary
primary# reboot
```

Interactions entre la contrainte `Whole-core` et les fonctions des autres domaines

Cette section décrit les interactions entre la contrainte `whole-core` et les fonctions suivantes :

- [“Reconfiguration dynamique du CPU”](#) à la page 168
- [“Gestion dynamique des ressources”](#) à la page 169
- [“Migration de domaine”](#) à la page 169
- [“Gestion de l'alimentation”](#) à la page 169

Reconfiguration dynamique du CPU

La contrainte `whole-core` est totalement compatible avec la reconfiguration dynamique (DR) du CPU. Lorsqu'un domaine est défini avec la contrainte `whole-core`, vous pouvez utiliser la commande `ldm add-vcpu -c`, `ldm set-vcpu -c` ou `remove-vcpu -c` pour modifier le nombre de cœurs sur un domaine actif.

Cependant, si un domaine associé ou actif n'est pas en mode de reconfiguration retardée, son nombre de cœurs ne peut pas dépasser le nombre maximum de cœurs. Ce maximum est défini avec la contrainte de cœur maximum, qui est automatiquement activée lorsque la contrainte `whole-core` est activée. Une opération de reconfiguration dynamique du CPU ne respectant pas la contrainte de cœur maximum échoue.

Gestion dynamique des ressources

La contrainte whole-core n'est pas compatible avec la gestion dynamique des ressources (DRM). Si une stratégie DRM est activée sur un domaine qui utilise la contrainte whole-core, cette stratégie est automatiquement désactivée. La contrainte whole-core reste activée.

Même si une stratégie DRM ne peut pas être activée lorsqu'une contrainte whole-core est en vigueur, vous pouvez toujours définir un stratégie DRM pour le domaine. Notez que lorsqu'une stratégie est désactivée automatiquement, elle reste toujours active. La stratégie est automatiquement réactivée si le domaine est redémarré sans la contrainte whole-core.

Voici les interactions attendues entre la contrainte whole-core et la DRM :

- Si la contrainte whole-core est définie sur un domaine, un message d'avertissement est émis lorsque vous essayez d'activer une stratégie DRM sur ce domaine.
- Si une stratégie DRM est en vigueur sur un domaine inactif, vous êtes autorisé à activer la contrainte whole-core sur ce domaine. Lorsque le domaine devient actif et que la stratégie est activé, le système désactive automatiquement la stratégie DRM pour ce domaine.
- Si une stratégie DRM est activée sur un domaine actif ou associé, vous n'êtes pas autorisé à activer la contrainte whole-core.

Migration de domaine

La configuration whole-core du CPU n'est pas compatible avec la migration de domaine. Cependant, vous pouvez toujours migrer un domaine qui est configuré avec des cœurs complets de CPU. Pour restaurer la contrainte whole-core après une telle migration, arrêtez le domaine et reconfigurez-le pour l'allocation whole-core.

Gestion de l'alimentation

La contrainte whole-core est totalement compatible avec les performances de gestion de l'alimentation (PM) et les modes élastiques. Lorsque le mode élastique est activé, le sous-système PM peut ajouter des cœurs de CPU configurés avec la contrainte whole-core à un domaine ou les supprimer. Dans ce cas, la contrainte whole-core continue à être honorée, et les domaines utilisant cette contrainte restent configurés avec des cœurs complets uniquement.

Utilisation de la reconfiguration dynamique de la mémoire

La version Oracle VM Server for SPARC 2.0 introduit la reconfiguration dynamique de la mémoire (DR). Cette fonction repose sur la capacité et vous permet d'ajouter une quantité arbitraire de mémoire à un domaine logique actif ou d'en supprimer.

Voici les contraintes et les restrictions d'utilisation de la fonction de DR de mémoire :

- Vous pouvez effectuer des opérations de reconfiguration dynamique sur tous les domaines. Cependant, une seule opération de DR de mémoire peut être en cours sur un domaine à un moment donné.
- La fonction de DR de mémoire force un alignement de 250 Mo sur l'adresse et la taille de la mémoire impliquée dans une opération donnée. Reportez-vous à la section [“Alignement de la mémoire”](#) à la page 172.
- La mémoire non alignée dans le pool de mémoire libre *ne peut pas* être assignée à un domaine à l'aide de la fonction de DR de mémoire. Reportez-vous à la section [“Ajout de mémoire non alignée”](#) à la page 173.

Si la mémoire d'un domaine ne peut pas être reconfigurée à l'aide d'une opération de DR de mémoire, le domaine doit être arrêté avant que la mémoire ne soit configurée. Si le domaine est le domaine de contrôle, vous devez d'abord lancer une reconfiguration retardée.

Ajout de mémoire

Si un domaine est actif, vous pouvez utiliser la commande `ldm add-memory` pour ajouter de la mémoire de manière dynamique au domaine. La commande `ldm set-memory` ajoute également de manière dynamique de la mémoire si la taille de la mémoire indiquée est supérieure à la taille de la mémoire actuelle du domaine.

Suppression de mémoire

Si un domaine est actif, vous pouvez utiliser la commande `ldm remove-memory` pour supprimer de la mémoire de manière dynamique au domaine. La commande `ldm set-memory` supprime également de manière dynamique de la mémoire si la taille de la mémoire indiquée est inférieure à la taille de la mémoire actuelle du domaine.

La suppression de mémoire peut être une opération nécessitant beaucoup de temps. Vous pouvez suivre la progression d'une opération ou annuler une demande de DR de mémoire en cours.

Suivi de la progression d'une demande de reconfiguration dynamique de mémoire

Vous pouvez suivre la progression d'une commande `ldm remove-memory` en exécutant la commande `ldm list -l` pour le domaine indiqué.

Annulation d'une demande de reconfiguration dynamique de mémoire

Vous pouvez annuler une demande de suppression en cours en interrompant la commande `ldm remove-memory command` (en appuyant sur Control-C) ou en émettant la commande `ldm cancel-operation memdr`. Si vous annulez une demande de suppression de mémoire, seule la partie restante de la demande de suppression est affectée, c'est-à-dire la quantité de mémoire devant encore être supprimée du domaine.

Demandes partielles de reconfiguration dynamique de mémoire

Une demande d'ajout de mémoire est rejetée s'il n'y a pas suffisamment de mémoire libre pour remplir toute la demande. Cependant, une demande d'ajout de mémoire peut être partiellement remplie si le domaine cible ne parvient pas à ajouter une partie de la mémoire demandée par le gestionnaire de domaines logiques.

Une demande de suppression de mémoire est rejetée s'il n'y a pas suffisamment de mémoire dans le domaine pour remplir toute la demande. Cependant, une demande de suppression de mémoire peut être partiellement remplie si le domaine cible ne parvient pas à supprimer une partie de la mémoire demandée par le gestionnaire de domaines logiques.

Remarque – La mémoire est effacée après sa suppression d'un domaine et avant d'être ajoutée à un autre domaine.

Reconfiguration de la mémoire du domaine de contrôle

La fonction de reconfiguration dynamique (DR) de la mémoire peut être utilisée pour reconfigurer la mémoire du domaine de contrôle. Si une demande de DR de mémoire ne peut pas être effectuée sur le domaine de contrôle, vous devez d'abord lancer une reconfiguration retardée.

L'utilisation de la reconfiguration dynamique de mémoire peut ne pas être adaptée à la suppression de grandes quantités de mémoire sur un domaine actif, car les opérations de reconfiguration dynamique de mémoire prennent beaucoup de temps. Surtout au cours de la configuration initiale du système, vous devez utiliser la reconfiguration retardée pour réduire la mémoire dans le domaine de contrôle.

Réduction de la mémoire du domaine de contrôle

Utilisez une reconfiguration retardée au lieu d'une reconfiguration dynamique de mémoire pour réduire la mémoire du domaine de contrôle dans une configuration usine par défaut initiale. Dans un tel cas, le domaine de contrôle possède toute la mémoire du système hôte. La

fonction de reconfiguration dynamique de la mémoire n'est pas adaptée à cet objectif, car un domaine actif n'est pas sûr d'ajouter ou plus généralement de donner toute la mémoire demandée. Au lieu de cela, le SE s'exécutant sur ce domaine fait de son mieux pour remplir la demande. Par ailleurs, la suppression de mémoire peut être une opération de longue durée. Ces problèmes sont amplifiés lorsque des opérations importantes sur la mémoire sont impliquées, comme c'est le cas pour la réduction initiale de la mémoire du domaine de contrôle.

Pour ces raisons, utilisez une reconfiguration retardée en procédant comme suit :

1. Utilisez la commande `ldm start -reconf primary` pour mettre le domaine de contrôle en mode de reconfiguration retardée.
2. Partitionnez les ressources du système hôte qui détenues par le domaine de contrôle, si nécessaire.
3. Utilisez la commande `ldm cancel -reconf` pour annuler les opérations de l'étape 2, si nécessaire, et recommencez.
4. Redémarrez le domaine de contrôle pour appliquer les modifications.

Reconfiguration dynamique et retardée

Si une reconfiguration retardée est en attente sur le domaine de contrôle, une demande de reconfiguration de la mémoire est rejetée pour tous les autres domaines. Si aucune reconfiguration retardée n'est en attente dans le domaine de contrôle, la demande de reconfiguration de la mémoire est rejetée pour tout domaine qui ne prend pas en charge la reconfiguration dynamique de la mémoire. Une demande de reconfiguration de la mémoire sur un domaine de contrôle qui ne prend pas en charge la reconfiguration dynamique de la mémoire est convertie en demande de reconfiguration retardée.

Alignement de la mémoire

Les demandes de reconfiguration de la mémoire ont différentes contraintes d'alignement qui dépendent de l'état du domaine sur lequel la demande est appliquée.

Alignement de la mémoire pour les domaines actifs

- **Ajout et suppression dynamiques.** L'adresse et la taille du bloc de mémoire sont alignés à 256 Mo pour un ajout et une suppression dynamiques. La taille de fonctionnement minimale est de 256 Mo.

Une demande non alignée ou une demande de suppression supérieure à la taille associée est rejetée.

Utilisez les commandes suivantes pour ajuster les allocations de mémoire :

- `ldm add-memory`. Si vous indiquez l'option `--auto-adj` dans cette commande, la quantité de mémoire à ajouter est alignée à 256 Mo, ce qui peut augmenter la quantité de mémoire réellement ajoutée au domaine.
- `ldm remove-memory`. Si vous indiquez l'option `--auto-adj` dans cette commande, la quantité de mémoire à supprimer est alignée à 256 Mo, ce qui peut diminuer la quantité de mémoire réellement supprimée du domaine.
- `ldm set-memory`. Cette commande est traitée comme une opération d'ajout ou de suppression. Si vous indiquez l'option `--auto-adj`, la quantité de mémoire à ajouter ou à supprimer est alignée sur 256 Mo comme décrit précédemment. Notez que cet alignement peut augmenter la taille de la mémoire obtenue pour le domaine.
- **Reconfiguration retardée.** L'adresse et la taille d'un bloc de mémoire sont alignés sur 4 Mo. Si vous faites une demande non alignée, la demande est arrondie à un alignement à 4 Mo.

Alignement de la mémoire pour les domaines associés

L'adresse et la taille d'un bloc de mémoire sont alignées à 4 Mo pour les domaines associés. Si vous faites une demande non alignée, la demande est arrondie à un alignement à 4 Mo. Cela signifie que la taille de la mémoire obtenue pour le domaine peut être supérieure à celle indiquée.

Pour les commandes `ldm add-memory`, `ldm set-memory` et `ldm remove-memory`, l'option `--auto-adj` arrondit la taille de la mémoire obtenue sur 256 Mo. Cela signifie que la taille de la mémoire obtenue peut être supérieure à celle indiquée.

Alignement de la mémoire pour les domaines inactifs

Pour les commandes `ldm add-memory`, `ldm set-memory` et `ldm remove-memory`, l'option `--auto-adj` arrondit la taille de la mémoire obtenue sur 256 Mo. Il n'y a aucune contrainte d'alignement pour un domaine inactif. Les restrictions décrites à la section "[Alignement de la mémoire pour les domaines associés](#)" à la page 173 prennent effet dès qu'un tel domaine est associé.

Ajout de mémoire non alignée

La fonction de reconfiguration dynamique de la mémoire force un alignement de mémoire de 256 Mo sur l'adresse et la taille de la mémoire qui est ajoutée de manière dynamique à un domaine actif ou supprimée de ce dernier. Cela signifie que la mémoire non alignée dans un domaine actif ne peut pas être supprimée à l'aide de la reconfiguration dynamique de la mémoire.

Cela signifie également que toute mémoire non alignée dans le pool de mémoire libre ne peut pas être ajoutée à un domaine actif à l'aide de la reconfiguration dynamique de la mémoire.

Après que toute la mémoire alignée a été allouée, vous pouvez utiliser la commande `ldm add-memory` pour ajouter la mémoire restante non alignée à un domaine associé ou inactif. Vous pouvez également utiliser cette commande pour ajouter la mémoire non alignée restante au domaine de contrôle au moyen d'une opération de reconfiguration retardée.

L'exemple suivant montre comment ajouter les deux blocs de mémoire de 128 Mo restants aux domaines `primary` et `ldm1`. Le domaine `ldm1` est à l'état associé. Les commandes suivantes ajoutent les deux blocs de mémoire restants. La première commande lance une opération de reconfiguration retardée sur le domaine de contrôle. La seconde commande ajoute les blocs de mémoire de 128 Mo au domaine de contrôle. La cinquième commande ajoute l'autre bloc de mémoire de 128 Mo au domaine `ldm1`.

```
# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the
primary domain reboots, at which time the new configuration for the
primary domain also takes effect.

# ldm add-memory 128M primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -ndcv-   SP      8       2688M     0.1%    23d 8h 8m

# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP      8       2560M     0.5%    23d 8h 9m
ldm1          bound     - - - - - 5000     1       524M

# ldm add-mem 128M ldm1
# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP      8       2560M     0.1%    23d 8h 9m
ldm1          bound     - - - - - 5000     1       652M
```

Exemples de reconfiguration dynamique de mémoire

Les exemples suivants montrent comment effectuer des opérations de reconfiguration dynamique de la mémoire. Pour plus d'informations sur les commandes de la CLI correspondantes, reportez-vous à la page de manuel [ldm\(1M\)](#).

EXEMPLE 10-1 Opérations de reconfiguration dynamique de la mémoire sur des domaines actifs

Cet exemple montre comment ajouter de la mémoire de manière dynamique et comment la supprimer d'un domaine actif, `ldm1`.

EXEMPLE 10-1 Opérations de reconfiguration dynamique de la mémoire sur des domaines actifs
(Suite)

La sortie `ldm list` montre la mémoire pour chaque domaine dans la zone Memory. La première commande `ldm add-mem` existe avec une erreur, car vous devez indiquer la mémoire en multiples de 256 Mo. La commande suivante `ldm add-mem` utilise l'option `--auto-adj` pour que la quantité de mémoire soit arrondie à 256 Mo bien que vous indiquiez 200M comme la quantité de mémoire à ajouter.

La commande `ldm rm-mem` existe avec une erreur, car vous devez indiquer la mémoire en multiples de 256 Mo. Lorsque vous ajoutez l'option `--auto-adj` à cette même commande, la suppression de mémoire aboutit, car la quantité de mémoire est arrondie à la limite suivante de 256 Mo.

```
# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  0.4%  1d 22h 53m
ldom1         active -n---- 5000   2     2G      0.4%  1d 1h 23m
ldom2         bound  ------ 5001   2     200M
```

```
# ldm add-mem 200M ldom1
The size of memory must be a multiple of 256MB.
```

```
# ldm add-mem --auto-adj 200M ldom1
Adjusting request size to 256M.
The ldom1 domain has been allocated 56M more memory
than requested because of memory alignment constraints.
```

```
# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  5.0%   8m
ldom1         active -n---- 5000   2    2304M   0.5%   1m
ldom2         bound  ------ 5001   2     200M
```

```
# ldm rm-mem --auto-adj 300M ldom1
Adjusting requested size to 256M.
The ldom1 domain has been allocated 44M more memory
than requested because of memory alignment constraints.
```

```
# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active -n-cv- SP    4    27392M  0.3%   8m
ldom1         active -n---- 5000   2     2G      0.2%   2m
ldom2         bound  ------ 5001   2     200M
```

EXEMPLE 10-2 Opérations de reconfiguration dynamique de la mémoire sur des domaines associés

Cet exemple montre comment ajouter de la mémoire de manière dynamique et comment la supprimer d'un domaine associé, `ldom2`.

EXEMPLE 10-2 Opérations de reconfiguration dynamique de la mémoire sur des domaines associés
(Suite)

La sortie `ldm list` montre la mémoire pour chaque domaine dans la zone Memory. La première commande `ldm add-mem` ajoute 100 Mo de mémoire au domaine `ldom2`. La commande suivante `ldm add-mem` définit l'option `--auto-adj`, qui provoque l'ajout dynamique de 112 Mo supplémentaires à `ldom2`.

La commande `ldm rm-mem` supprime de manière dynamique 100 Mo au domaine `ldom2`. Si vous indiquez l'option `--auto-adj` à cette même commande pour supprimer 300 Mo de mémoire, la quantité de mémoire est arrondie à la limite inférieure suivante de 256 Mo.

```
# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.4%  1d 22h 53m
ldom1         active  -n---- 5000  2     2G      0.4%  1d 1h 23m
ldom2         bound   ------ 5001  2     200M
```

```
# ldm add-mem 100M ldom2
```

```
# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.5%  1d 22h 54m
ldom1         active  -n---- 5000  2     2G      0.2%  1d 1h 25m
ldom2         bound   ------ 5001  2     300M
```

```
# ldm add-mem --auto-adj 100M ldom2
```

Adjusting request size to 256M.

The `ldom2` domain has been allocated 112M more memory than requested because of memory alignment constraints.

```
# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.4%  1d 22h 55m
ldom1         active  -n---- 5000  2     2G      0.5%  1d 1h 25m
ldom2         bound   ------ 5001  2     512M
```

```
# ldm rm-mem 100M ldom2
```

```
# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  3.3%  1d 22h 55m
ldom1         active  -n---- 5000  2     2G      0.2%  1d 1h 25m
ldom2         bound   ------ 5001  2     412M
```

```
# ldm rm-mem --auto-adj 300M ldom2
```

Adjusting request size to 256M.

The `ldom2` domain has been allocated 144M more memory than requested because of memory alignment constraints.

```
# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.5%  1d 22h 55m
ldom1         active  -n---- 5000  2     2G      0.2%  1d 1h 26m
ldom2         bound   ------ 5001  2     256M
```

EXEMPLE 10-3 Définition des tailles de mémoire du domaine

Cet exemple montre comment utiliser la commande `ldm set-memory` pour ajouter de la mémoire à un domaine et en supprimer.

La sortie `ldm list` montre la mémoire pour chaque domaine dans la zone Memory. La première commande `ldm set-mem` tente de définir la taille du domaine `primary` à 3 400 mégaoctets. L'erreur résultant indique que la valeur indiquée ne se trouve pas dans la limite de 256 Mo. L'ajout de l'option `--auto-adj` à cette même commande vous permet de supprimer avec succès de la mémoire et de rester dans la limite de 256 Mo. Cette commande émet également un avertissement pour indiquer que la totalité de la mémoire demandée n'a pas pu être supprimée, car le domaine utilise cette mémoire.

La commande `ldm set-mem` définit la taille de la mémoire du domaine `ldom2`, qui est à l'état associé, à 690 Mo. Si vous ajoutez l'option `--auto-adj` à cette même commande, 78 Mo supplémentaires de mémoire sont ajoutés de manière dynamique à `ldom2` pour rester dans une limite de 256 Mo.

```
# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   SP      4     27392M  0.5%  1d 22h 55m
ldom1         active    -n----   5000    2     2G      0.2%  1d 1h 26m
ldom2         bound    ------  5001    2     256M
```

ldm set-mem 3400M primary

An `ldm set-mem 3400M` command would remove 23992MB, which is not a multiple of 256MB. Instead, run `ldm rm-mem 23808MB` to ensure a 256MB alignment.

ldm set-mem --auto-adj 3400M primary

Adjusting request size to 3.4G.

The primary domain has been allocated 184M more memory than requested because of memory alignment constraints. Only 9472M of memory could be removed from the primary domain because the rest of the memory is in use.

ldm set-mem 690M ldom2

```
# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   SP      4     17920M  0.5%  1d 22h 56m
ldom1         active    -n----   5000    2     2G      0.6%  1d 1h 27m
ldom2         bound    ------  5001    2     690M
```

ldm set-mem --auto-adj 690M ldom2

Adjusting request size to 256M.

The `ldom2` domain has been allocated 78M more memory than requested because of memory alignment constraints.

```
# ldm list
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv-   SP      4     17920M  2.1%  1d 22h 57m
ldom1         active    -n----   5000    2     2G      0.2%  1d 1h 27m
ldom2         bound    ------  5001    2     768M
```

Utilisation de la gestion de l'alimentation

Pour utiliser la gestion de l'alimentation (PM), vous devez d'abord définir le mode PM dans le microprogramme Oracle Integrated Lights Out Manager (ILOM) 3.0. Cette section récapitule les informations nécessaires afin de pouvoir utiliser le mode PM avec le logiciel Oracle VM Server for SPARC.

Pour plus d'informations sur ILOM, reportez-vous aux documents suivants :

- “Surveillance de la consommation d'énergie” dans le *Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide (Guide des procédures Sun Integrated Lights Out Manager (ILOM) 3.0)* (<http://d1c.sun.com/pdf/820-6412-12/820-6412-12.pdf>)
- *Oracle Integrated Lights Out Manager (ILOM) 3.0 Feature Updates and Release Notes (Notes de version et mises à jour des fonctions d'Oracle Integrated Lights Out Manager (ILOM) 3.0)* (<http://d1c.sun.com/pdf/820-7329-17/820-7329-17.pdf>)

Le mode d'alimentation régit l'utilisation de l'énergie du système à tout moment. À partir de la version 1.3 des domaines logiques, les modes d'alimentation suivants sont pris en charge, en supposant que la plate-forme sous-jacente a implémenté les fonctions PM :

- **Mode de performance.** Le système est autorisé à utiliser toute l'énergie disponible.
- **Mode élastique.** L'utilisation de l'énergie du système s'adapte au niveau d'utilisation du courant. Par exemple, l'état d'alimentation des ressources est réduit lorsque l'utilisation diminue.

Voici les fonctions PM :

- **Auto-désactivation des cœurs de CPU.** Le mode PM désactive automatiquement un cœur de CPU lorsque tous les brins de ce cœur ont été désactivés.
- **Saut du cycle d'horloge du CPU.** À partir de la version Oracle VM Server for SPARC 2.0, le mode PM peut ajuster automatiquement le saut du cycle d'horloge du CPU sur les plates-formes SPARC T3. L'ajustement peut augmenter ou diminuer le nombre de cycles d'horloge qui sont ignorés pour maintenir tous les domaines dans les seuils d'utilisation d'énergie. Le mode PM détermine s'il faut faire des ajustements en fonction de l'utilisation du CPU. Lorsque le système passe en mode de performance, le nombre de cycles d'horloge ignorés est automatiquement ajusté à aucun.
- **Opérations de mémoire en mode veille profonde.** À partir de la version Oracle VM Server for SPARC 2.0, lorsque les plates-formes SPARC T3 sont en mode élastique, elles configurent automatiquement la mémoire sous-utilisée pour fonctionner en mode veille profonde pour économiser de l'énergie.
- **limite de puissance.** À partir de la version Oracle VM Server for SPARC 2.0, vous pouvez définir une *limite de puissance* sur les plates-formes SPARC T3 pour restreindre l'énergie consommée par un système. Si l'énergie consommée est supérieure à la limite de puissance, les techniques PM sont utilisées pour réduire la consommation. Vous pouvez utiliser le processeur de service (SP) d'ILOM pour définir la limite de puissance.

Consultez les documents suivants :

- *Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide (Guide des procédures Sun Integrated Lights Out Manager (ILOM) 3.0)* (<http://d1c.sun.com/pdf/820-6412-12/820-6412-12.pdf>)
- *Oracle Integrated Lights Out Manager (ILOM) 3.0 Feature Updates and Release Notes (Notes de version et mises à jour des fonctions d'Oracle Integrated Lights Out Manager (ILOM) 3.0)* (<http://d1c.sun.com/pdf/820-7329-17/820-7329-17.pdf>)

Vous pouvez utiliser l'interface d'ILOM pour définir une limite de puissance, une période de grâce et une action de violation. Si la limite de puissance est dépassée pendant plus longtemps que la période de grâce, l'action de violation est effectuée.

Si la consommation d'énergie actuelle dépasse la limite de puissance, une tentative est faite pour réduire l'état d'alimentation des ressources en mode de gestion d'alimentation. Si la consommation d'énergie chute en dessous de la limite de puissance, l'état d'alimentation de ces ressources peut augmenter. Si le système est en mode élastique, une augmentation de l'état d'alimentation des ressources est générée par le niveau d'utilisation.

Lorsqu'un système est en mode élastique, certaines modifications de la configuration du domaine sont d'abord validées pour confirmer que la limite de puissance n'est pas dépassée. Si la limite de puissance est dépassée, seules certaines ressources peuvent être modifiées ou ajoutées comme demandé. Si la limite de puissance est augmentée ultérieurement, vous pouvez ensuite ajouter les ressources ayant été modifiées sans succès.

Si la charge du domaine fait que les ressources consomment plus d'énergie, seules les ressources pouvant maintenir la consommation inférieure à la limite de puissance sont alimentées avec succès.

Pour obtenir des instructions sur la configuration du mode d'alimentation à l'aide de la CLI du microprogramme ILOM 3.0, reportez-vous à la section "Surveillance de la consommation d'énergie" du *Sun Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide (Guide de procédures de la CLI de Sun Integrated Lights Out Manager (ILOM) 3.0)* (<http://d1c.sun.com/pdf/820-6412-12/820-6412-12.pdf>).

Liste des brins avec gestion de l'alimentation du CPU

Cette section montre comment répertorier les brins avec gestion de l'alimentation et les CPU virtuels.

▼ Liste des brins avec gestion de l'alimentation du CPU

- Répertoriez les brins avec gestion de l'alimentation à l'aide de l'une des commandes suivantes :

a. Utilisez la sous-commande `list -l`.

Dans la sortie, un tiret (---) dans la colonne UTIL du CPU signifie que le brin est en mode de gestion d'alimentation.

```
# ldm list -l primary
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    8     4G      4.3%  7d 19h 43m

SOFTSTATE
Solaris running

MAC
    00:14:4f:fa:ed:88

HOSTID
    0x84faed88

CONTROL
    failure-policy=ignore

DEPENDENCY
    master=

VCPU
    VID    PID    UTIL  STRAND
    0      0      0.0%  100%
    1      1      ---   100%
    2      2      ---   100%
    3      3      ---   100%
    4      4      ---   100%
    5      5      ---   100%
    6      6      ---   100%
    7      7      ---   100%
    ....
```

b. Utilisez l'option d'analyse (-p) avec la sous-commande `list -l`.

Dans la sortie, un espace après `util=` signifie que le brin est en mode de gestion d'alimentation.

```
# ldm list -l -p

VCPU
|vid=0|pid=0|util=0.7%|strand=100
|vid=1|pid=1|util= |strand=100
|vid=2|pid=2|util= |strand=100
|vid=3|pid=3|util= |strand=100
|vid=4|pid=4|util=0.7%|strand=100
|vid=5|pid=5|util= |strand=100
|vid=6|pid=6|util= |strand=100
|vid=7|pid=7|util= |strand=100
```

▼ Liste des CPU en mode de gestion de l'alimentation

- Répertoriez les CPU en mode de gestion d'alimentation à l'aide des commandes suivantes :

a. Utilisez la commande `ldm list-devices -a cpu`.

Dans la colonne PM de la sortie, un yes signifie que le CPU est en mode de gestion de l'alimentation, un no indique que le CPU est sous tension. Il est supposé que les CPU libres à 100 % sont en mode de gestion de l'alimentation par défaut, d'où le tiret (- - -) sous la colonne PM pour eux.

```
# ldm list-devices -a cpu
VCPU
  PID    %FREE    PM
  0      0        no
  1      0        yes
  2      0        yes
  3      0        yes
  4      100     - - -
  5      100     - - -
  6      100     - - -
  7      100     - - -
```

b. Utilisez l'option d'analyse `-p` sur la sous-commande `ldm list-devices -a cpu`.

Dans la colonne pm= de la sortie, un yes signifie que le CPU est en mode de gestion de l'alimentation, un no indique que le CPU est sous tension. Il est supposé que les CPU 100 % libres sont en mode de gestion de l'alimentation par défaut, d'où l'espace dans cette zone pour eux.

```
# ldm list-devices -a -p cpu
VERSION 1.4
VCPU
|pid=0|free=0|pm=no
|pid=1|free=0|pm=yes
|pid=2|free=0|pm=yes
|pid=3|free=0|pm=yes
|pid=4|free=0|pm=no
|pid=5|free=0|pm=yes
|pid=6|free=0|pm=yes
|pid=7|free=0|pm=yes
|pid=8|free=100|pm=
|pid=9|free=100|pm=
|pid=10|free=100|pm=
```

Utilisation de la gestion dynamique des ressources

À partir de la version 1.3 des domaines logiques, vous pouvez utiliser des stratégies pour déterminer comment effectuer automatiquement des activités de reconfiguration dynamique. À ce moment, vous pouvez *uniquement* créer des stratégies pour régir la gestion dynamique des ressources des CPU virtuels.



Attention – Les problèmes suivants affectent la gestion dynamique des ressources (DRM) CPU :

- Lorsque la gestion de l'alimentation est en mode élastique, la DRM ne peut pas être activée.
- Tout passage du mode de performance au mode élastique est retardé lorsque la DRM est activée.
- Vérifiez que vous avez désactivé la DRM CPU avant d'effectuer une opération de migration de domaine.
- Les stratégies DRM ne s'appliquent pas aux domaines configurés avec la contrainte `whole-core`.

Une *stratégie de gestion des ressources* définit dans quelles conditions des CPU virtuels peuvent être ajoutés automatiquement à un domaine logique et supprimés de celui-ci. Une stratégie est gérée à l'aide des commandes `ldm add-policy`, `ldm set-policy` et `ldm remove-policy` :

```
ldm add-policy [enable=yes|no] [priority=value] [attack=value] [decay=value]
  [elastic-margin=value] [sample-rate=value] [tod-begin=hh:mm[:ss]]
  [tod-end=hh:mm[:ss]] [util-lower=percent] [util-upper=percent] [vcpu-min=value]
  [vcpu-max=value] name=policy-name ldom...
ldm set-policy [enable=[yes|no]] [priority=[value]] [attack=[value]] [decay=[value]]
  [elastic-margin=[value]] [sample-rate=[value]] [tod-begin=[hh:mm:ss]]
  [tod-end=[hh:mm:ss]] [util-lower=[percent]] [util-upper=[percent]] [vcpu-min=[value]]
  [vcpu-max=[value]] name=policy-name ldom...
ldm remove-policy [name=policy-name... ldom
```

Pour plus d'informations sur ces commandes et sur la création de stratégies de gestion des ressources, reportez-vous à la page de manuel [ldm\(1M\)](#).

Une stratégie est en vigueur pendant la durée indiquée par les propriétés `tod-begin` et `tod-end`. La stratégie utilise la valeur de la propriété `priority` pour déterminer quelle stratégie utiliser si plusieurs stratégies sont appliquées en même temps.

La stratégie utilise les valeurs de la propriété `util-high` et `util-low` pour définir les seuils haut et bas d'utilisation du CPU. Si l'utilisation dépasse la valeur de `util-high`, des CPU virtuels sont ajoutés au domaine jusqu'à ce que le nombre soit compris entre les valeurs `vcpu-min` et `vcpu-max`. Si l'utilisation est inférieure à la valeur `util-low`, les CPU virtuels sont supprimés du domaine jusqu'à ce que le nombre soit compris entre les valeurs `vcpu-min` et `vcpu-max`. Si `vcpu-min` est atteint, plus aucun CPU virtuel ne peut être supprimé de manière dynamique. Si `vcpu-max` est atteint, plus aucun CPU virtuel ne peut être ajouté de manière dynamique.

EXEMPLE 10-4 Ajout de stratégies de gestion des ressources

Par exemple, après avoir observé l'utilisation classique de vos systèmes pendant plusieurs semaines, vous pouvez définir des stratégies pour optimiser l'utilisation des ressources. L'utilisation la plus importante s'effectue tous les jours entre 9 heures et 18 heures, heure du Pacifique, et l'utilisation la plus faible a lieu tous les jours de 18 heures à 9 heures.

EXEMPLE 10-4 Ajout de stratégies de gestion des ressources (Suite)

En fonction de cette observation de l'utilisation du système, vous décidez de créer les stratégies d'utilisation élevée et basse et fonction de l'utilisation globale de votre système :

- **Élevée** : Tous les jours de 9 heures à 18 heures, heure du Pacifique
- **Basse** : Tous les jours de 18 heures à 9 heures, heure du Pacifique

La commande `ldm add-policy` suivante crée la stratégie `high-usage` à utiliser au cours de la période d'utilisation élevée sur le domaine `ldom1`.

La stratégie `high-usage` suivant effectue les opérations suivantes :

- Indique que les heures de début et de fin sont 9 heures et 18 heures en définissant les propriétés `tod-begin` et `tod-end` respectivement.
- Indique que les limites inférieure et supérieure auxquelles effectuer une analyse de stratégie sont 25 % et 75 % en définissant les propriétés `util-lower` et `util-upper` respectivement.
- Indique que le nombre minimum et maximum de CPU virtuels est et 16 en définissant les propriétés `vcpu-min` et `vcpu-max` respectivement.
- Indique que le nombre maximum de CPU virtuels à ajouter au cours d'un cycle de contrôle des ressources est 1 en définissant la propriété `attack`.
- Indique que le nombre maximum de CPU virtuels à supprimer au cours d'un cycle de contrôle des ressources est 1 en définissant la propriété `decay`.
- Indique que la priorité de cette stratégie est 1 en définissant la propriété `priority`. Une priorité de 1 signifie que cette stratégie sera appliquée même si une autre stratégie est en vigueur.
- Indique que le nom du fichier de stratégie est `high-usage` en définissant la propriété `name`.
- Utilise les valeurs par défaut pour les propriétés qui ne sont pas définies, notamment `enable` et `sample-rate`. Reportez-vous à la page de manuel [ldm\(1M\)](#).

```
# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-upper=75 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=high-usage ldom1
```

La commande `ldm add-policy` suivante crée la stratégie `med-usage` à utiliser au cours de la période d'utilisation faible sur le domaine `ldom1`.

La stratégie `med-usage` suivant effectue les opérations suivantes :

- Indique que les heures de début et de fin sont 18 heures et 9 heures en définissant les propriétés `tod-begin` et `tod-end` respectivement.
- Indique que les limites inférieure et supérieure auxquelles effectuer une analyse de stratégie sont 10 % et 50 % en définissant les propriétés `util-lower` et `util-upper` respectivement.
- Indique que le nombre minimum et maximum de CPU virtuels est et 16 en définissant les propriétés `vcpu-min` et `vcpu-max` respectivement.

EXEMPLE 10-4 Ajout de stratégies de gestion des ressources (Suite)

- Indique que le nombre maximum de CPU virtuels à ajouter au cours d'un cycle de contrôle des ressources est 1 en définissant la propriété `attack`.
- Indique que le nombre maximum de CPU virtuels à supprimer au cours d'un cycle de contrôle des ressources est 1 en définissant la propriété `decay`.
- Indique que la priorité de cette stratégie est 1 en définissant la propriété `priority`. Une priorité de 1 signifie que cette stratégie sera appliquée même si une autre stratégie est en vigueur.
- Indique que le nom du fichier de stratégie est `high-usage` en définissant la propriété `name`.
- Utilise les valeurs par défaut pour les propriétés qui ne sont pas définies, notamment `enable` et `sample-rate`. Reportez-vous à la page de manuel [ldm\(1M\)](#).

```
# ldm add-policy tod-begin=18:00 tod-end=09:00 util-lower=10 util-upper=50 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=med-usage ldom1
```

Liste des ressources du domaine

Cette section montre l'utilisation de la syntaxe des sous-commandes `ldm`, définit certains termes de sortie, tels que les balises et les statistiques d'utilisation et fournit des exemples qui sont semblables à ce que vous verrez réellement en sortie.

Sortie lisible par la machine

Si vous créez des scripts utilisant la sortie de la commande `ldm list`, utilisez *toujours* l'option `-p` pour produire une formule lisible par la machine de la sortie. Reportez-vous à la section “Génération d'une liste analysable lisible par la machine (-p)” à la page 186 pour plus d'informations.

▼ Affichage de l'utilisation de la syntaxe des sous-commandes `ldm`

- Regardez l'utilisation de la syntaxe de toutes les sous-commandes `ldm`.

```
primary# ldm --help
```

Pour plus d'informations sur les sous-commandes `ldm`, reportez-vous à la page de manuel [ldm\(1M\)](#).

Définitions des balises

Les balises suivantes peuvent être affichées dans la sortie pour un domaine (`ldm list`). Si vous utilisez l'option longue analysable (`-l -p`) pour la commande, les balises sont écrites en entier, par exemple, `flags=normal, control, vio-service`. Sinon, vous voyez des abréviations en lettres, par exemple `-n-cv-`. Les valeurs des balises de la liste dépendent de leur position. Voici les valeurs pouvant apparaître dans chacune des six colonnes de gauche à droite.

Colonne 1

- s démarrage ou arrêt
- - paramètre substituable

Colonne 2

- n normal
- t transition

Colonne 3

- d reconfiguration retardée
- r reconfiguration dynamique (DR) de la mémoire
- - paramètre substituable

Colonne 4

- c domaine de contrôle
- - paramètre substituable

Colonne 5

- v domaine de service d'E/S virtuelle
- - paramètre substituable

Colonne 6

- s domaine source dans une migration
- t domaine cible dans une migration
- e erreur survenue au cours d'une migration
- - paramètre substituable

Utilisation de la définition des statistiques

Les statistiques d'utilisation des CPU virtuels (UTIL) sont affichées sur l'option longue (`-l`) de la commande `ldm list`. Les statistiques sont le pourcentage de temps que le CPU virtuel a passé sur l'exécution pour le compte d'un système d'exploitation invité. Un CPU virtuel est considéré comme un exécutant pour le compte du système d'exploitation invité sauf lorsqu'il a été cédé à

l'hyperviseur. Si le système d'exploitation invité ne cède pas de CPU virtuels à l'hyperviseur, l'utilisation des CPU dans le système d'exploitation invité sera toujours affiché comme étant à 100%.

Les statistiques d'utilisation indiquées pour un domaine logique sont la moyenne des utilisations des CPU virtuels dans le domaine. Un tiret (- -) dans la colonne UTIL signifie que le brin est en mode de gestion de l'alimentation.

Affichage des différentes listes

▼ Affichage des versions logicielles (-V)

- Affiche la version logicielle actuellement installée.

```
primary# ldm -V
```

▼ Génération d'une liste abrégée

- Génère une liste abrégée de tous les domaines.

```
primary# ldm list
```

▼ Génération d'une liste longue (-l)

- Générez une liste longue pour tous les domaines.

```
primary# ldm list -l
```

▼ Génération d'une liste étendue (-e)

- Générez une liste étendue de tous les domaines.

```
primary# ldm list -e
```

▼ Génération d'une liste analysable lisible par la machine (-p)

- Génère une liste analysable lisible par la machine de tous les domaines.

```
primary# ldm list -p
```

▼ Génération d'un sous-ensemble d'une liste longue (-o *format*)

- Générez une sortie comme un sous-ensemble de ressources en entrant une ou plusieurs des options *format* suivantes. Si vous indiquez plusieurs formats, délimitez les éléments par des virgules sans espaces.

```
primary# ldm list -o resource[,resource...] ldom
```

- `console` – La sortie contient la console virtuelle (`vcons`) et le service de concentrateur de console virtuelle (`vcc`)
- `core` – La sortie contient des informations sur les domaines ayant des cœurs complets alloués
- `cpu` – La sortie contient des informations sur le CPU virtuel (`vcpu`), le CPU physique (`pcpu`) et l'ID du cœur
- `crypto` – La sortie de l'unité cryptographique contient une unité arithmétique modulaire (`mau`) et toute autre unité cryptographique prise en charge par les LDoms, notamment la Control Word Queue (`CWQ`)
- `disk` – La sortie contient le disque virtuel (`vdisk`) et le serveur de disque virtuel (`vds`)
- `domain` – La sortie contient des variables (`var`), host ID (`hostid`), l'état du domaine, des balises, l'UUID et l'état du logiciel
- `memory` – La sortie contient `memory`
- `network` – La sortie contient l'adresse de contrôle d'accès au média (`mac`), le commutateur de réseau virtuel (`vsw`) et, le périphérique de réseau virtuel (`vnet`)
- `physio` – L'entrée/sortie physique contient l'interconnexion de composants périphériques (`pci`) et l'unité d'interface réseau (`niu`)
- `resgmt` – La sortie contient des informations sur la stratégie DRM (gestion dynamique des ressources), indique quelle stratégie est actuellement utilisée et répertorie les contraintes associées à la configuration `whole-core`
- `serial` – La sortie contient le service de canal de domaine logique (`vldc`), le client du canal de domaine logique (`vldcc`), le client du canal de plan de données virtuelles (`vdpsc`), le service du canal de plan de données virtuelles (`vdpcs`)
- `stats` – La sortie contient des statistiques relatives aux stratégies de gestion des ressources
- `status` – La sortie contient l'état sur la migration du domaine en cours

Les exemples suivantes montrent différents sous-ensembles de sortie que vous pouvez définir :

- Liste des informations des CPU pour le domaine de contrôle


```
# ldm list -o cpu primary
```
- Liste des informations de domaine pour un domaine invité


```
# ldm list -o domain ldm2
```
- Liste des informations sur la mémoire et le réseau pour un domaine invité


```
# ldm list -o network,memory ldm1
```
- Liste des informations sur la stratégie DRM pour un domaine invité


```
# ldm list -o resgmt,stats ldm1
```

▼ Liste des variables

- Affiche une variable et sa valeur pour un domaine.

```
primary# ldm list-variable variable-name ldom
```

Par exemple, la commande suivante affiche la valeur de la variable `boot-device` sur le domaine `ldg1`:

```
primary# ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

▼ Liste des associations

- Répertoriez les ressources associées à un domaine.

```
primary# ldm list-bindings ldom
```

▼ Liste des configurations

- Répertoriez les configurations de domaine logique ayant été stockées sur le SP.

Exemple 10-5 Liste des configurations

La commande `ldm list-config` répertorie les configurations de domaine logique qui sont stockées sur le processeur de service. Si l'option `-r` est utilisée, cette commande répertorie les configurations pour lesquelles des fichiers enregistrés automatiquement existent sur le domaine de contrôle.

Pour plus d'informations sur les configurations, reportez-vous à la section “[Gestion des configurations des domaines logiques](#)” à la page 194. Pour plus d'exemples, reportez-vous à la page de manuel [ldm\(1M\)](#).

```
primary# ldm list-config
factory-default
3guests
foo [next poweron]
primary
reconfig-primary
```

Informations supplémentaires

Signification des libellés

Les libellés à la droite du nom de la configuration ont la signification suivante :

- `[current]` – Dernière configuration démarrée, uniquement tant qu'elle correspond à la configuration en cours d'exécution, c'est-à-dire jusqu'à ce que vous démarreriez une reconfiguration. Après la reconfiguration, the annotation devient `[next poweron]`.
- `[next poweron]` – Configuration à utiliser à la prochaine remise sous tension.

▼ Liste des périphériques

- Répertoriez toutes les ressources du serveur, associées ou non.

```
primary# ldm list-devices -a
```

▼ Liste de la mémoire disponible

- Répertoriez la quantité de mémoire disponible à allouer.

```
primary# ldm list-devices mem
MEMORY
  PA                SIZE
  0x14e000000      2848M
```

▼ Liste des services

- Répertoriez les services disponibles.

```
primary# ldm list-services
```

Liste des contraintes

Les contraintes du gestionnaire de domaines logiques sont une ou plusieurs ressources que vous voulez assigner à un domaine particulier. Soit vous recevez toutes les ressources dont vous avez demandé l'ajout au domaine, soit vous ne recevez aucune d'entre elles, en fonction des ressources disponibles. La sous-commande `list-constraints` répertorie les ressources que vous voulez assigner au domaine.

▼ Liste des contraintes pour un domaine

- Répertoriez les contraintes pour un domaine.

```
primary# ldm list-constraints ldom
```

▼ Liste des contraintes au format XML

- Répertoriez les contraintes au format XML pour un domaine particulier.

```
primary# ldm list-constraints -x ldom
```

▼ Liste des contraintes dans un format lisible par la machine.

- Répertoriez les contraintes pour tous les domaines dans un format analysable.

```
primary# ldm list-constraints -p
```


Gestion des configurations

Ce chapitre contient des informations sur la gestion des configurations de domaine.

Ce chapitre aborde les sujets suivants :

- “Enregistrement des configurations de domaine pour régénération ultérieure” à la page 191
- “Gestion des configurations des domaines logiques” à la page 194

Enregistrement des configurations de domaine pour régénération ultérieure

Le processus basique consiste à enregistrer les informations de contraintes de ressource pour chaque domaine dans un fichier XML, qui peut être renvoyé au gestionnaire de domaines logiques, par exemple, après une panne matérielle pour régénérer la configuration souhaitée.

“Restauration d'une configuration de domaine à partir d'un fichier XML (`ldm add-domain`)” à la page 192 fonctionne pour les domaines invités, mais pas pour le domaine de contrôle (`primary`). Vous pouvez enregistrer les contraintes du domaine `primary` dans un fichier XML, mais vous ne pouvez pas réintégrer le fichier dans la commande `ldm add-domain -i`. Cependant, vous pouvez utiliser la commande `ldm init-system` et les contraintes de ressource du fichier XML pour reconfigurer votre domaine `primary`. Vous pouvez également utiliser la commande `ldm init-system` pour reconfigurer d'autres domaines décrits dans le fichier XML, mais ces derniers resteront inactifs lorsque la configuration sera terminée.

La méthode suivante ne conserve pas les liaisons actuelles, mis uniquement les contraintes utilisées pour créer ces liaisons. Cela signifie, qu'après cette procédure, les domaines auront les mêmes ressources virtuelles, mais qu'ils ne seront pas nécessairement liés aux mêmes ressources physiques.

▼ Enregistrement des configurations de domaine

Cette procédure décrit comment enregistrer une configuration de domaine pour un domaine unique ou pour tous les domaines sur un système.

- **Enregistrez la configuration de domaine pour un ou plusieurs domaines.**

- **Pour enregistrer la configuration pour un seul domaine, créez un fichier XML contenant les contraintes du domaine.**

```
# ldm list-constraints -x ldom >ldom.xml
```

L'exemple suivant indique comment créer un fichier XML, `ldg1.xml`, qui contient les contraintes du domaine `ldg1` :

```
# ldm list-constraints -x ldg1 >ldg1.xml
```

- **Pour enregistrer les configurations pour tous les domaines sur un système, créez un fichier XML contenant les contraintes de tous les domaines.**

```
# ldm list-constraints -x >file.xml
```

L'exemple suivant indique comment créer un fichier XML, `config.xml`, contenant les contraintes de tous les domaines sur un système :

```
# ldm list-constraints -x >config.xml
```

▼ Restauration d'une configuration de domaine à partir d'un fichier XML (`ldm add-domain`)

Au lieu de cette procédure, vous pouvez utiliser la commande `ldm init-system` pour restaurer les configurations de domaine à partir d'un fichier XML. Reportez-vous à la section [“Restauration d'une configuration de domaine à partir d'un fichier XML \(`ldm init-system`\)”](#) à la page 193.

- 1 **Créez le domaine à l'aide du fichier XML créé en entrée.**

```
# ldm add-domain -i ldom.xml
```

- 2 **Associez le domaine.**

```
# ldm bind-domain ldom
```

- 3 **Démarrez le domaine.**

```
# ldm start-domain ldom
```

Exemple 11-1 Restauration d'un seul domaine à partir d'un fichier XML

L'exemple suivant décrit comment restaurer un domaine unique. En premier lieu, vous restaurez le domaine `ldg1` à partir du fichier XML. Ensuite, vous associez et démarrez le domaine `ldg1` que vous avez restauré.

```
# ldm add-domain -i ldg1.xml
# ldm bind ldg1
# ldm start ldg1
```

▼ Restauration d'une configuration de domaine à partir d'un fichier XML (`ldm init-system`)

Cette procédure explique comment utiliser la commande `ldm init-system` avec un fichier XML pour recréer la configuration enregistrée précédemment. Le fichier XML décrit une ou plusieurs configurations de domaine. Le fichier XML peut être créé en exécutant la commande `ldm ls-constraints -x`. La commande `ldm init-system` doit être exécutée dans la configuration `factory-default`, mais elle ne peut pas restaurer de configuration à partir du fichier XML. Le domaine `primary` est reconfiguré comme indiqué dans le fichier, et tous les domaines autres que `primary` ayant des configurations dans le fichier XML sont reconfigurés mais laissés inactifs.

Au lieu de cette procédure, vous pouvez utiliser la commande `ldm add-domain` pour restaurer la configuration d'un seul domaine à partir d'un fichier XML. Reportez-vous à la section [“Restauration d'une configuration de domaine à partir d'un fichier XML \(`ldm add-domain`\)”](#) à la page 192.

- 1 **Connectez-vous au domaine `primary`.**
- 2 **Vérifiez que le système est en configuration `factory-default`.**

```
primary# ldm list-config | grep "factory-default"
factory-default [current]
```

Si le système n'est pas en configuration `factory-default`, reportez-vous à la section [“Restauration de la configuration usine par défaut”](#) à la page 38.

- 3 **Devenez un superutilisateur ou assumez un rôle équivalent.**
Les rôles contiennent des autorisations et des commandes nécessitant des droits d'accès. Pour plus d'informations sur les rôles, reportez-vous à la section [“Configuring RBAC \(Task Map\)”](#) du *System Administration Guide: Security Services*.
- 4 **Restaurez la ou les configurations de domaine à partir d'un fichier XML.**

```
# ldm init-system [-rs] -i filename.xml
```

L'option `-r` redémarre le domaine `primary` après la configuration. Si vous ne spécifiez pas l'option `-r`, vous devez effectuer un redémarrage manuel. L'option `-s` restaure uniquement la configuration des services virtuels (`vds`, `vcc` et `vsw`) et peut être effectuée sans redémarrage.

Exemple 11-2 Restauration de domaines à partir de fichiers de configuration XML

Les exemples suivants décrivent comment utiliser la commande `ldm init-system` pour restaurer le domaine `primary` et tous les domaines sur un système à partir de la configuration `factory-default`.

- **Restaurer le domaine `primary`.** L'option `-r` est utilisée pour redémarrer le domaine `primary` à la fin de la configuration. Le fichier `primary.xml` contient la configuration de domaine XML que vous avez enregistrée précédemment.

```
primary# ldm init-system -r -i primary.xml
```

- **Restaurer tous les domaines sur un système.** Restaurez les domaines sur le système à la configuration contenue dans le fichier XML `config.xml`. Le fichier `config.xml` contient les configurations de domaine XML que vous avez enregistrées précédemment. Le domaine `primary` est redémarré automatiquement par la commande `ldm init-system`. Tous les autres domaines sont restaurés, mais ne sont pas associés, ni redémarrés.

```
# ldm init-system -r -i config.xml
```

Après le redémarrage du système, les commandes suivantes associent et redémarrent les domaines `ldg1` et `ldg2` :

```
# ldm bind ldg1
# ldm start ldg1
# ldm bind ldg2
# ldm start ldg2
```

Gestion des configurations des domaines logiques

Une *configuration* de domaines logiques est une description complète de tous les domaines et de toutes leurs allocations de ressource au sein d'un seul système. Vous pouvez enregistrer et stocker les configurations sur le processeur de service (SP) pour une utilisation ultérieure.

Lorsque vous mettez un système sous tension, le SP démarre la configuration sélectionnée. En initialisant une configuration, le système exécute le même ensemble de domaines et utilise les mêmes allocations de virtualisation et de partitionnement que celles indiquées dans la configuration. La configuration par défaut est celle qui a été enregistrée le plus récemment.

À partir de la version 1.2 des domaines logiques, une copie de la configuration actuelle est automatiquement enregistrée sur le domaine de contrôle à chaque fois que la configuration des domaines logiques est modifiée.

L'opération d'enregistrement automatique est effectuée immédiatement, même dans les situations suivantes :

- Lorsque la nouvelle configuration n'est pas explicitement enregistrée sur le SP
- Lorsque la modification de la configuration réelle n'est pas effectué tant que le domaine concerné n'a pas redémarré

L'opération d'enregistrement automatique vous permet de récupérer une configuration lorsque les configurations enregistrées sur le SP sont perdues. Cette opération vous permet également de récupérer une configuration lorsque la configuration actuelle n'a pas été explicitement enregistrée sur le SP lorsque le système a été remis sous tension. Dans ce cas, le gestionnaire de domaines logiques peut restaurer cette configuration au redémarrage si elle est plus récente que la configuration sélectionnée pour le redémarrage suivant.

Remarque – Les événements de mise à jour de gestion de l'alimentation, FMA, ASR et PRI ne provoquent pas une mise à jour des fichiers enregistrés automatiquement.

Vous pouvez restaurer automatiquement ou manuellement les fichiers enregistrés automatiquement sur des configurations nouvelles ou existantes. Par défaut, lorsqu'une configuration enregistrée automatiquement est plus récente que la configuration correspondante en cours d'exécution, un message est écrit dans le journal des domaines logiques. Vous devez donc utiliser la commande `ldm add-spconfig -r` pour mettre à jour manuellement une configuration existante ou en créer une nouvelle en fonction des données enregistrées automatiquement.

Remarque – Lorsque qu'une reconfiguration retardée est en attente, les modifications de configuration sont immédiatement enregistrées automatiquement. Par conséquent, si vous exécutez la commande `ldm list-config -r`, la configuration enregistrée automatiquement apparaît comme plus récente que la configuration actuelle.

Pour plus d'informations sur l'utilisation des commandes `ldm *-spconfig` pour gérer les configurations et récupérer manuellement les fichiers enregistrés automatiquement, reportez-vous à la page de manuel [ldm\(1M\)](#).

Pour plus d'informations sur la procédure de sélection d'une configuration de démarrage, reportez-vous à la section [“Utilisation des domaines logiques avec le processeur de service”](#) à la page 204.

▼ Modification de la stratégie de récupération automatique

La stratégie de récupération automatique définit comment traiter la récupération d'une configuration lorsqu'une configuration automatiquement enregistrée sur le domaine de contrôle est plus récente que la configuration en cours d'exécution correspondante. La stratégie de récupération automatique est spécifiée en définissant la propriété `autorecovery_policy` du service SMF `ldmd`. La propriété `autorecovery_policy` peut avoir les valeurs suivantes :

- `autorecovery_policy=1` – Consigne tous les messages d'avertissement lorsqu'une configuration enregistrée automatiquement est plus récente que la configuration en cours d'exécution correspondante. Ces messages sont consignés dans le fichier journal SMF `ldmd`. L'utilisateur peut effectuer manuellement toute récupération de configuration. Il s'agit de la stratégie par défaut.
- `autorecovery_policy=2` – Affiche un message de notification si une configuration enregistrée automatiquement est plus récente que la configuration en cours d'exécution correspondante. Ce message de notification est imprimé dans la sortie des commandes `ldm` la première fois qu'une commande `ldm` est émise après chaque redémarrage du gestionnaire de domaines logiques. L'utilisateur doit effectuer manuellement la récupération de la configuration.
- `autorecovery_policy=3` – Met automatiquement à jour la configuration si une configuration enregistrée automatiquement est plus récente que la configuration en cours d'exécution correspondante. Cette action remplace la configuration du SP qui sera utilisée à la prochaine remise sous tension. Cette configuration est mise à jour avec la nouvelle configuration qui est enregistrée sur le domaine de contrôle. Cette action n'a aucune incidence sur la configuration actuellement en cours d'exécution. Elle a uniquement une incidence sur la configuration qui sera utilisée à la prochaine remise sous tension. Un message est également consigné. Celui-ci indique qu'une configuration plus récente a été enregistrée sur le SP et qu'elle sera initialisée à la prochaine remise sous tension du système. Ces messages sont consignés dans le fichier journal SMF `ldmd`.

1 Connectez-vous au domaine de contrôle.

2 Devenez un superutilisateur ou assumez un rôle équivalent.

Les rôles contiennent des autorisations et des commandes nécessitant des droits d'accès. Pour plus d'informations sur les rôles, reportez-vous à la section "[Configuring RBAC \(Task Map\)](#)" du *System Administration Guide: Security Services*.

3 Consultez la valeur de la propriété `autorecovery_policy`.

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
```

4 Arrêtez le service `ldmd`.

```
# svcadm disable ldmd
```

5 Modifiez la valeur de la propriété `autorecovery_policy`.

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=value
```

Par exemple, pour faire que la stratégie effectue une récupération automatique, définissez la valeur de la propriété sur 3 :

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
```

6 Actualisez et redémarrez le service `ldmd`.

```
# svcadm refresh ldmd
# svcadm enable ldmd
```

Exemple 11-3 Modification de la stratégie de récupération automatique à partir du journal de récupération automatique

L'exemple suivant décrit comment consulter la valeur actuelle de la propriété `autorecovery_policy` et la remplacer par une nouvelle valeur. La valeur d'origine de cette propriété est 1, ce qui signifie que les modifications enregistrées automatiquement sont consignées. La commande `svcadm` permet d'arrêter et de redémarrer le service `ldmd`, et la commande `svccfg` permet de consulter et de définir la valeur de la propriété.

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
ldmd/autorecovery_policy integer 1
# svcadm disable ldmd
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
# svcadm refresh ldmd
# svcadm enable ldmd
```


Réalisation d'autres tâches d'administration

Ce chapitre contient des informations et des tâches sur l'utilisation du logiciel Oracle VM Server for SPARC qui ne sont pas décrites dans les chapitres précédents.

Ce chapitre aborde les sujets suivants :

- “Entrée de noms dans la CLI” à la page 199
- “Connexion à une console invitée sur un réseau” à la page 200
- “Utilisation de groupes de consoles” à la page 201
- “Délai d'arrêt d'un domaine fortement chargé pouvant être dépassé” à la page 202
- “Utilisation du SE Oracle Solaris avec Oracle VM Server for SPARC” à la page 202
- “Utilisation des domaines logiques avec le processeur de service” à la page 204
- “Configuration des dépendances de domaine” à la page 205
- “Détermination de l'endroit où les erreurs sont survenues lors du mappage du CPU et des adresses de mémoire” à la page 209
- “Utilisation des identificateurs uniques universellement” à la page 212
- “Commande et API d'information sur le domaine virtuel” à la page 212

Entrée de noms dans la CLI

Les sections suivantes décrivent les restrictions dans l'entrée des noms dans la CLI du gestionnaire de domaines logiques.

Noms de fichier (*file*) et noms de variable (*var-name*)

- Le premier caractère doit être une lettre, un nombre ou une barre oblique (/).
- Les lettres suivantes doivent être des lettres, des nombres ou des signes de ponctuation.

backend du serveur de disque virtuel et noms de périphérique de commutateur virtuel

Les noms doivent contenir des lettres, des nombres ou des signes de ponctuation.

Nom de configuration (*config-name*)

Le nom de la configuration de domaine logique (*config-name*) que vous assignez à une configuration stockée sur le processeur de service (SP) doit comporter 64 caractères au maximum.

Tous les autres noms

Le reste des noms, notamment le nom du domaine logique (*ldom*), les noms de service (*vswitch-name*, *service-name*, *vdpcs-service-name* et *vcc-name*), le nom du réseau virtuel (*if-name*) et le nom du disque virtuel (*disk-name*) doivent présenter le format suivant :

- Le premier caractère doit être une lettre ou un nombre.
- Les caractères suivants doivent être des lettres, des nombres ou l'un des caractères suivants `-_+#. : ; ~ ()`.

Connexion à une console invitée sur un réseau

Vous pouvez connecter une console invitée sur un réseau si la propriété `listen_addr` est définie sur l'adresse IP du domaine de contrôle dans le manifeste SMF `vntsd(1M)`. Par exemple :

```
$ telnet host-name 5001
```

Remarque – L'activation de l'accès réseau à une console a des implications de sécurité. N'importe quel utilisateur peut se connecter à une console et pour cette raison, elle est désactivée par défaut.

Un manifeste d'utilitaire de gestion des services est un fichier XML qui décrit un service. Pour plus d'informations sur la création d'un manifeste SMF, reportez-vous à l'[ensemble de documents administrateur système Solaris 10](http://docs.sun.com/app/docs/coll/47.16) (<http://docs.sun.com/app/docs/coll/47.16>).

Remarque – Pour accéder à un SE non anglais dans un domaine invité via la console, le terminal de la console doit être dans les paramètres linguistiques requis par le SE.

Utilisation de groupes de consoles

Le démon du serveur de terminal de réseau virtuel, `vntsd(1M)`, vous permet de fournir l'accès à plusieurs consoles du domaine à l'aide du port TCP. Après la création du domaine, le gestionnaire de domaines logiques assigne un port TCP unique à chaque console en créant un nouveau groupe par défaut pour cette console du domaine. Le port TCP est ensuite assigné au groupe de consoles au contraire de la console elle-même. La console peut être associée à un groupe existant à l'aide de la sous-commande `set -vcons`.

▼ Association de plusieurs consoles en un groupe

1 Associez les consoles pour les domaines en un groupe unique.

L'exemple suivant présente l'association de la console pour trois domaines différents (`ldg1`, `ldg2` et `ldg3`) au même groupe de consoles (`group1`).

```
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg1
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg2
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg3
```

2 Connectez le port TCP associé (`localhost` sur le port `5000` dans cet exemple).

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
```

Vous êtes invité à sélectionner une des consoles du domaine.

3 Répertoriez les domaines dans le groupe en sélectionnant `l` (liste).

```
primary-vnts-group1: h, l, c{id}, n{name}, q: l
DOMAIN ID      DOMAIN NAME      DOMAIN STATE
0               ldg1             online
1               ldg2             online
2               ldg3             online
```

Remarque – Pour réassigner la console à un groupe différent ou à une instance `vcc`, le domaine doit être dissocié, c'est-à-dire qu'il doit être à l'état inactif. Reportez-vous à la page de manuel `vntsd(1M)` du SE Oracle Solaris pour plus d'informations sur la configuration et l'utilisation de SMF pour gérer `vntsd` et l'utilisation des groupes de console.

Délai d'arrêt d'un domaine fortement chargé pouvant être dépassé

Le délai d'une commande `ldm stop-domain` peut être dépassé avant que le domaine ne termine l'arrêt. Lorsque cela se produit, une erreur similaire à la suivante est renvoyée par le gestionnaire de domaines logiques.

```
LDom ldg8 stop notification failed
```

Cependant, le domaine peut toujours traiter la demande d'arrêt. Utilisez la commande `ldm list-domain` pour vérifier l'état du domaine. Par exemple :

```
# ldm list-domain ldg8
NAME      STATE  FLAGS  CONS  VCPU MEMORY  UTIL  UPTIME
ldg8      active s---- 5000   22  3328M  0.3% 1d 14h 31m
```

La liste précédente montre le domaine comme actif, mais l'indicateur `s` indique que le domaine est en cours d'arrêt. Il doit s'agir d'un état transitoire.

L'exemple suivante montre que le domaine s'est maintenant arrêté.

```
# ldm list-domain ldg8
NAME      STATE  FLAGS  CONS  VCPU MEMORY  UTIL  UPTIME
ldg8      bound  ----- 5000   22  3328M
```

Utilisation du SE Oracle Solaris avec Oracle VM Server for SPARC

Cette section décrit les modifications de comportement du SE Oracle Solaris survenant une fois qu'une configuration a été créée par le gestionnaire de domaines logiques a été instanciée.

Microprogramme OpenBoot indisponible une fois que le SE Oracle Solaris a démarré

Le microprogramme OpenBoot n'est pas disponible après le démarrage du SE Oracle Solaris, car il a été déchargé de la mémoire.

Pour atteindre l'invite `ok` à partir du SE Oracle Solaris, vous devez arrêter le domaine. Pour ce faire, vous pouvez utiliser la commande `halt` du SE Oracle Solaris.

Cycle d'arrêt et de redémarrage d'un serveur

Lorsque vous effectuez de la maintenance sur un système exécutant le logiciel Oracle VM Server for SPARC nécessitant un cycle d'arrêt et de redémarrage du serveur, vous devez d'abord enregistrer les configurations actuelles des domaines logiques sur le SP.

▼ Enregistrement des configurations actuelles des domaines logiques sur le SP

- Utilisez la commande suivante.

```
# ldm add-config config-name
```

N'utilisez pas la commande `ps radm(1M)` sur les CPU actifs dans domaine avec gestion de l'alimentation

N'essayez pas de modifier l'état opérationnel d'un CPU actif dans un domaine avec gestion de l'alimentation à l'aide de la commande `ps radm(1M)`.

Résultat des interruptions du SE Oracle Solaris

Le comportement décrit dans cette section est visible lorsque vous effectuez les opérations suivantes :

1. Appuyer sur la séquence de touches L1-A lorsque le périphérique d'entrée est défini sur keyboard.
2. Entrer la commande `send break` lorsque la console est à l'invite `telnet`.

Après ces types d'interruptions, vous recevez l'invite suivante :

```
c)ontinue, s)ync, r)eset, h)alt?
```

Tapez la lettre représentant ce que vous voulez que le système fasse après ces types d'interruptions.

Résultats de l'arrêt ou de la réinitialisation du domaine de contrôle

Le tableau suivant indique le comportement attendu suite à l'arrêt ou à la réinitialisation du domaine (`primary`) de contrôle.

TABLEAU 12-1 Comportement attendu suite à l'arrêt ou à la réinitialisation du domaine (primary) de contrôle

Commande	Autre domaine configuré ?	Comportement
halt	Non configuré	L'hôte est hors tension et le reste tant qu'il n'est pas remis sous tension au niveau du SP.
	Configuré	Le logiciel redémarre et s'initialise si la variable <code>auto-boot?=true</code> . Le logiciel se réinitialise et s'arrête à l'invite <code>ok</code> si la variable <code>auto-boot?=false</code> .
reboot	Non configuré	Redémarre l'hôte sans mise hors tension.
	Configuré	Redémarre l'hôte sans mise hors tension.
shutdown -i 5	Non configuré	L'hôte est hors tension et le reste tant qu'il n'est pas remis sous tension au niveau du SP.
	Configuré	Le logiciel se réinitialise et redémarre.

Pour plus d'informations sur les conséquences du redémarrage d'un domaine de contrôle ayant le rôle de domaine racine, voir [“Redémarrage du domaine primary” à la page 75](#).

Utilisation des domaines logiques avec le processeur de service

Cette section décrit les informations à connaître lors de l'utilisation du processeur de service (SP) Integrated Lights Out Manager (ILOM) avec le gestionnaire de domaines logiques. Pour plus d'informations sur l'utilisation du logiciel ILOM, consultez les documents correspondant à votre plate-forme, notamment l'[Sun SPARC Enterprise T5120 and T5220 Servers Topic Set](#) pour les serveurs Sun SPARC Enterprise T5120 et T5220.

Une option supplémentaire est disponible dans la commande ILOM existante.

```
-> set /HOST/bootmode config=config-name
```

L'option `config=config-name` vous permet de définir la configuration à la prochaine remise sous tension sur une autre configuration, comprenant la configuration d'expédition `factory-default`.

Vous pouvez appeler la commande que l'hôte soit sous tension ou non. Elle est appliquée au prochain redémarrage ou à la prochaine réinitialisation de l'hôte.

▼ Réinitialisation de la configuration du domaine à la configuration par défaut ou à une configuration différente

- Réinitialisez la configuration du domaine logique à la prochaine mise sous tension à la configuration d'expédition par défaut en exécutant cette commande :

```
-> set /HOST/bootmode config=factory-default
```

Vous pouvez également sélectionner d'autres configurations ayant été créées avec le gestionnaire de domaines logiques à l'aide de la commande `ldm add-config` et stockées sur le processeur de service (SP). Le nom que vous indiquez dans la commande `ldm add-config` du &Manager peut être utilisé pour sélectionner cette configuration avec la commande `bootmode` d'ILOM. Par exemple, supposons que vous stockiez la configuration portant le nom `ldm-config1`.

```
-> set /HOST/bootmode config=ldm-config1
```

Vous devez maintenant remettre le système sous tension pour charger la nouvelle configuration.

Reportez-vous à la page de manuel [ldm\(1M\)](#) pour plus d'informations sur la commande `ldm add-config`.

Configuration des dépendances de domaine

Vous pouvez utiliser le &Manager pour établir des relations de dépendance entre les domaines. Un domaine ayant un ou plusieurs domaines dépendant de lui-même est appelé un *domaine maître*. Un domaine dépendant d'un autre domaine est appelé un *domaine esclave*.

Chaque domaine esclave peut définir jusqu'à quatre domaines maîtres en définissant la propriété `master`. Par exemple, le domaine esclave `pine` définit quatre domaines maîtres dans la liste séparée par des virgules suivante :

```
# ldm add-domain master=apple,lemon,orange,peach pine
```

Chaque domaine maître peut définir ce qui se produit sur ses domaines esclaves en cas de panne du domaine maître. Par exemple, si un domaine maître est en panne, il peut demander à ses domaines esclaves de paniquer. Si un domaine esclave a plusieurs domaines maîtres, le premier domaine maître à être en panne déclenche la stratégie de panne qu'il a définie sur tous ses domaines esclaves.

Remarque – Si plusieurs domaines maîtres tombent en panne en même temps, seul une des stratégies de panne définies sera appliquée sur tous les domaines esclaves affectés. Par exemple, si le domaine maître en panne a des stratégies de panne `stop` et `panic`, tous les domaines esclaves s'arrêteront ou paniqueront.

La stratégie de panne du domaine maître est contrôlée en définissant l'une des valeurs suivantes dans la propriété `failure-policy` :

- `ignore` ignore tous les domaines esclaves lorsque le domaine maître est en panne.
- `panic` panique tous les domaines esclaves lorsque le domaine maître est en panne.
- `reset` réinitialise tous les domaines esclaves lorsque le domaine maître est en panne.
- `stop` arrête tous les domaines esclaves lorsque le domaine maître est en panne.

Dans cet exemple, les domaines maîtres définissent leur stratégie de panne comme suit :

```
# ldm set-domain failure-policy=ignore apple
# ldm set-domain failure-policy=panic lemon
# ldm set-domain failure-policy=reset orange
# ldm set-domain failure-policy=stop peach
```

Vous pouvez utiliser ce mécanisme pour créer des dépendances explicites entre les domaines. Par exemple, un domaine invité dépend implicitement du domaine de service devant lui fournir ses périphériques virtuels. L'E/S d'un domaine invité est bloquée lorsque le domaine de service duquel il dépend n'est pas en état de fonctionnement. En définissant un domaine invité en tant qu'esclave de son domaine de service, vous pouvez définir le comportement du domaine invité lorsque son domaine de service est en panne. Lorsque cette dépendance n'est pas établie, un domaine invité attend simplement la reprise du service de son domaine de service.

Remarque – Le gestionnaire de domaines logiques ne vous autorise pas à créer des relations de domaine qui créent un cycle de dépendance. Pour plus d'informations, voir [“Cycles de dépendance” à la page 208](#).

Pour obtenir des exemples XML de dépendance de domaine, voir l'[Exemple D-6](#).

Exemples de dépendance de domaine

Les exemples suivants montrent comment configurer les dépendances de domaine.

- La première commande crée un domaine maître appelé `twizzle`. Cette commande utilise `failure-policy=reset` pour indiquer que les domaines esclaves se réinitialisent si le domaine `twizzle` est en panne. La seconde commande modifie un domaine maître appelé `primary`. Cette commande utilise `failure-policy=panic` pour indiquer que les domaines esclaves paniquent si le domaine `primary` est en panne. La troisième commande crée un domaine esclave appelé `chocktaw` qui dépend de deux domaines maîtres, `twizzle` et `primary`. Le domaine esclave utilise `master=twizzle,primary` pour définir ses domaines maîtres. Si le domaine `twizzle` ou `primary` est en panne, le domaine `chocktaw` se réinitialise ou panique. Le premier domaine maître à tomber en panne est celui responsable de la détermination du comportement des domaines esclaves.

```
# ldm add-domain failure-policy=reset twizzle
# ldm set-domain failure-policy=panic primary
# ldm add-domain master=twizzle,primary chocktaw
```

- Cet exemple montre comment utiliser la commande `ldm set-domain` pour modifier le domaine `orange` afin d'assigner le domaine `primary` en tant que domaine maître. La seconde commande utilise la commande `ldm set-domain` pour assigner `orange` et `primary` en tant que domaines maîtres pour le domaine `tangerine`. La troisième commande répertorie les informations sur tous ces domaines.

```
# ldm set-domain master=primary orange
# ldm set-domain master=orange,primary tangerine
# ldm list -o domain
NAME          STATE      FLAGS    UTIL
primary       active    -n-cv-  0.2%
```

```
SOFTSTATE
Solaris running
```

```
HOSTID
0x83d8b31c
```

```
CONTROL
failure-policy=ignore
```

```
DEPENDENCY
master=
```

```
-----
NAME          STATE      FLAGS    UTIL
orange        bound      - - - - -
```

```
HOSTID
0x84fb28ef
```

```
CONTROL
failure-policy=stop
```

```
DEPENDENCY
```

```

master=primary

VARIABLES
  test_var=Aloha

-----
NAME          STATE   FLAGS  UTIL
tangerine     bound  -----

HOSTID
  0x84f948e9

CONTROL
  failure-policy=ignore

DEPENDENCY
  master=orange,primary

VARIABLES
  test_var=A hui hou

```

- L'exemple suivant montre la liste d'exemples avec une sortie analysable :

```
# ldm list -o domain -p
```

Cycles de dépendance

Le gestionnaire de domaines logiques ne vous autorise pas à créer des relations de domaine qui créent un cycle de dépendance. Un *cycle de dépendance* est une relation entre deux domaines ou plus qui aboutit à une situation dans laquelle un domaine esclave dépend de lui-même, ou un domaine maître dépend d'un de ses domaines esclaves.

Le gestionnaire de domaines logiques détermine si un cycle de dépendance existe avant d'ajouter une dépendance. Le gestionnaire de domaines logiques démarre sur le domaine esclave et effectue une recherche sur tous les chemins indiqués par la matrice maître jusqu'à ce que le chemin soit atteint. Tous les cycles de dépendance trouvés sur le chemin sont signalés comme des erreurs.

L'exemple suivant montre comment un cycle de dépendance peut être créé. La première commande crée un domaine esclave nommé mohawk qui définit le domaine maître en tant que primary. Par conséquent, mohawk dépend de primary dans la chaîne de dépendance suivante :

FIGURE 12-1 Dépendance d'un domaine unique



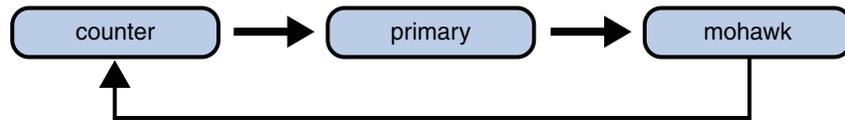
La seconde commande crée un domaine esclave nommé primary qui définit son domaine maître en tant que counter. Par conséquent, mohawk dépend de primary, qui dépend de counter dans la chaîne de dépendance suivante :

FIGURE 12-2 Dépendance de plusieurs domaines



La troisième commande essaie de créer une dépendance entre les domaines counter et mohawk, ce qui produirait le cycle de dépendance suivant :

FIGURE 12-3 Cycle de dépendance de domaines



La commande `ldm set-domain` échouera avec le message d'erreur suivant :

```

# ldm add-domain master=primary mohawk
# ldm set-domain master=counter primary
# ldm set-domain master=mohawk counter
Dependency cycle detected: LDom "counter" indicates "primary" as its master
  
```

Détermination de l'endroit où les erreurs sont survenues lors du mappage du CPU et des adresses de mémoire

Cette section décrit comment vous pouvez corréler les informations transmises par l'architecture de gestion des pannes (FMA) d'Oracle Solaris avec les ressources des domaines logiques identifiées comme étant en panne.

Le FAM signale les erreurs CPU en terme de nombres de CPU physiques et les erreurs de mémoire en termes d'adresses de mémoire physique.

Si vous voulez déterminer dans quel domaine logique une erreur est survenue et le numéro de CPU virtuel correspondant ou l'adresse de mémoire réelle dans le domaine, vous devez effectuer un mappage.

Mappage du CPU

Le numéro de domaine et de CPU virtuel dans le domaine, qui correspond à un numéro de CPU physique donné, peut être déterminé avec les procédures suivantes.

▼ Détermination du numéro de CPU

- 1 Générez une longue liste analysable de tous les domaines.

```
primary# ldm list -l -p
```

- 2 Recherchez l'entrée dans les sections VCPU de la liste ayant une zone pid équivalente au numéro de CPU physique.
 - Si vous trouvez une entrée de ce type, le CPU est dans le domaine sous lequel l'entrée est répertoriée, et le numéro de CPU virtuel dans le domaine est indiqué par la zone vid de l'entrée.
 - Si vous ne trouvez pas d'entrée de ce type, le CPU ne se trouve dans aucun domaine.

Mappage de la mémoire

Le domaine et l'adresse de mémoire réelle dans le domaine, qui correspondent à une adresse de mémoire physique (PA) donnée, peuvent être déterminés comme suit.

▼ Détermination de l'adresse de la mémoire réelle

- 1 Générez une longue liste analysable de tous les domaines.

```
primary# ldm list -l -p
```

- 2 Recherchez la ligne dans les sections MEMORY de la liste où le PA se trouve dans la plage comprenant pa à $(pa + taille - 1)$, c'est-à-dire, $pa \leq PA < (pa + taille - 1)$.

Ici pa et $taille$ concernent les valeurs dans les zones correspondantes de la ligne.

- Si vous trouvez une entrée de ce type, le PA est dans le domaine sous lequel l'entrée est répertoriée, et l'adresse réelle correspondante dans le domaine est indiquée par la zone $ra + (PA - pa)$.
- Si vous ne trouvez pas d'entrée de ce type, le PA ne se trouve dans aucun domaine.

Exemples de mappage de CPU et de mémoire

Supposons que vous ayez une configuration de domaine logique identique à celle représentée dans l'[Exemple 12-1](#), et que vous souhaitiez déterminer le domaine et le CPU virtuel correspondant au numéro de CPU physique 5, et le domaine et l'adresse réelle correspondant à l'adresse physique 0x7e816000.

En recherchant dans les entrées VCPU de la liste l'une des zones pid égale à 5, vous pouvez trouver l'entrée suivante sous le domaine logique ldg1.

```
|vid=1|pid=5|util=29|strand=100
```

Par conséquent, le numéro de CPU physique 5 se trouve dans le domaine ldg1 et dans le domaine, il a le numéro de CPU virtuel 1.

En recherchant dans les entrées MEMORY de la liste, vous pouvez trouver l'entrée suivante sous le domaine ldg2.

```
ra=0x8000000|pa=0x78000000|size=1073741824
```

Où $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$, c'est-à-dire $pa \leq PA \leq (pa + taille - 1)$. Par conséquent, le PA se trouve dans le domaine ldg2 et l'adresse réelle correspondante est $0x8000000 + (0x7e816000 - 0x78000000) = 0xe816000$.

EXEMPLE 12-1 Liste longue analysable des configurations de domaines logiques

```
primary# ldm list -l -p
VERSION 1.0
DOMAIN|name=primary|state=active|flags=normal,control,vio-service|cons=SP|ncpu=4|mem=1073741824|util=0.6|
uptime=64801|softstate=Solaris running
VCPU
|vid=0|pid=0|util=0.9|strand=100
|vid=1|pid=1|util=0.5|strand=100
|vid=2|pid=2|util=0.6|strand=100
|vid=3|pid=3|util=0.6|strand=100
MEMORY
|ra=0x8000000|pa=0x80000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
|dev=pci@7c0|alias=bus_b
...
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|ncpu=2|mem=805306368|util=29|uptime=903|
softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x8000000|pa=0x48000000|size=805306368
...
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|ncpu=3|mem=1073741824|util=35|uptime=775|
softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x8000000|pa=0x78000000|size=1073741824
...
```

Utilisation des identificateurs uniques universellement

À partir de la version Oracle VM Server for SPARC 2.0, un identificateur unique universellement (UUID) est assigné à chaque domaine. L'UUID est assigné lorsqu'un domaine est créé. Pour les domaines propriétaire, l'UUID est assigné lorsque le démon `ldmd` s'initialise.

Remarque – L'UUID est perdu si vous utilisez la commande `ldm migrate -domain -f` pour migrer un domaine vers la machine cible qui exécute une version antérieure du gestionnaire de domaines logiques. Lorsque vous migrez un domaine d'une machine source qui exécute une version antérieure du gestionnaire de domaines logiques, le domaine est associé à un nouvel UUID dans le cadre de la migration. Sinon, l'UUID est migré.

Vous pouvez obtenir l'UUID pour un domaine en exécutant la commande `ldm list -l`, `ldm list -bindings` ou `ldm list -o domain`. Les exemples suivants montrent l'UUID pour le domaine `ldg1` :

```
primary# ldm create ldg1
primary# ldm ls -l ldg1
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
ldg1          inactive  - - - - -
UUID
6c908858-12ef-e520-9eb3-f1cd3dbc3a59

primary# ldm ls -l -p ldg1
VERSION 1.4
DOMAIN|name=ldg1|state=inactive|flags=|cons=|ncpu=|mem=|util=|uptime=
UUID|uuid=6c908858-12ef-e520-9eb3-f1cd3dbc3a59
```

Commande et API d'information sur le domaine virtuel

La commande `virtinfo` vous permet de rassembler des informations sur un domaine virtuel en cours d'exécution. Vous pouvez également utiliser l'API d'informations sur le domaine virtuel pour créer des programmes destinés à rassembler des informations concernant les domaines virtuels.

La liste suivante montre certaines informations que vous pouvez rassembler sur un domaine virtuel à l'aide de la commande ou de l'API :

- Type de domaine (implémentation, contrôle, invité, E/S, service, racine)
- Nom du domaine déterminé par le gestionnaire de domaines logiques
- Identificateur unique universellement (UUID) du domaine
- Nom du nœud de réseau du domaine de contrôle du domaine
- Numéro de série du châssis sur lequel le domaine est en cours d'exécution

Pour plus d'informations sur la commande `virtinfo`, reportez-vous à la page de manuel [virtinfo\(1M\)](#). Pour plus d'informations sur l'API, reportez-vous aux pages de manuel [libv12n\(3LIB\)](#) et [v12n\(3EXT\)](#).

Outil de conversion physique-à-virtuel Oracle VM Server for SPARC

Cette annexe aborde les sujets suivants :

- “Présentation de l’outil P2V Oracle VM Server for SPARC” à la page 215
- “Périphériques du moteur de traitement” à la page 217
- “Installation de l’outil P2V Oracle VM Server for SPARC” à la page 218
- “Utilisation de la commande `ldmp2v`” à la page 221

Présentation de l’outil P2V Oracle VM Server for SPARC

L’outil P2V Oracle VM Server for SPARC convertit automatiquement un système physique existant en système virtuel qui s’exécute dans un domaine logique sur un système multithreading de puce (CMT). Le système source peut être l’un des suivants :

- Un système sun4u SPARC qui exécute au moins le SE Solaris 8
- Un système sun4v qui exécute le SE Oracle Solaris 10, mais ne s’exécute pas dans un domaine logique

La conversion d’un système physique en un système virtuel est effectué selon la procédure suivante :

- **Phase collecte.** S’exécute sur le système source physique. Dans la phase de collecte, une image du système de fichier du système source est créée en fonction des informations de configuration collectées sur le système source.
- **Phase de préparation.** S’exécute sur le domaine de contrôle du système cible. Dans le phase de préparation, un domaine logique est créé sur le système cible en fonction des informations de configuration collectées au cours de la phase de collecte. L’image du système de fichiers est restaurée sur un ou plusieurs disques virtuels. Vous pouvez utiliser l’outil P2V pour créer les disques virtuels sur des fichiers bruts ou des volumes ZFS. Vous pouvez également créer des disques virtuels sur des disques physiques ou des LUN ou sur les volumes du gestionnaire de volumes que vous avez créé. L’image est modifiée afin de pouvoir s’exécuter en tant que domaine logique.

- **Phase de conversion.** S'exécute sur le domaine de contrôle du système cible. Dans la phase de conversion, le domaine logique créé est converti en domaine logique exécutant le SE Oracle Solaris 10 à l'aide du processus de mise à niveau standard Solaris.

Pour plus d'informations sur l'outil P2V, reportez-vous à la page de manuel [ldmp2v\(1M\)](#).

Les sections suivantes décrivent comme la conversion d'un système physique à un système virtuel est effectuée par phases.

Phase de collecte

La phase de collecte s'exécute sur le système à convertir. Pour créer un image du système de fichiers cohérente, vérifiez que le système est aussi silencieux que possible et que toutes les applications sont arrêtées. La commande `ldmp2v` crée une sauvegarde de tous les systèmes de fichiers UFS montés. Vérifiez donc que les systèmes de fichiers à déplacer vers un domaine logique sont montés. Vous pouvez exclure les systèmes de fichiers montés que vous ne souhaitez pas déplacer, notamment les systèmes de fichiers sur un stockage SAN ou les systèmes de fichiers qui seront déplacés par d'autres moyens. Utilisez l'option `-x` pour exclure de tels systèmes de fichiers. Les systèmes de fichiers exclus par l'option `-x` ne sont pas recréés sur le domaine invité. Vous pouvez utiliser l'option `-O` pour exclure des fichiers et des répertoires.

Aucune modification n'est nécessaire dans le système source. La seule contrainte est que le script `ldmp2v` qui a été installé sur le domaine de contrôle. Vérifiez que l'utilitaire `flarc create` est présent sur le système source.

Phase de préparation

La phase de préparation utilise les données collectées au cours de la phase de collecte pour créer un domaine logique qui est comparable au système source.

Vous pouvez utiliser la commande `ldmp2v` de l'une des manières suivantes :

- **Mode automatique.** Ce mode crée automatiquement des disques virtuels et restaure les données du système de fichiers.

- Crée le domaine logique et les disques virtuels requis de la même taille sur le système source.
- Partitionne les disques et restaure les systèmes de fichiers.

Si la taille associée des systèmes de fichiers `/`, `/usr` et `/var` est inférieure à 10 Go, les tailles de ces systèmes de fichiers sont automatiquement ajustées pour s'adapter aux contraintes d'espace disque plus importantes du SE Oracle Solaris 10. Le redimensionnement automatique peut être désactivé à l'aide de l'option `-x no-auto-adjust -fs` ou à l'aide de l'option `-m` pour redimensionner manuellement un système de fichiers.

- Modifie l'image du SE du domaine logique pour remplacer toutes les références au matériel physique avec les versions appropriées à un domaine logique. Cela vous permet de mettre à niveau le système au SE Oracle Solaris 10 à l'aide du processus de mise à niveau normal Solaris. Les modifications comprennent la mise à niveau du fichier `/etc/vfstab` pour prendre en compte les nouveaux noms de disque. Tous les disques de démarrage encapsulés Solaris Volume Manager ou Veritas Volume Manager (VxVM) sont automatiquement désencapsulés au cours de ce processus. Lorsqu'un disque est désencapsulé, il est converti en segments de disque bruts. Si VxVM est installé sur le système source, le processus P2V désactive VxVM sur le domaine invité créé.
- **Mode non automatique.** Vous devez créer les disques virtuels et restaurer les données du système de fichiers manuellement. Ce mode vous permet de modifier la taille et le nombre de disques, le partitionnement et la disposition du système de fichiers. La phase de préparation de ce mode exécute uniquement la création du domaine logique et les étapes de modification de l'image du SE sur le système de fichiers.
- **Mode de nettoyage.** Supprime un domaine logique et tous les périphériques du moteur de traitement sous-jacents créés par `ldmp2v`.

Phase de conversion

Dans la phase de conversion, le domaine logique utilise le processus de mise à niveau Solaris pour mettre à niveau le SE Oracle Solaris 10. L'opération de mise à niveau supprime tous les packages existants et installe les packages `sun4v` Oracle Solaris 10, qui exécutent automatiquement une conversion `sun4u` à `sun4v`. La phase de conversion peut utiliser une image iso du DVD Oracle Solaris ou une image d'installation réseau. Vous pouvez également utiliser une installation JumpStart personnalisée pour effectuer une opération de mise à niveau totalement automatisée.

Périphériques du moteur de traitement

Vous pouvez créer des disques virtuels pour un domaine invité sur un certain nombre de types de moteur de traitement : fichiers (`file`), volumes ZFS (`zvol`), disques physiques ou LUN (`disk`) ou volumes de gestionnaire de volumes (`disk`). La commande `ldmp2v` crée automatiquement des fichiers ou des volumes ZFS de la taille appropriée si vous indiquez `file` ou `zvol` comme type de moteur de traitement avec l'une des méthodes suivantes :

- À l'aide de l'option `-b`
- En indiquant la valeur du paramètre `BACKEND_TYPE` dans le fichier `/etc/ldmp2v.conf`

Le type de moteur de traitement `disk` vous permet d'utiliser un disque physique, un LUN ou un volume de gestionnaire de volumes (Solaris Volume Manager et Veritas Volume Manager (VxVM)) en tant que périphérique de moteur de traitement pour les disques virtuels. Vous devez créer le disque ou le volume avec une taille appropriée avant de commencer la phase de

préparation. Pour un disque physique ou un LUN, définissez le périphérique du moteur de traitement en tant que segment 2 du bloc ou périphérique de caractère du disque, notamment `/dev/dsk/c0t3d0s2`. Pour un volume de gestionnaire de volumes, définissez le bloc ou le périphérique de caractère pour le volume, notamment `/dev/md/dsk/d100` pour Solaris Volume Manager ou `/dev/vx/dsk/ldomdg/vol1` pour VxVM.

À moins de définir les noms de volume et de disque virtuel avec l'option `-B backend:volume:vdisk`, les volumes et les disques virtuels que vous créez pour l'invité sont les noms donnés par défaut.

- *backend* définit le nom du moteur de traitement à utiliser. Vous devez définir *backend* pour le type de moteur de traitement `disk`. *backend* est facultatif pour les types de moteur de traitement `file` et `zvol`, et peut être utilisé pour définir un nom non par défaut pour le fichier ou le volume ZFS que `ldmp2v` crée. Le nom par défaut est `$_BACKEND_PREFIX/guest-name/diskN`.
- *volume* est facultatif pour tous les types de moteur de traitement et définit le nom du volume du serveur de disques virtuels pour créer le domaine invité. Si cette option n'est pas indiquée, *volume* est `guest-name-volN`.
- *vdisk* est facultatif pour tous les types de moteur de traitement et définit le nom du volume dans le domaine invité. Si cette option n'est pas indiquée, *vdisk* est `diskN`.

Remarque – Au cours du processus de conversion, le disque virtuel est temporairement nommé `guest-name-diskN` pour garantir que le nom dans le domaine de contrôle est unique.

Pour indiquer une valeur vide pour *backend*, *volume* ou *vdisk*, incluez uniquement le séparateur deux-points. Par exemple, `-B :vdisk001` définit le nom du disque virtuel sur `vdisk001` et utilise les noms par défaut pour le moteur de traitement et le volume. Si vous ne définissez pas *vdisk*, vous pouvez omettre le séparateur deux-points de fin. Par exemple, `-B /ldoms/ldom1/vol001:vol001` définit le nom du fichier du moteur de traitement comme `/ldoms/ldom1/vol001` et le nom du volume comme `vol001`. Le nom du disque virtuel par défaut est `disk0`.

Installation de l'outil P2V Oracle VM Server for SPARC

Le package de l'outil P2V Oracle VM Server for SPARC doit être installé et configuré *uniquement* sur le domaine de contrôle du système cible. Il est inutile d'installer le package sur le système source. Au lieu de cela, il vous suffit de copier le script `/usr/sbin/ldmp2v` du système cible sur le système source.

Conditions requises

Avant de pouvoir exécuter l'outil P2V Oracle VM Server for SPARC, vérifiez que les conditions suivantes sont remplies :

- Les patches de l'utilitaire flash suivant sont installés sur le système source :
 - **Pour le SE Solaris 8** : au moins le patch ID 109318-34
 - **Pour le SE Solaris 9** : au moins le patch ID 113343-06
- Le système cible exécute au moins &LDoms 1.1 sur :
 - SE Oracle Solaris 10 10/08
 - SE Oracle Solaris 10 5/08 avec les patches appropriés &LDoms 1.1
- Les domaines invités exécutent au moins le SE Oracle Solaris 10 5/08
- Le système source exécute au moins le SE Solaris 8

En plus de ces conditions requises, configurez un système de fichiers NFS à partager par les systèmes source et cible. Le système de fichiers doit être accessible en écriture par root. Cependant, si un système de fichiers partagé n'est pas disponible, utilisez un système de fichiers local de taille suffisante pour contenir un vidage du système de fichiers du système source sur les systèmes source et cible.

Restrictions

La version 2.0 de l'outil P2V Oracle VM Server for SPARC présente les restrictions suivantes :

- Seuls les systèmes de fichiers UFS sont pris en charge.
- Seuls les disques bruts (/dev/dsk/c0t0d0s0), les métapériphériques Solaris Volume Manager (/dev/md/dsk/dNNN) et les disques de démarrage encapsulés VxVM sont pris en charge sur le système source.
- Au cours du processus P2V, chaque domaine invité ne peut avoir qu'un seul commutateur virtuel et un seul serveur de disque virtuel. Vous pouvez ajouter plusieurs commutateurs virtuels et serveurs de disque virtuel au domaine après la conversion P2V.
- La prise en charge VxVM est limitée aux volumes suivants sur le disque de démarrage encapsulé : rootvol, swapvol, usr, var, opt et home. Les segments d'origine de ces volumes doivent toujours être présents sur le disque de démarrage. L'outil P2V prend en charge Veritas Volume Manager 5.x sur le SE Solaris 10. Cependant, vous pouvez également utiliser l'outil P2V pour convertir les systèmes d'exploitation Solaris 8 et Solaris 9 utilisant VxVM.
- Vous ne pouvez pas convertir les systèmes Solaris 10 configurés avec des zones.

▼ Installation de l'outil P2V Oracle VM Server for SPARC

- 1 Accédez à la page de téléchargement Oracle VM Server for SPARC à l'adresse <http://www.sun.com/servers/coolthreads/ldoms/get.jsp>.

- 2 Téléchargez le package logiciel P2V, SUNWldmp2v.

À partir de la version domaines logiques 1.2, le package SUNWldmp2v est inclus dans le fichier zip Oracle VM Server for SPARC.

- 3 Devenez un superutilisateur ou assumez un rôle équivalent.

Les rôles contiennent des autorisations et des commandes nécessitant des droits d'accès. Pour plus d'informations sur les rôles, reportez-vous à la section “Configuring RBAC (Task Map)” du *System Administration Guide: Security Services*.

- 4 Utilisez la commande `pkgadd` pour installer le package SUNWldmp2v.

```
# pkgadd -d . SUNWldmp2v
```

- 5 Créez le fichier `/etc/ldmp2v.conf` et configurez les propriétés par défaut suivantes :

- VDS – Nom du service de disque virtuel, par exemple `VDS="primary-vds0"`
- VSW – Nom du commutateur virtuel, par exemple `VSW="primary-vsw0"`
- VCC – Nom du concentrateur de console virtuelle, par exemple `VCC="primary-vcc0"`
- BACKEND_TYPE – Type de moteur de traitement de zvol, file ou disk
- BACKEND_SPARSE – Créer des périphériques de moteur de traitement en tant que volumes ou fichiers d'analyse syntaxique `BACKEND_SPARSE="yes"` ou en tant que fichiers ou volumes sans analyse syntaxique `BACKEND_SPARSE="no"`
- BACKEND_PREFIX – Emplacement pour créer les périphériques de moteur de traitement du disque virtuel

Si `BACKEND_TYPE="zvol"`, définissez la valeur `BACKEND_PREFIX` en tant que nom d'ensemble de données ZFS. Si `BACKEND_TYPE="files"`, la valeur `BACKEND_PREFIX` est interprétée comme un nom de chemin d'un répertoire relatif à/.

Par exemple, `BACKEND_PREFIX="tank/ldoms"` créerait des ZVOL dans l'ensemble de données `tank/ldoms/domain-name` et des fichiers dans le sous-répertoire `/tank/ldoms/domain-name`.

La propriété `BACKEND_PREFIX` n'est pas applicable à l'arrière-plan `disk`.

- BOOT_TIMEOUT – Délai d'attente du démarrage du SE Oracle Solaris en secondes

Pour plus d'informations, consultez le fichier de configuration `ldmp2v.conf.sample` qui fait partie du bundle téléchargeable.

Utilisation de la commande `ldmp2v`

Cette section inclut des exemples pour ces trois phrases.

EXEMPLE A-1 Exemples de phase de collecte

Les exemples suivants montrent comment vous pouvez utiliser la commande `ldmp2v collect`.

- **Partage d'un système de fichiers monté NFS.** L'exemple suivant montre la méthode la plus simple pour effectuer l'étape de collecte si les systèmes source et cible partagent un système de fichiers monté NFS.

En tant que superutilisateur, vérifiez que tous les systèmes de fichiers UFS requis sont montés.

```
volumia# df -k
Filesystem            kbytes    used    avail capacity  Mounted on
/dev/dsk/c1t1d0s0    16516485  463289 15888032    3%      /
/proc                  0          0        0      0%     /proc
fd                     0          0        0      0%     /dev/fd
mnttab                 0          0        0      0%     /etc/mnttab
/dev/dsk/c1t1d0s3    8258597   4304   8171708    1%     /var
swap                  4487448    16   4487432    1%     /var/run
swap                  4487448    16   4487432    1%     /tmp
/dev/dsk/c1t0d0s0    1016122    9   955146     1%     /u01
vandikhout:/u1/home/dana
6230996752 1051158977 5179837775    17%    /home/dana
```

L'exemple suivant montre comment exécuter l'outil de collecte lorsque les systèmes source et cible partagent un système de fichiers monté NFS :

```
volumia# ldmp2v collect -d home/dana/volumia
Collecting system configuration ...
Archiving file systems ...
Determining which filesystems will be included in the archive...
Creating the archive...
895080 blocks
Archive creation complete.
```

- **Non partage d'un système de fichiers monté NFS.** Lorsque les systèmes source et cible ne partagent pas un système de fichiers monté NFS, l'image du système de fichiers peut être écrite sur le stockage local, puis copiée ultérieurement sur le domaine de contrôle. L'utilitaire Flash exclut automatiquement l'archive qu'il crée.

```
volumia# ldmp2v collect -d /var/tmp/volumia
Collecting system configuration ...
Archiving file systems ...
Determining which filesystems will be included in the archive...
Creating the archive...
895080 blocks
Archive creation complete.
```

Copiez l'archive Flash et le fichier manifest du répertoire `/var/tmp/volumia` sur le système cible.

EXEMPLE A-1 Exemples de phase de collecte (Suite)

- **Omission de l'étape de sauvegarde du système de fichiers.** Si les sauvegardes du système sont déjà disponibles à l'aide d'un outil de sauvegarde tiers tel que NetBackup, vous pouvez passer l'étape de sauvegarde du système de fichiers en utilisant la méthode d'archivage `none`. Lorsque vous utilisez cette option, seul le manifeste de la configuration du système est créé.

```
volumia# ldmp2v collect -d /home/dana/p2v/volumia -a none
Collecting system configuration ...
The following file system(s) must be archived manually: / /u01 /var
```

Notez que, si la répertoire indiqué par `-d` n'est pas partagé par les systèmes source et cible, vous devez copier le contenu de ce répertoire sur le domaine de contrôle. Le contenu du répertoire doit être copié sur le domaine de contrôle avant la phase de préparation.

EXEMPLE A-2 Exemples de phase de préparation

Les exemples suivants montrent comment vous pouvez utiliser la commande `ldmp2v prepare`.

- L'exemple suivant crée un domaine logique appelé `volumia` à l'aide des valeurs par défaut configurées dans `/etc/ldmp2v.conf` tout en conservant les adresses MAC du système physique :

```
# ldmp2v prepare -d /home/dana/p2v/volumia -o keep-mac volumia
Creating vdisks ...
Creating file systems ...
Populating file systems ...
Modifying guest domain OS image ...
Removing SVM configuration ...
Unmounting guest file systems ...
Creating domain volumia ...
Attaching vdisks to domain volumia ...
```

- La commande suivante montre des informations sur le domaine logique `volumia` :

```
# ldm list -l volumia
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  UPTIME
volumia      inactive  -----    2      4G

NETWORK
NAME  SERVICE          DEVICE    MAC              MODE  PVID  VID
vnet0 primary-vsw0          00:03:ba:1d:7a:5a  1

DISK
NAME  DEVICE  TOUT  MPGROUP          VOLUME          SERVER
disk0          volumia-vol0@primary-vds0
disk1          volumia-voll@primary-vds0
```

- L'exemple suivant montre que vous pouvez totalement supprimer un domaine et ses périphériques de moteur de traitement à l'aide de l'option `-C` :

```
# ldmp2v prepare -C volumia
Cleaning up domain volumia ...
Removing vdisk disk0 ...
Removing vdisk disk1 ...
Removing domain volumia ...
```

EXEMPLE A-2 Exemples de phase de préparation (Suite)

```
Removing volume volumnia-vol0@primary-vds0 ...
Removing ZFS volume tank/ldoms/volumnia/disk0 ...
Removing volume volumnia-vol1@primary-vds0 ...
Removing ZFS volume tank/ldoms/volumnia/disk1 ...
```

- L'exemple suivant montre que vous pouvez redimensionner un ou plusieurs systèmes de fichiers au cours de P2V en indiquant le point de montage et la nouvelle taille avec l'option `-m`.

```
# ldmp2v prepare -d /home/dana/p2v/normaal -m /:8g normaal
Resizing file systems ...
Creating vdisks ...
Creating file systems ...
Populating file systems ...
Modifying guest domain OS image ...
Removing SVM configuration ...
Modifying file systems on SVM devices ...
Unmounting guest file systems ...
Creating domain normaal ...
Attaching vdisks to domain normaal ...
```

EXEMPLE A-3 Exemples de phase de conversion

Les exemples suivants montrent comment vous pouvez utiliser la commande `ldmp2v convert`.

- **Utilisation d'un serveur d'installation réseau.** La commande `ldmp2v convert` démarre le domaine sur le réseau à l'aide de l'interface de réseau virtuelle indiquée. Vous devez exécuter les scripts `setup_install_server` et `add_install_client` sur le serveur d'installation.

Vous pouvez utiliser la fonction d'installation JumpStart personnalisée pour effectuer une conversion totalement automatisée. Cette fonction nécessite que vous créiez et configuriez les fichiers `sysidcfg` et de profil appropriés pour le client du serveur JumpStart. Le profil doit être composé des lignes suivantes :

```
install_type    upgrade
root_device    c0d0s0
```

Le fichier `sysidcfg` est uniquement utilisé pour l'opération de mise à niveau. Par conséquent, une configuration telle que la suivante devrait être suffisante :

```
name_service=NONE
root_password=uQkoX\MLCsZhI
system_locale=C
timeserver=localhost
timezone=Europe/Amsterdam
terminal=vt100
security_policy=NONE
nfs4_domain=dynamic
network_interface=PRIMARY {netmask=255.255.255.192
                           _default_route=none protocol_ipv6=no}
```

Pour en savoir plus sur l'utilisation d'une installation JumpStart personnalisée, voir le [Guide d'installation d'Oracle Solaris 10 9/10 : installation JumpStart personnalisée et installation avancée](#).

EXEMPLE A-3 Exemples de phase de conversion (Suite)

```
# ldmp2v convert -j -n vnet0 -d /p2v/volumia volumia
LDom volumia started
Waiting for Solaris to come up ...
Using Custom JumpStart
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "volumia" in group "volumia" ....
Press ~? for control options ..
SunOS Release 5.10 Version Generic 137137-09 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.
Configuring devices.
Using RPC Bootparams for network configuration information.
Attempting to configure interface vnet0...
Configured interface vnet0
Reading ZFS config: done.
Setting up Java. Please wait...
Serial console, reverting to text install
Beginning system identification...
Searching for configuration file(s)...
Using sysid configuration file
  129.159.206.54:/opt/SUNWjet/Clients/volumia/sysidcfg
Search complete.
Discovering additional network configuration...
Completing system identification...
Starting remote procedure call (RPC) services: done.
System identification complete.
Starting Solaris installation program...
Searching for JumpStart directory...
Using rules.ok from 129.159.206.54:/opt/SUNWjet.
Checking rules.ok file...
Using begin script: Clients/volumia/begin
Using profile: Clients/volumia/profile
Using finish script: Clients/volumia/finish
Executing JumpStart preinstall phase...
Executing begin script "Clients/volumia/begin"...
Begin script Clients/volumia/begin execution completed.
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.
WARNING: Backup media not specified. A backup media (backup_media)
keyword must be specified if an upgrade with disk space reallocation
is required

Processing profile

Loading local environment and services

Generating upgrade actions
Checking file system space: 100% completed
Space check complete.
```

EXEMPLE A-3 Exemples de phase de conversion (Suite)

```
Building upgrade script
Preparing system for Solaris upgrade
Upgrading Solaris: 10% completed
[...]
```

- **Utilisation d'une image ISO.** La commande ldmp2v convert associe l'image iso du DVD Oracle Solaris au domaine logique et démarre à partir de celui-ci. Pour effectuer la mise à niveau, répondez à toutes les invites sysid et sélectionnez Upgrade (Mise à niveau).

Remarque – Les réponses aux questions sysid questions sont *uniquement* utilisées pour la durée du processus de mise à niveau. Ces données ne sont pas appliquées à l'image du SE existant sur le disque. La méthode la plus simple et la plus rapide pour exécuter la conversion consiste à sélectionner Non-networked (Non mis en réseau). Le mot de passe root que vous indiquez ne doit pas nécessairement correspondre au mot de passe root du système source. L'identité d'origine du système est préservée par la mise à niveau et prend effet après le redémarrage post mise à niveau. Le temps requis pour effectuer la mise à niveau dépend du cluster Oracle Solaris installé sur le système d'origine.

```
# ldmp2v convert -i /tank/iso/s10s_u5.iso -d /home/dana/p2v/volumia volumia
Testing original system status ...
LDom volumia started
Waiting for Solaris to come up ...
```

```
        Select 'Upgrade' (F2) when prompted for the installation type.
        Disconnect from the console after the Upgrade has finished.
```

```
Trying 0.0.0.0...
Connected to 0.
Escape character is '^'.
```

```
Connecting to console "volumia" in group "volumia" ....
Press ~? for control options ..
Configuring devices.
Using RPC Bootparams for network configuration information.
Attempting to configure interface vnet0...
Extracting windowing system. Please wait...
Beginning system identification...
Searching for configuration file(s)...
Search complete.
Discovering additional network configuration...
Configured interface vnet0
Setting up Java. Please wait...
```

```
Select a Language

0. English
1. French
2. German
3. Italian
```

EXEMPLE A-3 Exemples de phase de conversion (Suite)

4. Japanese
5. Korean
6. Simplified Chinese
7. Spanish
8. Swedish
9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help:
[...]

- Solaris Interactive Installation -----

This system is upgradable, so there are two ways to install the Solaris software.

The Upgrade option updates the Solaris software to the new release, saving as many modifications to the previous version of Solaris software as possible. Back up the system before using the Upgrade option.

The Initial option overwrites the system disks with the new version of Solaris software. This option allows you to preserve any existing file systems. Back up any modifications made to the previous version of Solaris software before starting the Initial option.

After you select an option and complete the tasks that follow, a summary of your actions will be displayed.

F2_Upgrade F3_Go Back F4_Initial F5_Exit F6_Help

Assistant de configuration Oracle VM Server for SPARC

L'assistant de configuration Oracle VM Server for SPARC vous guide tout au long de la configuration d'un domaine logique en paramétrant les propriétés de base. Il s'exécute sur des systèmes basés sur le multithreading de puce (CMT).

Après avoir rassemblé les données de configuration, l'assistant de configuration crée une configuration adaptée à l'initialisation en tant que domaine logique. Vous pouvez également utiliser les valeurs par défaut sélectionnées par l'assistant de configuration afin de créer une configuration système utilisable.

L'assistant de configuration est disponible sous forme d'interface graphique (GUI) et d'outil basé sur un terminal, `ldmconfig`.

Pour plus d'informations à propos de l'outil basé sur un terminal, reportez-vous à la section "[Utilisation de l'assistant de configuration \(`ldmconfig`\)](#)" à la page 228 et à la page de manuel `ldmconfig(1M)`.

Pour plus d'informations sur le démarrage de l'outil d'interface graphique, reportez-vous à la section "[Utilisation de l'assistant de configuration \(interface graphique\)](#)" à la page 227.

Utilisation de l'assistant de configuration (interface graphique)

L'interface graphique de l'assistant de configuration est fournie dans le bundle zip d'Oracle VM Server for SPARC.

Vérifiez que le système cible exécute au moins le logiciel 1.2 des domaines logiques et que votre système exécute au moins la version 1.6 de l'environnement d'exécution SE Java.

Pour exécuter l'interface graphique de l'assistant de configuration, tapez la commande suivante :

```
$ java -jar "Configurator.jar"
```

L'outil d'interface graphique comprend une documentation à l'écran pour vous aider à créer la configuration pour votre système.

Utilisation de l'assistant de configuration (`ldmconfig`)

L'assistant de configuration basé sur un terminal, `ldmconfig`, utilise une série d'opérations qui correspondent aux écrans de l'interface graphique. Le résultat final est la création d'une configuration que vous pouvez déployer sur un domaine logique.

Les sections suivantes décrivent la procédure d'installation de la commande `ldmconfig` ainsi que certaines fonctions de l'outil Assistant de configuration.

Installation de l'assistant de configuration

L'assistant de configuration est livré avec le package `SUNWldm`.

Après avoir installé le package `SUNWldm`, vous pouvez rechercher la commande `ldmconfig` dans le répertoire `/usr/bin`. La commande est également installée dans le répertoire `/opt/SUNWldm/bin` à des fins de récupération.

Conditions requises

Avant d'installer et d'exécuter l'assistant de configuration, vérifiez que les conditions suivantes sont remplies :

- Le système cible doit exécuter au moins le logiciel domaines logiques 1.2.
- Votre fenêtre de terminal doit présenter une largeur d'au moins 80 caractères sur une hauteur de 24 lignes.

Limitations et problèmes recensés

L'assistant de configuration présente les limitations suivantes :

- Le redimensionnement du terminal pendant l'utilisation de `ldmconfig` peut provoquer une sortie confuse
- Prise en charge des fichiers de disque UFS en tant que disques virtuels uniquement
- Fonctionne uniquement avec des systèmes où aucune configuration de domaines logiques existante n'est présente
- Les ports du concentrateur de la console virtuelle vont de 5000 à 5100
- Les noms par défaut utilisés pour les domaines, les services et les périphériques invités ne peuvent pas être modifiés

ldmconfig : Fonctions

L'assistant de configuration basé sur un terminal, `ldmconfig`, utilise une série d'opérations qui correspondent aux écrans de l'interface graphique. Vous pouvez naviguer en arrière (précédent) et en avant (suivant) dans ces écrans jusqu'à ce que vous atteigniez l'étape finale. L'étape finale produit la configuration. Vous pouvez quitter à tout moment l'assistant de configuration ou réinitialiser la configuration sur les valeurs par défaut. À partir de l'écran final, vous pouvez déployer la configuration sur un domaine logique.

L'assistant de configuration inspecte tout d'abord le système pour déterminer les valeurs de propriété par défaut les plus adaptées en fonction des pratiques recommandées, puis affiche les propriétés requises pour contrôler le déploiement. Notez qu'il ne s'agit pas d'une liste exhaustive. Vous pouvez définir d'autres propriétés pour personnaliser encore la configuration.

Pour plus d'informations sur l'utilisation de l'outil `ldmconfig`, reportez-vous à la page de manuel `ldmconfig(1M)`.

Vous pouvez ajuster les propriétés suivantes :

- **Nombre de domaines invités.** Indiquez le nombre de domaines invités pour l'application à créer. Le minimum est un domaine invité. La valeur maximale est déterminée par la disponibilité des ressources du VCPU. Par exemple, vous pouvez créer jusqu'à 60 domaines invités avec un seul thread, chacun sur un système CMT 64 threads et quatre threads étant réservés pour le domaine de contrôle. Si les pratiques recommandées sont sélectionnées, le nombre minimum de ressources VCPU par domaine invité est un serveur de base unique. Par conséquent, sur un système 8 serveurs de base, 8 threads par serveur de base pour lequel les pratiques recommandées sont sélectionnées, vous pouvez créer jusqu'à sept domaines invités avec chacun un serveur de base. Par conséquent, un seul serveur de base est assigné au domaine de contrôle.

L'assistant de configuration présente le nombre maximum de domaines pouvant être configurés pour ce système.

L'assistant de configuration effectue les tâches suivantes pour créer les domaines :

- **Pour tous les domaines.**
 - Crée un service de terminal virtuel sur les ports de 5000 à 5100
 - Crée un service de disque virtuel
 - Crée un commutateur de réseau virtuel sur chaque adaptateur réseau indiqué
 - Active le démon du serveur de terminal virtuel
- **Pour chaque domaine.**
 - Crée le domaine logique
 - Configure les VCPU assignés au domaine
 - Configure la mémoire assignée au domaine
 - Crée un fichier de disque UFS à utiliser en tant que disque virtuel

- Crée un périphérique de serveur de disque virtuel vdsdev pour le fichier de disque
 - Assigne le fichier de disque comme disque virtuel vdisk0 pour le domaine
 - Ajoute un adaptateur réseau virtuel au commutateur virtuel sur l'adaptateur réseau indiqué
 - Définit la propriété OBP auto-boot?=true
 - Définit la propriété OBP boot-device=vdisk0
 - Associe le domaine
 - Démarre le domaine
- **Réseau par défaut.** Spécifie l'adaptateur réseau que les nouveaux domaines utiliseront pour la mise en réseau virtuelle. L'adaptateur doit être présent dans le système. L'assistant de configuration met en évidence les adaptateurs qui sont utilisés par le système en tant qu'adaptateurs par défaut ainsi que ceux ayant un état de liaison active (adaptateurs de câble).
 - **Taille du disque virtuel.** Créez des disques virtuels pour chacun des nouveaux domaines. Ces disques virtuels sont créés en fonction des fichiers de disque situés dans les systèmes de fichiers locaux. Cette propriété contrôle la taille de chaque disque virtuel en Go. La taille minimale, 8 Go, est basée sur la taille approximative requise pour contenir un SE Oracle Solaris 10, et la taille maximale est 100 Go.

Si l'assistant de configuration ne peut pas trouver les systèmes de fichiers contenant un espace adapté pour contenir les fichiers du disque pour tous les domaines, un écran d'erreur s'affiche. Dans ce cas, vous devrez suivre la procédure suivante avant de réexécuter l'application :

- Réduisez la taille des disques virtuels
 - Réduisez le nombre de domaines
 - Ajoutez des systèmes de fichiers avec une capacité plus importante
- **Répertoire du disque virtuel.** Spécifiez un système de fichiers disposant d'une capacité suffisante sur lequel stocker les fichiers à créer en tant que disques virtuels pour les nouveaux domaines. Le répertoire est basé sur le nombre de domaines sélectionnés et la taille des disques virtuels. La valeur doit être recalculée et les répertoires de destination sélectionnés à chaque fois que ces valeurs de propriété sont modifiées. L'assistant de configuration vous présente une liste des systèmes de fichiers ayant suffisamment d'espace. Après avoir indiqué le nom du système de fichiers, l'assistant de configuration crée un répertoire dans ce système de fichiers appelé /ldoms/disks dans lequel créer les images de disque.
 - **Pratique recommandée.** Indiquez s'il faut utiliser les pratiques recommandées pour les valeurs de propriété.
 - Lorsque la valeur est yes, l'assistant de configuration utilise la pratique recommandée pour plusieurs valeurs de propriété de configuration. Il force le minimum d'un serveur de base par domaine, y compris pour les domaines du système. En conséquence, cette opération limite le nombre maximum de domaines invités au nombre total de serveurs de base présents dans le système moins un serveur de base pour les domaines du

système. Par exemple, dans le cas d'un T5140 SPARC Enterprise deux sockets avec huit serveurs de base chacun, le nombre maximum de domaines invités est 15 plus le domaine du système.

- Si la valeur est `no`, l'assistant de configuration autorise la création de domaines ayant un thread au minimum, mais conserve au moins quatre threads pour le domaine du système.

Ensuite, l'assistant de configuration récapitule la configuration de déploiement à créer, qui comprend les informations suivantes :

- Nombre de domaines
- CPU assigné à chaque domaine invité
- Mémoire assignée à chaque domaine invité
- Taille et emplacement des disques virtuels
- Adaptateur réseau à utiliser pour les services de réseau virtuel pour les domaines invités
- Quantité de CPU et de mémoire à utiliser par le système pour les services
- Si un DVD du SE Oracle Solaris valide a été identifié, il sera utilisé pour créer un périphérique CD-ROM virtuel partagé pour permettre aux domaines invités d'installer le SE Oracle Solaris.

Enfin, l'assistant de configuration configure le système pour créer le déploiement des domaines logiques spécifiés. Il décrit également les actions à effectuer et montre les commandes à exécuter pour configurer le système. Ces informations peuvent vous aider à apprendre comment utiliser les commandes `ldm` nécessaires pour configurer le système.



Attention – N'interagissez *pas* avec cette étape de configuration et n'interrompez *pas* ce processus, car le système risquerait d'être partiellement configuré.

Une fois les commandes exécutées avec succès, redémarrez le système pour appliquer les modifications.

Recherche du gestionnaire de domaines logiques

Le gestionnaire de domaines logiques peut être recherché sur un sous-réseau à l'aide des messages de multidiffusion. Le démon `ldmd` peut écouter sur un réseau pour recherche un paquet de multidiffusion spécifique. Si ce message de multidiffusion est d'un certain type, `ldmd` répond au programme appelant. Cela permet à `ldmd` d'être recherchés sur les systèmes exécutant Oracle VM Server for SPARC.

Cette annexe fournit des informations sur la recherche du gestionnaire de domaines logiques s'exécutant sur des systèmes dans un sous-réseau.

Recherche des systèmes exécutant le gestionnaire de domaines logiques

Communication en multidiffusion

Ce mécanisme de découverte utilise le même réseau de multidiffusion que celui utilisée par le démon `ldmd` pour détecter les collisions lors de l'adressage automatique des adresses MAC. Pour configurer le socket de multidiffusion, vous devez fournir les informations suivantes :

```
#define MAC_MULTI_PORT 64535
#define MAC_MULTI_GROUP "239.129.9.27"
```

Par défaut, *seuls* les paquets de multidiffusion peuvent être envoyés sur le sous-réseau auquel la machine est connectée. Vous pouvez modifier ce comportement en définissant la propriété `SMF ldmd/hops` pour le démon `ldmd`.

Format du message

Les messages de recherche doivent être clairement identifiés afin de ne pas être confondus avec d'autres messages. Le format de message de multidiffusion suivant garantit que les messages de recherche peuvent être distingués par le processus d'écoute de recherche :

```
#include <netdb.h> /* Used for MAXHOSTNAMELEN definition */
#define MAC_MULTI_MAGIC_NO 92792004
#define MAC_MULTI_VERSION 1

enum {
    SEND_MSG = 0,
    RESPONSE_MSG,
    LDMD_DISC_SEND,
    LDMD_DISC_RESP,
};

typedef struct {
    uint32_t version_no;
    uint32_t magic_no;
    uint32_t msg_type;
    uint32_t resv;
    union {
        mac_lookup_t Mac_lookup;
        ldmd_discovery_t Ldmd_discovery;
    } payload;
#define lookup payload.Mac_lookup
#define discovery payload.Ldmd_discovery
} multicast_msg_t;

#define LDMD_VERSION_LEN 32

typedef struct {
    uint64_t mac_addr;
    char source_ip[INET_ADDRSTRLEN];
} mac_lookup_t;

typedef struct {
    char ldmd_version[LDMD_VERSION_LEN];
    char hostname[MAXHOSTNAMELEN];
    struct in_addr ip_address;
    int port_no;
} ldmd_discovery_t;
```

▼ Recherche du gestionnaire de domaines logiques s'exécutant sur votre sous-réseau

1 Ouvrez un socket de multidiffusion.

Vérifiez que vous utilisez le port et les informations de groupe indiquées dans [“Communication en multidiffusion”](#) à la page 233.

2 Envoyez un message `multicast_msg_t` sur le socket.

Le message doit inclure les éléments suivants :

- Une valeur valide pour `version_no`, qui est 1 comme défini par `MAC_MULTI_VERSION`
- Une valeur valide pour `magic_no`, qui est 92792004 comme défini par `MAC_MULTI_MAGIC_NO`
- `msg_type` de `LDMD_DISC_SEND`

3 Écoutez sur le socket de multidiffusion les réponse du gestionnaire de domaines logiques.

Les réponses doivent être un message `multicast_msg_t` avec les éléments suivants :

- Valeur valide pour `version_no`
- Valeur valide pour `magic_no`
- `msg_type` défini sur `LDMD_DISC_RESP`
- Charge utile composée d'une structure `ldmd_discovery_t`, qui contient les informations suivantes :
 - `ldmd_version` - Version du gestionnaire de domaines logiques s'exécutant sur le système
 - `hostname` - Nom d'hôte du système
 - `ip_address` - Adresse IP du système
 - `port_no` - Numéro de port utilisé par le gestionnaire de domaines logiques pour les communications, qui doit être le port XMPP 6482

Lors de l'écoute d'une réponse du gestionnaire de domaines logiques, vérifiez que tous les paquets de détection-collision MAC d'allocation automatique sont ignorés.

Utilisation de l'interface XML avec le gestionnaire de domaines logiques

Ce chapitre explique le mécanisme de communication Extensible Markup Language (XML) par lequel les programmes des utilisateurs externes peuvent communiquer avec le logiciel Oracle VM Server for SPARC. Les sujets de base suivants sont abordés :

- “Transport XML” à la page 237
- “Protocole XML” à la page 238
- “Messages d'événements” à la page 243
- “Actions du gestionnaire de domaines logiques” à la page 247
- “Ressources et propriétés du gestionnaire de domaines logiques” à la page 248

Pour obtenir différents schémas à utiliser avec le gestionnaire de domaines logiques, reportez-vous à l'Annexe E, “Schémas XML du gestionnaire de domaines logiques”.

Transport XML

Les programmes externes peuvent utiliser le Extensible Messaging and Presence Protocol (XMPP - RFC 3920) pour communiquer avec le gestionnaire de domaines logiques. Le protocole XMPP est pris en charge par les connexions locales et distantes et est activé par défaut. Pour interrompre une connexion à distance, définissez la propriété SMF `ldmd/xmpp_enabled` sur `false` et redémarrez le gestionnaire de domaines logiques.

```
# svccfg -s ldmd/ldmd setprop ldmd/xmpp_enabled=false
# svcadm refresh ldmd
# svcadm restart ldmd
```

Remarque – La désactivation du serveur XMPP empêche également la migration du domaine et la reconfiguration dynamique de la mémoire.

Serveur XMPP

Le gestionnaire de domaines logiques implémente un serveur XMPP qui peut communiquer avec de nombreuses applications et bibliothèques client XMPP disponibles. Le gestionnaire de domaines logiques utilise les mécanismes de sécurité suivants :

- TLS (Transport Layer Security - Sécurité de la couche de transport) pour sécuriser le canal de communication entre le client et lui-même.
- SASL (Simple Authentication and Security Layer - Authentification simple et couche de sécurité) pour l'authentification. PLAIN est le seul mécanisme SASL pris en charge. Vous devez envoyer un nom d'utilisateur et un mot de passe au serveur afin qu'il puisse vous accorder les droits avant d'autoriser des opérations de surveillance ou de gestion.

Connexions locales

Le gestionnaire de domaines logiques détecte si les clients utilisateur sont en cours d'exécution sur le même domaine que lui-même et, le cas échéant, effectue une communication XMPP minimale avec ce client. Plus précisément, l'étape d'authentification SASL après la configuration d'un canal sécurisé par TLS est ignorée. L'authentification et l'autorisation sont effectuées en fonction des données d'identification du processus implémentant l'interface client.

Les clients peuvent choisir d'implémenter un client totalement XMPP ou d'exécuter simplement un analyseur syntaxique XML de transmission en continu, tel que l'analyseur syntaxique d'API simple `libxml2` pour XML (SAX). Quelle que soit la méthode, le client doit traiter une communication XMPP au moment de la communication TLS. Reportez-vous à la spécification XMPP pour la séquence nécessaire.

Protocole XML

À la fin de l'initialisation de la communication, les messages XML définis par les domaines logiques sont envoyés. Il a deux types généraux de messages XML :

- Les messages de demande et de réponse utilisent la balise `<LDM_interface>`. Ce type de message XML est utilisé pour communiquer les commandes et obtenir les résultats du gestionnaire de domaines logiques, comme lors de l'exécution des commandes utilisant l'interface de ligne de commande (CLI). Cette balise est également utilisée pour l'enregistrement et l'annulation de l'enregistrement des événements.
- Les messages d'événement utilisent la balise `<LDM_event>`. Ce type de message XML est utilisé pour signaler de manière asynchrone les événements postés par le gestionnaire de domaines logiques.

Messages de demande et de réponse

L'interface XML des domaines logiques a deux formats différents :

- Un format pour envoyer des commandes dans le gestionnaire de domaines logiques
- Un autre format pour que le gestionnaire de domaines logiques réponde sur le statut du message entrant et les actions demandées dans le message.

Ces deux formats partagent de nombreuses structures XML communes, mais sont séparés dans cette présentation pour une meilleure compréhension de leurs différences. Ce document contient également un schéma XML qui détaille le XML entrant et sortant combiné (voir “LDM_Event (Schéma XML)” à la page 263).

Messages de demande

Une demande XML entrante vers le gestionnaire de domaines logiques à son niveau le plus basique comprend une description d'une seule commande, s'appliquant à un seul objet. Les demandes plus complexes peuvent traiter plusieurs commandes et plusieurs objets par commande. Vous trouverez ci-dessous la structure d'une commande XML basique.

EXEMPLE D-1 Format d'une commande unique s'appliquant sur un seul objet

```
<LDM_interface version="1.0">
  <cmd>
    <action>Place command here</action>
    <option>Place options for certain commands here</option>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content> -->
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
          </Section>
          <!-- Note: More Sections sections can be placed here -->
        </Content>
      </Envelope>
    </data>
    <!-- Note: More Data sections can be placed here -->
  </cmd>
  <!-- Note: More Commands sections can be placed here -->
</LDM_interface>
```

Balise <LDM_interface>

Toutes les commandes envoyées au gestionnaire de domaines logiques doivent commencer par la balise <LDM_interface>. Tout document envoyé au gestionnaire de domaines logiques doit

comporter une balise unique `<LDM_interface>` contenue dedans. La balise `<LDM_interface>` doit comprendre un attribut de version comme indiqué dans l'[Exemple D-1](#).

Balise `<cmd>`

Dans la balise `<LDM_interface>`, le document doit inclure au moins une balise `<cmd>`. Chaque section `<cmd>` doit comporter uniquement une seule balise `<action>`. Utilisez la balise `<action>` pour décrire la commande à exécuter. Chaque balise `<cmd>` doit inclure au moins une balise `<data>` pour décrire les objets sur lesquels la commande doit être appliquée.

La balise `<cmd>` peut également avoir une balise `<option>`, qui est utilisée pour les options et les indicateurs associés à certaines commandes. Les commandes suivantes utilisent des options :

- La commande `remove-domain` peut utiliser l'option `-a`.
- La commande `stop-domain` peut utiliser l'option `-f`.
- La commande `cancel-operation` peut utiliser l'option `migration` ou `reconf`.
- La commande `add-sponfig` peut utiliser l'option `-r autosave-name`.
- La commande `remove-sponfig` peut utiliser l'option `-r`.
- La commande `list-sponfig` peut utiliser l'option `-r autosave-name`].

Balise `<data>`

Chaque section `<data>` contient une description d'un objet correspondant à la commande indiquée. Le format de la section de données repose sur la portion de schéma XML de la spécification brouillon OVF (Open Virtualization Format). Le schéma définit une section `<Envelope>` qui contient une balise `<References>` (non utilisée par les domaines logiques) et des sections `<Content>` et `<Section>`.

Pour les domaines logiques, la section `<Content>` est utilisée pour identifier et décrire un domaine particulier. Le nom de domaine dans l'attribut `id=` du nœud `<content>` identifie le domaine. Dans la section `<Content>`, il ya une ou plusieurs sections `<Section>` décrivant les ressources du domaine comme requis par la commande spécifique.

Si vous devez uniquement identifier un nom de domaine, il est alors inutile d'utiliser des balises `<Section>`. Au contraire de cela, si aucun identifiant de domaine n'est nécessaire pour la commande, vous devez alors fournir une section `<Section>`, décrivant les ressources nécessaires à la commande, en dehors de la section `<Content>`, mais toujours dans la section `<Envelope>`.

Une section `<data>` ne doit pas contenir une balise `<Envelope>` si les informations de l'objet peuvent être induites. Cette situation s'applique principalement aux demandes de surveillance de tous les objets applicables à une action, à l'enregistrement des événements et aux demandes d'annulation d'enregistrement.

Pour permettre l'utilisation du schéma de spécification OVF afin de définir correctement tous les types d'objets, deux types OVF supplémentaires ont été définis :

- `<gprop:GenericProperty>` (voir [“Schéma XML GenericProperty”](#) à la page 280.)
- `<Binding>` (Voir [“Binding_Type \(Schéma XML\)”](#) à la page 280).

La balise `<gprop:GenericProperty>` a été définie pour traiter la propriété de n'importe quel objet pour lequel la spécification OVF n'a pas de définition. Le nom de la propriété est défini dans l'attribut `key=` du nœud et la valeur de la propriété est le contenu du nœud. La balise `<binding>` est utilisée dans la sous-commande `list-bindings` pour définir les ressources associées à d'autres ressources.

Messages de réponse

Une réponse XML sortante correspond étroitement à la structure de la demande entrante en termes de commandes et d'objets inclus, en ajoutant une section `<Response>` pour chaque objet et commande indiqué, ainsi qu'une section globale `<Response>` pour la demande. Les sections `<Response>` fournissent des informations d'état et des messages comme décrit dans l'[Exemple D-2](#). Vous trouverez ci-dessous la structure d'une réponse à une demande XML basique.

EXEMPLE D-2 Format d'une réponse à une commande unique s'appliquant sur un seul objet

```
<LDM_interface version="1.0">
  <cmd>
    <action>Place command here</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content> -->
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>
                LDom Resource Type
              </rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">
                Property Value
              </gprop:GenericProperty>
            </Item>
          </Section>
          <!-- Note: More <Section> sections can be placed here -->
        </Content>
      </Envelope>
      <response>
        <status>success or failure</status>
        <resp_msg>Reason for failure</resp_msg>
      </response>
    </data>
    <!-- Note: More Data sections can be placed here -->
  </response>
  <status>success or failure</status>
```

EXEMPLE D-2 Format d'une réponse à une commande unique s'appliquant sur un seul objet (Suite)

```
<resp_msg>Reason for failure</resp_msg>
</response>
</cmd>
<!-- Note: More Command sections can be placed here -->
<response>
  <status>success or failure</status>
  <resp_msg>Reason for failure</resp_msg>
</response>
</LDM_interface>
```

Réponse globale

Cette section `<response>`, qui est l'enfant direct de la section `<LDM_interface>`, indique le succès ou l'échec global de toute la demande. À moins que le document XML entrant ne soit pas correctement formé, la section `<response>` comprend uniquement une balise `<status>`. Si l'état de la réponse indique un succès, toutes les commandes sur tous les objets ont abouti. Si cette réponse est un échec et qu'il n'y a pas de balise `<resp_msg>`, une des commandes incluse dans la demande d'origine a échoué. La balise `<resp_msg>` est utilisée uniquement pour décrire certains problèmes avec le document XML lui-même.

Réponse de la commande

La section `<response>` sous la section `<cmd>` alerte l'utilisateur du succès ou de l'échec de cette commande en particulier. La balise `<status>` indique si cette commande a abouti ou échoué. Tout comme pour la réponse globale, si la commande échoue, la section `<response>` comprend uniquement une balise `<resp_msg>` si le contenu de la section `<cmd>` de la demande n'est pas correctement formé. Sinon, l'état d'échec signifie qu'un des objets sur lesquels la commande a été exécutée a provoqué une erreur.

Réponse de l'objet

Enfin, la section `<data>` d'une section `<cmd>` comporte également une section `<response>`. Cela indique si la commande en cours d'exécution sur cet objet en particulier aboutit ou échoue. Si l'état de la réponse est SUCCESS, il n'y a pas de balise `<resp_msg>` dans la section `<response>`. Si l'état est FAILURE, il y a une ou plusieurs balises `<resp_msg>` dans la zone `<response>`, en fonction des erreurs rencontrées lors de l'exécution de la commande sur cet objet. Les erreurs des objets peuvent provenir de problèmes détectés lors de l'exécution de la commande ou d'un objet mal formé ou inconnu.

En plus de la section `<response>`, la section `<data>` peut contenir d'autres informations. Ces informations sont dans le même format qu'une zone `<data>` entrante, décrivant l'objet qui a provoqué l'erreur. Voir [“Balise `<data>`” à la page 240](#). Ces informations supplémentaires sont surtout utiles dans les cas suivants :

- Lorsqu'une commande échoue sur une section `<data>` spécifique mais aboutit pour d'autres sections `<data>`
- Lorsqu'une section `<data>` est transmise dans une commande et échoue pour certains domaines, mais aboutit pour d'autres

Messages d'événements

Au lieu d'interroger, vous pouvez vous abonner pour recevoir des notifications sur les événements lorsque certains changements d'état surviennent. Il y a trois types d'événements auxquels vous pouvez vous abonner, individuellement ou collectivement. Voir la section [“Types d'événements” à la page 244](#) pour obtenir des détails complets.

Enregistrement et annulation de l'enregistrement

Utilisez un message `<LDM_interface>` pour enregistrer les événements. Voir [“Balise `<LDM_interface>`” à la page 239](#). Les détails de la balise d'action définissent le type d'événement pour lequel s'enregistrer ou annuler l'enregistrement et la section `<data>` est laissée vide.

EXEMPLE D-3 Exemple de message de demande d'enregistrement d'événement

```
<LDM_interface version="1.0">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
  </cmd>
</LDM_interface>
```

Le gestionnaire de domaines logiques répond avec un message de réponse `<LDM_interface>` indiquant si l'enregistrement ou l'annulation de l'enregistrement a abouti.

EXEMPLE D-4 Exemple de message de réponse d'enregistrement d'événement

```
<LDM_interface version="1.0">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
      <response>
        <status>success</status>
      </response>
    </data>
  </response>
```

EXEMPLE D-4 Exemple de message de réponse d'enregistrement d'événement (Suite)

```

    <status>success</status>
  </response>
</cmd>
<response>
  <status>success</status>
</response>
</LDM_interface>

```

La chaîne d'action pour chaque type d'événement est répertoriée dans la sous-section des événements.

Messages <LDM_event>

Les messages d'événements ont le même format qu'un message <LDM_interface> entrant à l'exception que la balise de début pour le message est <LDM_event>. La balise d'action du message est l'action qui a été effectuée pour déclencher l'événement. La section de données du message décrit l'objet associé à l'événement : les détails dépendent du type d'événement survenu.

EXEMPLE D-5 Exemple de notification <LDM_event>

```

<LDM_event version='1.0'>
  <cmd>
    <action>Event command here</action>
    <data version='3.0'>
      <Envelope>
        <References/>
        <Content xsi:type='ovf:VirtualSystem_Type' ovf:id='ldg1'/>
        <Section xsi:type="ovf:ResourceAllocationSection_type">
          <Item>
            <rasd:OtherResourceType>LDM Resource Type</rasd:OtherResourceType>
            <gprop:GenericProperty
              key="Property name">Property Value</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_event>

```

Types d'événements

Ci-dessous se trouvent les types d'événements auxquels vous pouvez vous abonner :

- Événements du domaine
- Événements matériels
- Événements de progression

- Événements de ressource

Tous les événements correspondent aux sous-commandes `ldm`.

Événements du domaine

Les événements du domaine décrivent quelles actions peuvent être effectuées directement sur un domaine. Le tableau suivant présente les événements de domaine qui peuvent être répertoriés dans la balise `<action>` du message `<LDM_event>`.

Événements du domaine	Événements du domaine	Événements du domaine
<code>add-domain</code>	<code>remove-domain</code>	<code>bind-domain</code>
<code>unbind-domain</code>	<code>start-domain</code>	<code>stop-domain</code>
<code>domain-reset</code>	<code>panic-domain</code>	<code>migrate-domain</code>

Ces événements contiennent toujours *uniquement* une balise `<Content>` dans la section des données OVF qui décrit sur quel domaine l'événement est survenu. Pour vous abonner aux événements de domaine, envoyez un message `<LDM_interface>` avec la balise `<action>` définie sur **reg-domain-events**. Le désabonnement à ces événements nécessite un message `<LDM_interface>` avec la balise d'action définie sur **unreg-domain-events**.

Événements matériels

Les événements matériels concernent les modifications du matériel du système physique. Dans le cas du logiciel Oracle VM Server for SPARC, les seules modifications matérielles pouvant être effectuées sont celles sur le processeur de service (SP) lorsqu'un utilisateur ajoute, supprime ou définit une configuration de SP. Actuellement, les trois seuls événements de ce type sont :

- `add-sponfig`
- `set-sponfig`
- `remove-sponfig`

Les événements matériels contiennent toujours *uniquement* une balise `<Section>` dans la section des données OVF qui décrit la configuration SP sur laquelle l'événement se produit. Pour vous abonner à ces événements, envoyez un message `<LDM_interface>` avec la balise `<action>` définie sur **reg-hardware-events**. Le désabonnement à ces événements nécessite un message `<LDM_interface>` avec la balise `<action>` définie sur **unreg-hardware-events**.

Événements de progression

Les événements de progression sont émis pour les commandes à exécution longue, notamment la migration de domaine. Ces événements signalent la quantité de progression ayant été effectuée au cours du cycle de vie de la commande. Actuellement, seul l'événement `migration-process` est signalé.

Les événements de progression contiennent toujours *uniquement* une balise <Section> dans la section des données OVF qui décrit la configuration SP affectée par l'événement. Pour vous abonner à ces événements, envoyez un message <LDM_interface> avec la balise <action> définie sur reg-hardware-events. Le désabonnement à ces événements nécessite un message <LDM_interface> avec la balise <action> définie sur unreg-hardware-events.

La section <data> d'un événement de progression est composée d'une section <content> qui décrit le domaine concerné. Cette section <content> utilise une balise ldom_info <Section> pour mettre à jour la progression. Les propriétés génériques suivantes sont affichées dans la section ldom_info :

- --progress - Pourcentage de progression effectué par la commande
- --status - État de la commande qui peut être ongoing, failed ou done
- --source - Machine qui signale la progression

Événements de ressource

Les événements de ressource surviennent lorsque les ressources sont ajoutées, supprimées ou modifiées dans un domaine. La section de données de certains de ces événements contient la balise <Content> avec une balise <Section> donnant un nom de service dans la section des données OVF. Le tableau suivant présente les événements qui peuvent être répertoriés dans la balise <action> du message <LDM_event>.

Événements de ressource	Événements de ressource
add-vdiskserverdevice	remove-vdiskserverdevice
set-vdiskserverdevice	remove-vdiskserver
set-vconscon	remove-vconscon
set-vswitch	remove-vswitch
remove-vdpcs	

Les événements de ressource restants contiennent toujours *uniquement* une balise <Content> dans la section des données OVF qui décrit sur quel domaine l'événement est survenu.

Événements de ressource	Événements de ressource	Événements de ressource
add-vcpu	add-crypto	add-memory
add-io	add-variable	add-vconscon
add-vdisk	add-vdiskserver	add-vnet
add-vswitch	add-vdpcs	add-vdpsc

Événements de ressource	Événements de ressource	Événements de ressource
set-vcpu	set-crypto	set-memory
set-variable	set-vnet	set-vconsole
set-vdisk	remove-vcpu	remove-crypto
remove-memory	remove-io	remove-variable
remove-vdisk	remove-vnet	remove-udpcc

Pour vous abonner aux événements de ressource, envoyez un message `<LDM_interface>` avec la balise `<action>` définie sur **reg-resource-events**. Le désabonnement à ces événements nécessite un message `<LDM_interface>` avec la balise `<action>` définie sur **unreg-resource-events**.

Tous les événements

Vous pouvez également vous abonner pour écouter ces trois types d'événements sans vous abonner à chacun individuellement. Pour vous abonner à ces trois types d'événements en même temps, envoyez un message `<LDM_interface>` avec la balise `<action>` définie sur **reg-all-events**. Le désabonnement à ces événements nécessite un message `<LDM_interface>` avec la balise `<action>` définie sur **unreg-all-events**.

Actions du gestionnaire de domaines logiques

Les commandes indiquées dans la balise `<action>`, à l'exception des commandes `*-*-events`, correspondent à celles de l'interface de ligne de commande `ldm`. Pour plus de détails sur les sous-commandes `ldm`, reportez-vous à la page de manuel [ldm\(1M\)](#).

Remarque – L'interface XML ne prend *pas* en charge l'accès ou la commande *aliases* prise en charge par la CLI du gestionnaire de domaines logiques.

Les chaînes prises en charge de la balise `<action>` sont les suivantes :

Actions du gestionnaire de domaines logiques	Actions du gestionnaire de domaines logiques	Actions du gestionnaire de domaines logiques
list-bindings	list-services	list-constraints
list-devices	add-domain	remove-domain
list-domain	start-domain	stop-domain
bind-domain	unbind-domain	add-io

Actions du gestionnaire de domaines logiques	Actions du gestionnaire de domaines logiques	Actions du gestionnaire de domaines logiques
remove-io	add-mau	set-mau
remove-mau	add-memory	set-memory
remove-memory	remove-reconf	add-spconfig
set-spconfig	remove-spconfig	list-spconfig
add-variable	set-variable	remove-variable
list-variable	add-vconscon	set-vconscon
remove-vconscon	set-vconsole	add-vcpu
set-vcpu	remove-vcpu	add-vdisk
remove-vdisk	add-vdiskserver	remove-vdiskserver
add-udpcc	remove-udpcc	add-udpccs
remove-udpccs	add-vdiskserverdevice	remove-vdiskserverdevice
add-vnet	set-vnet	remove-vnet
add-vswitch	set-vswitch	remove-vswitch
reg-domain-events	unreg-domain-events	reg-resource-events
unreg-resource-events	reg-hardware-events	unreg-hardware-events
reg-all-events	unreg-all-events	migrate-domain
cancel-operation	set-domain	

Ressources et propriétés du gestionnaire de domaines logiques

Vous trouverez ci-dessous les ressources et les propriétés du gestionnaire de domaines logiques qui peuvent être définies pour chacune de ces ressources. Les ressources et les propriétés sont présentées en **gras** dans les exemples XML. Ces exemples présentent des ressources, non une sortie de liaison. La sortie de contrainte peut être utilisée pour créer une entrée pour les actions du gestionnaire de domaines logiques. L'exception à cela est la sortie de la migration de domaine. Voir [“Migration de domaine” à la page 259](#). Chaque ressource est définie dans une section OVF <Section> et est définie par une balise <rasd:OtherResourceType>.

Ressource d'informations sur le domaine (ldom_info)

EXEMPLE D-6 Exemple de sortie XML ldom_info

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="primary">
    <Section xsi:type="ovf:ResourceAllocationSection_type">
      <Item>
        <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
        <rasd:Address>00:03:ba:d8:ba:f6</rasd:Address>
        <gprop:GenericPropertykey="hostid">83d8baf6</gprop:GenericProperty>
        <gprop:GenericProperty key="master">plum</gprop:GenericProperty>
        <gprop:GenericProperty key="failure-policy">reset</gprop:GenericProperty>
        <gprop:GenericProperty key="progress">45%</gprop:GenericProperty>
        <gprop:GenericProperty key="status">ongoing</gprop:GenericProperty>
        <gprop:GenericProperty key="source">dt90-319</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

La ressource ldom_info est toujours contenue dans une section <Content>. Les propriétés suivantes de la ressource ldom_info sont des propriétés facultatives :

- <rasd:Address>, qui définit l'adresse MAC à assigner au domaine.
- <gprop:GenericPropertykey="failure-policy">, qui définit comment les domaines esclaves doivent se comporter, si le domaine maître est en panne. La valeur par défaut est ignore. Les valeurs valides de la propriétés sont indiquées ci-après :
 - ignore ignore les erreurs du domaine maître (les domaines esclaves ne sont pas affectés).
 - panic panique tous les domaines esclaves lorsque le domaine maître est en panne.
 - reset réinitialise tous les domaines esclaves lorsque le domaine maître est en panne.
 - stop arrête tous les domaines esclaves lorsque le domaine maître est en panne.
- <gprop:GenericPropertykey="hostid">, qui définit l'ID hôte à assigner au domaine.
- <gprop:GenericPropertykey="master">, qui définit jusqu'à quatre noms de domaine maître séparés par des virgules.
- <gprop:GenericPropertykey="progress">, qui définit le pourcentage de progression effectué par la commande.
- <gprop:GenericPropertykey="source">, qui définit le compte-rendu de la machine sur la progression de la commande.
- <gprop:GenericPropertykey="status">, qui définit l'état de la commande (done, failed ou ongoing).

Ressource du CPU (cpu)

L'équivalent des actions de demande XML `add -vcpu`, `set -vcpu` et `remove -vcpu` est défini à la valeur de la balise `<gpropGenericProperty key="wcore">` comme suit :

- Si l'option `-c` est utilisée, définissez la propriété `wcore` sur le nombre du total de serveurs de base spécifié.
- Si l'option `-c` n'est *pas* utilisée, définissez la propriété `wcore` sur `0`.

Notez que la propriété des unités d'allocation, `<rasd:AllocationUnits>`, pour la ressource `cpu` spécifie toujours le nombre de CPU virtuels et non le nombre de serveurs de base.

EXEMPLE D-7 Exemple de XML `cpu`

L'exemple suivant présente la demande XML équivalente pour la commande `ldm add -vcpu -c 1 ldg1` :

```
<?xml version="1.0"?>
<LDM_interface version="1.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding">
  <cmd>
    <action>add-vcpu</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1">
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
              <rasd:AllocationUnits>8</rasd:AllocationUnits>
              <gprop:GenericProperty key="wcore">1</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

Une ressource `cpu` est toujours contenue dans une section `<Content>`.

Ressource MAU (mau)

Remarque – La ressource mau est une unité cryptographique prise en charge sur un serveur pris en charge. Actuellement, les deux unités cryptographiques prises en charge sont l'unité arithmétique modulaire (MAU) et la file d'attente de mot de contrôle (CWQ).

EXEMPLE D-8 Exemple de XML mau

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>mau</rasd:OtherResourceType>
        <rasd:AllocationUnits>1</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

Une ressource mau est toujours contenue dans une section <Content>. La seule propriété est la balise <rasd:AllocationUnits>, qui signifie le nombre de MAU ou d'autres unités cryptographiques.

Ressource de mémoire (memory)

EXEMPLE D-9 Exemple de XML memory

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>memory</rasd:OtherResourceType>
        <rasd:AllocationUnits>4G</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

Une ressource memory est toujours contenue dans une section <Content>. La seule propriété est la balise <rasd:AllocationUnits>, qui signifie la quantité de mémoire.

Ressource de serveur de disque virtuel (vds)

EXEMPLE D-10 Exemple de XML (vds)

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vds</rasd:OtherResourceType>
        <gprop:GenericProperty
          key="service_name">vdstmp</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

Une ressource de serveur de disque virtuel (vds) peut se trouver dans une section <Content> comme faisant partie de la description du domaine, ou elle peut apparaître seule dans une section <Envelope>. La seule propriété est la balise <gprop:GenericProperty> avec une clé de service_name et qui contient le nom de la ressource vds étant décrite.

Ressource du volume de disque virtuel (vds_volume)

EXEMPLE D-11 Exemple de XML vds_volume

```
<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vds_volume</rasd:OtherResourceType>
      <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
      <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
      <gprop:GenericProperty key="block_dev">
        opt/SUNWldm/domain_disks/testdisk1</gprop:GenericProperty>
      <gprop:GenericProperty key="vol_opts">ro</gprop:GenericProperty>
      <gprop:GenericProperty key="mpgroup">mpgroup-name</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

Une ressource vds_volume peut se trouver dans une section <Content> comme faisant partie de la description du domaine, ou elle peut apparaître seule dans une section <Envelope>. Elle doit comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- vol_name – Nom du volume
- service_name – Nom du serveur de disque virtuel auquel ce volume doit être associé
- block_dev – Nom du fichier ou du périphérique à associer à ce volume

Une ressource `vds_volume` peut également avoir les propriétés suivantes :

- `vol_opts` – Une ou plusieurs des chaînes suivantes séparées par des virgules :
{`ro`,`slice`,`excl`}
- `mpgroup` – Nom du groupe multivoie (basculément)

Ressource de disque (disk)

EXEMPLE D-12 Exemple de XML `disk`

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>disk</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdisk_name">vdisk0</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">primary-vds0</gprop:GenericProperty>
        <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
        <gprop:GenericProperty key="timeout">60</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

Une ressource `disk` est toujours contenue dans une section `<Content>`. Elle doit comporter des balises `<gprop:GenericProperty>` avec les clés suivantes :

- `vdisk_name` – Nom du disque virtuel
- `service_name` – Nom du serveur de disque virtuel auquel ce disque virtuel doit être associé
- `vol_name` – Périphérique du service de disque virtuel avec lequel le disque virtuel doit être associé

La ressource `disk` peut également avoir la propriété `timeout`, qui est la valeur de temporisation en secondes pour l'établissement d'une connexion entre un client de disque virtuel (`vdc`) et un serveur de disque virtuel (`vds`). S'il y a plusieurs chemins vers le disque virtuel (`vdisk`), le `vdc` peut donc essayer de se connecter sur un `vds` différent, et la valeur de temporisation garantit qu'une connexion à un `vds` est établie pendant le délai imparti.

Ressource de commutateur virtuel (vsw)

EXEMPLE D-13 Exemple de XML `vsw`

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
```

EXEMPLE D-13 Exemple de XML vsw (Suite)

```

<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>vsw</rasd:OtherResourceType>
    <gprop:GenericProperty key="service_name">vsw1-ldg1</gprop:GenericProperty>
    <gprop:GenericProperty key="dev_path">bge0</gprop:GenericProperty>
    <gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
    <rasd:Address>00:14:4f:fc:00:01</rasd:Address>
    <gprop:GenericProperty key="mode">sc</gprop:GenericProperty>
    <gprop:GenericProperty key="pvid">12345678</gprop:GenericProperty>
    <gprop:GenericProperty key="vid">87654321</gprop:GenericProperty>
  </Item>
</Section>
</Content>
</Envelope>

```

Une ressource vsw peut se trouver dans une section <Content> comme faisant partie de la description du domaine, ou elle peut apparaître seule dans une section <Envelope>. Elle doit comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- service_name – Nom à assigner au commutateur virtuel.
- linkprop – Indique si le périphérique virtuel doit obtenir les mises à jour d'état du lien physique. Lorsque la valeur est phys-state, le périphérique virtuel obtient les mises à jour de l'état du lien physique. Lorsque la valeur est vide, le périphérique virtuel n'obtient pas les mises à jour de l'état du lien physique. Par défaut, le périphérique physique n'obtient pas les mises à jour de l'état du lien physique.
- dev_path – Chemin du périphérique physique à associer à ce commutateur virtuel

La ressource vsw peut également avoir les propriétés suivantes :

- <rasd:Address> – Assigne une adresse MAC au commutateur virtuel
- pvid – L'identificateur (ID) du réseau local virtuel (VLAN) du port indique le VLAN dont le réseau virtuel doit être membre, en mode non balisé.
- vid – L'identificateur (ID) du réseau local virtuel (VLAN) indique le VLAN dont le réseau virtuel et le commutateur virtuel doivent être membres, en mode balisé.
- mode – sc pour la prise en charge de la pulsation du cluster Oracle Solaris.

Ressource réseau (network)

EXEMPLE D-14 Exemple de XML network

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>

```

EXEMPLE D-14 Exemple de XML network (Suite)

```

<rasd:OtherResourceType>network</rasd:OtherResourceType>
<gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
<gprop:GenericProperty key="vnet_name">ldg1-vnet0</gprop:GenericProperty>
<gprop:GenericProperty
  key="service_name">primary-vsw0</gprop:GenericProperty>
<rasd:Address>00:14:4f:fc:00:01</rasd:Address>
</Item>
</Section>
</Content>
</Envelope>

```

Une ressource network est toujours contenue dans une section <Content>. Elle doit comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- linkprop – Indique si le périphérique virtuel doit obtenir les mises à jour d'état du lien physique. Lorsque la valeur est phys-state, le périphérique virtuel obtient les mises à jour de l'état du lien physique. Lorsque la valeur est vide, le périphérique virtuel n'obtient pas les mises à jour de l'état du lien physique. Par défaut, le périphérique physique n'obtient pas les mises à jour de l'état du lien physique.
- vnet_name – Nom du réseau virtuel (vnet)
- service_name – Nom du commutateur virtuel (vswitch) auquel ce réseau virtuel doit être associé

La ressource network peut également avoir les propriétés suivantes :

- <rasd:Address> – Assigne une adresse MAC au commutateur virtuel
- pvid – L'identificateur (ID) du réseau local virtuel (VLAN) du port indique le VLAN dont le réseau virtuel doit être membre, en mode non balisé.
- vid – L'identificateur (ID) du réseau local virtuel (VLAN) indique le VLAN dont le réseau virtuel et le commutateur virtuel doivent être membres, en mode balisé.
- mode – hybrid pour activer l'E/S hybride pour ce réseau virtuel.

Ressource de concentrateur de console virtuelle (vcc)

EXEMPLE D-15 Exemple de XML vcc

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vcc</rasd:OtherResourceType>
        <gprop:GenericProperty key="service_name">vcc1</gprop:GenericProperty>
        <gprop:GenericProperty key="min_port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="max_port">6100</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

EXEMPLE D-15 Exemple de XML vcc (Suite)

```

    </Item>
  </Section>
</Content>
</Envelope>

```

Une ressource vcc peut se trouver dans une section <Content> comme faisant partie de la description du domaine, ou elle peut apparaître seule dans une section <Envelope>. Elle peut comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- service_name – Nom à assigner au service de concentrateur de console virtuelle
- min_port – Numéro de port minimum à associer à ce vcc
- max_port – Numéro de port maximum à associer à ce vcc

Ressource de variable (var)

EXEMPLE D-16 Exemple de XML var

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>var</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">test_var</gprop:GenericProperty>
        <gprop:GenericProperty key="value">test1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

Une ressource var est toujours contenue dans une section <Content>. Elle peut comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- name – Name de la variable
- value – Valeur de la variable

Ressource de périphérique d'E/S physique (physio_device)

EXEMPLE D-17 Exemple de XML physio_device

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">

```

EXEMPLE D-17 Exemple de XML `physio_device` (Suite)

```

<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
    <gprop:GenericProperty key="name">pci@780</gprop:GenericProperty>
  </Item>
</Section>
</Content>
</Envelope>

```

Une ressource `physio_device` est toujours contenue dans une section `<Content>`. La seule propriété est la balise `<gprop:GenericProperty>` avec la valeur de propriété de clé `name`, qui est le nom du périphérique d'E/S étant décrit.

Ressource de configuration du SP (`spconfig`)

EXEMPLE D-18 Exemple de XML `spconfig`

```

<Envelope>
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>
      <rasd:OtherResourceType>spconfig</rasd:OtherResourceType>
      <gprop:GenericProperty
        key="spconfig_name">primary</gprop:GenericProperty>
      <gprop:GenericProperty
        key="spconfig_status">current</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

Une configuration de processeur de service (SP) (`spconfig`) apparaît toujours seule dans une section `<Envelope>`. Elle peut comporter des balises `<gprop:GenericProperty>` avec les clés suivantes

- `spconfig_name` – Nom d'une configuration à stocker sur le SP
- `spconfig_status` – État actuel d'une configuration de SP spécifique. Cette propriété est utilisée dans la sortie d'une commande `ldm list-spconfig`.

Ressource de service de canal de plan de données virtuelles (`vdpcs`)

EXEMPLE D-19 Exemple de XML `vdpcs`

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">

```

EXEMPLE D-19 Exemple de XML vdpcs (Suite)

```

<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>vdpcs</rasd:OtherResourceType>
    <gprop:GenericProperty key="service_name">dg1-vdpcs</gprop:GenericProperty>
  </Item>
</Section>
</Content>
</Envelope>

```

Cette ressource ne présente un intérêt que dans un environnement DPS Netra. Une ressource vdpcs peut se trouver dans une section <Content> comme faisant partie de la description du domaine, ou elle peut apparaître seule dans une section <Envelope>. La seule propriété est la balise <gprop:GenericProperty> avec la valeur de propriété de clé service_name, qui est le nom de la ressource du service de plan de données virtuelles (vdpcs) étant décrite.

Ressource de client de canal de plan de données virtuelles (vdpc)

EXEMPLE D-20 Exemple de XML vdpc

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vdpc</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdpc_name">vdpc</gprop:GenericProperty>
        <gprop:GenericProperty
          key="service_name">ldg1-vdpc</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

Cette ressource ne présente un intérêt que dans un environnement DPS Netra. Un ressource de client de canal de plan de données virtuelles est toujours contenue dans une section <Content>. Elle peut comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- vdpc_name – Nom du client du canal de plan de données virtuelles (vdpc)
- service_name – Nom du service de canal de plan de données virtuelles vdpc auquel vdpc doit être associé

Ressource de console (console)

EXEMPLE D-21 Exemple de XML console

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>console</rasd:OtherResourceType>
        <gprop:GenericProperty key="port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">vcc2</gprop:GenericProperty>
        <gprop:GenericProperty key="group">group-name</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

Une ressource console est toujours contenue dans une section <Content>. Elle peut comporter des balises <gprop:GenericProperty> avec les clés suivantes :

- port – Port sur lequel passer sur cette console virtuelle (console)
- service_name – Service de concentrateur de console virtuelle (vcc) auquel associer cette console
- group – Nom du groupe auquel associer cette console

Migration de domaine

Cet exemple présente le contenu de la section <data> pour une sous-commande migrate-domain.

EXEMPLE D-22 Exemple de section migrate-domain <data>

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
    <Section xsi:type="ovf:ResourceAllocationSection_Type">
      <Item>
        <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
        <gprop:GenericProperty key="target">target-host</gprop:GenericProperty>
        <gprop:GenericProperty key="username">user-name</gprop:GenericProperty>
        <gprop:GenericProperty key="password">password</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

où :

- Le premier nœud <Content> (sans section <ldom_info>) est le domaine source à migrer.
- Le second nœud <Content> (avec une section <ldom_info>) est le domaine cible vers lequel migrer. Les noms des domaines source et cible peuvent être identiques.
- La section <ldom_info> du domaine cible décrit la machine que laquelle migrer et les détails nécessaires à la migration sur cette machine :
 - `target-host` est la machine cible vers laquelle migrer.
 - `user-name` est le nom d'utilisateur pour la connexion à la machine cible. Doit être codé en SASL 64 bits.
 - `password` est le mot de passe à utiliser pour la connexion à la machine cible. Doit être codé en SASL 64 bits.

Remarque – Le gestionnaire de domaines logiques utilise `sasl_decode64()` pour décoder le nom et le mot de passe de l'utilisateur cible et utilise `sasl_encode64()` pour coder ces valeurs. Le codage SASL 64 est équivalent au codage `base64`.

Schémas XML du gestionnaire de domaines logiques

Cette annexe fournit différents schémas XML à utiliser avec le gestionnaire de domaines logiques.

Ce chapitre aborde les sujets suivants :

- “LDM_interface (Schéma XML)” à la page 261
- “LDM_Event (Schéma XML)” à la page 263
- “Schéma ovf-envelope.xsd” à la page 264
- “Schéma ovf-section.xsd” à la page 266
- “Schéma ovf-core.xsd” à la page 266
- “Schéma ovf-virtualhardware.xsc” à la page 271
- “Schéma cim-rasd.xsd” à la page 272
- “Schéma cim-vssd.xsd” à la page 276
- “Schéma cim-common.xsd” à la page 276
- “Schéma XML GenericProperty” à la page 280
- “Binding_Type (Schéma XML)” à la page 280

LDM_interface (Schéma XML)

Ce schéma est un instantané de la spécification brouillon OVF (Open Virtualization Format) version 0.98.

EXEMPLE E-1 LDM_interface (Schéma XML)

```
<?xml version="1.0"?>
xs:schema
  xmlns:ovf="/var/opt/SUNWldom/envelope"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:import namespace="/var/opt/SUNWldom/envelope" schemaLocation="ovf-envelope.xsd"/>

  <xs:annotation>
    <xs:documentation>
      Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
```

EXEMPLE E-1 LDM_interface (Schéma XML) (Suite)

```

    </xs:documentation>
  </xs:annotation>

<!--
=====
Type Definitions
=====
-->
<xs:simpleType name="statusStringType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="success"/>
    <xs:enumeration value="failure"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="responseType">
  <xs:sequence>
    <xs:element name="status" type="statusStringType"/>
    <xs:element name="resp_msg" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- LDM interface document -->
<xs:element name="LDM_interface">
  <xs:complexType>
    <xs:sequence>

      <!-- START cmd -->
      <xs:element name="cmd" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="action" type="xs:string" minOccurs="0"/>

            <!-- START data -->
            <xs:element name="data" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:choice minOccurs="1" maxOccurs="unbounded">

                  <!--OVF Envelope Version 0.9 -->
                  <xs:element name="Envelope" type="ovf:Envelope_Type"/>
                  <!-- DATA response -->
                  <xs:element name="response" type="responseType" minOccurs="0" maxOccurs="1"/>
                </xs:choice>
                <xs:attribute name="version" type="xs:string" use="required"/>
              </xs:complexType>
            </xs:element> <!-- END data -->

            <!-- CMD response -->
            <xs:element name="response" type="responseType" minOccurs="0" maxOccurs="1"/>

          </xs:sequence>
        </xs:complexType>
      </xs:element> <!-- END cmd -->

      <!-- DOCUMENT response -->
      <xs:element name="response" type="responseType" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element> <!-- END LDM_interface -->

```

EXEMPLE E-1 LDM_interface (Schéma XML) (Suite)

```

    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element> <!-- LDM interface document -->

</xs:schema>

```

LDM_Event (Schéma XML)

EXEMPLE E-2 LDM_Event (Schéma XML)

```

<?xml version="1.0"?>
<xs:schema
  xmlns:ovf="/var/opt/SUNWldom/envelope"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="/var/opt/SUNWldom/envelope" schemaLocation="ovf-envelope.xsd"/>

  <xs:annotation>
    <xs:documentation>
      Copyright (c) 2007, 2010, Oracle and/or its affiliates. All rights reserved.
    </xs:documentation>
  </xs:annotation>

  <!-- LDM interface document -->
  <xs:element name="LDM_event">
    <xs:complexType>
      <xs:sequence>

        <!-- START cmd -->
        <xs:element name="cmd" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="action" type="xs:string" minOccurs="0"/>

              <!-- START data -->
              <xs:element name="data" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:choice minOccurs="1" maxOccurs="unbounded">

                    <!--OVF Envelope Version 0.9 -->
                    <xs:element name="Envelope" type="ovf:Envelope_Type"/>

                  </xs:choice>
                  <xs:attribute name="version" type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element> <!-- END data -->

            </xs:sequence>
          </xs:complexType>
        </xs:element> <!-- END cmd -->

      </xs:sequence>
    </xs:complexType>
  </xs:element> <!-- END LDM_event -->

```

EXEMPLE E-2 LDM_Event (Schéma XML) (Suite)

```

    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element> <!-- LDM interface document -->

</xs:schema>

```

Schéma ovf-envelope.xsd

EXEMPLE E-3 Schéma ovf-envelope.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/envelope"
  xmlns:ovf="/var/opt/SUNWldom/envelope"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Include virtual hardware schema -->
  <xs:include schemaLocation="./ovf-section.xsd"/>
  <xs:include schemaLocation="./cim-virtualhardware.xsd"/>
  <xs:include schemaLocation="./ovf-core.xsd"/>

  <!-- Root element of a OVF package -->
  <xs:element name="Envelope" type="ovf:Envelope_Type"/>

  <xs:complexType name="Envelope_Type">
    <xs:sequence>
      <!--- References to all external files -->
      <xs:element name="References" type="ovf:References_Type"/>

      <!-- Package level meta-data -->
      <xs:element name="Section" type="ovf:Section_Type" minOccurs="0" maxOccurs="unbounded"/>

      <!-- Content. A virtual machine or a vService -->
      <xs:element name="Content" type="ovf:Entity_Type" minOccurs="0" maxOccurs="unbounded"/>

      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="signed" type="xs:boolean" use="optional"/>
    <xs:attribute name="manifest" type="xs:boolean" use="optional"/>
    <xs:anyAttribute namespace="##any"/>
  </xs:complexType>

  <xs:complexType name="References_Type">
    <xs:sequence>
      <xs:element name="File" type="ovf:File_Type" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:anyAttribute namespace="##any"/>
</xs:schema>

```

EXEMPLE E-3 Schéma ovf-envelope.xsd (Suite)

```

<!-- Type for an external reference to a resource -->
<xs:complexType name="File_Type">
  <xs:sequence>
    <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>

  <!-- Reference key used in other parts of the package -->
  <xs:attribute name="id" type="xs:string" use="required"/>
  <!-- Same as using a single part element -->
  <xs:attribute name="href" type="xs:string" use="required"/>
  <!-- Size in bytes of the files (if known) -->
  <xs:attribute name="size" type="xs:integer" use="optional"/>
  <!-- Estimated size in bytes of the files (if a good guess is known) -->
  <xs:attribute name="estSize" type="xs:integer" use="optional"/>
  <!-- Compression type (gzip or bzip2) -->
  <xs:attribute name="compression" type="xs:string" use="optional"/>
  <!-- Chunk size (except of last chunk) -->
  <xs:attribute name="chunkSize" type="xs:long" use="optional"/>

  <xs:anyAttribute namespace="##any"/>
</xs:complexType>

<!-- Base class for an entity -->
<xs:complexType name="Entity_Type" abstract="true">
  <xs:sequence>
    <xs:element name="Info" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Section" type="ovf:Section_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<!-- A Virtual Machine Entity -->
<xs:complexType name="VirtualSystem_Type">
<xs:complexContent>
  <xs:extension base="ovf:Entity_Type"> </xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- A Composite Service -->
<xs:complexType name="VirtualSystemCollection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Entity_Type">
      <xs:sequence>
        <xs:element name="Content" type="ovf:Entity_Type" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>

```

Schéma ovf-section.xsd

EXEMPLE E-4 Schéma ovf-section.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/envelope"
  xmlns:ovf="/var/opt/SUNWldom/envelope"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <!-- The base class for a section. Subclassing this is the most common form of extensibility -->
  <xs:complexType name="Section_Type" abstract="true">
    <xs:sequence>
      <!-- The info element specifies the meaning of the section. This is typically shown
           if the section is not understood by the importer -->
      <xs:element name="Info" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  <!-- Whether the import should fail or not, if the section is not understood -->
  <xs:attribute name="required" type="xs:boolean" use="optional"/>
  <xs:anyAttribute namespace="##any"/>
  <!-- Subtypes defines more specific elements -->
</xs:complexType>

<!-- A basic type for a localizable string -->
<xs:complexType name="Info_Type">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="xml:lang"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>
```

Schéma ovf-core.xsd

EXEMPLE E-5 Schéma ovf-core.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/envelope"
  xmlns:ovf="/var/opt/SUNWldom/envelope"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:include schemaLocation="ovf-section.xsd"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <!-- A user defined annotation on an entity -->
  <xs:complexType name="AnnotationSection_Type">
    <xs:complexContent>
      <xs:extension base="ovf:Section_Type">
        <xs:sequence>
```

EXEMPLE E-5 Schéma ovf-core.xsd (Suite)

```

<!-- Several localized annotations can be included -->
<xs:element name="Annotation" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
<xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
  maxOccurs="unbounded"/>
<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute namespace="##any"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- Product information about a virtual appliance -->
<xs:complexType name="ProductSection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>
        <xs:element name="Product" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Vendor" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Version" type="xs:string" minOccurs="0"/>
        <xs:element name="Full-version" type="xs:string" minOccurs="0"/>
        <xs:element name="ProductUrl" type="xs:string" minOccurs="0"/>
        <xs:element name="VendorUrl" type="xs:string" minOccurs="0"/>
        <xs:element name="AppUrl" type="xs:string" minOccurs="0"/>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Configuration parameters that can be passed to the virtual machine for
application-level configuration -->
<xs:complexType name="PropertySection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>
        <xs:element name="Property" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Description" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
              <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
              <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="key" type="xs:string"/>
            <xs:attribute name="type" type="xs:string"/>
            <xs:attribute name="configurableByUser" type="xs:boolean" use="optional"/>
            <xs:attribute name="configurableAtRuntime" type="xs:boolean" use="optional"/>
            <xs:attribute name="defaultValue" type="xs:string" use="optional"/>
            <xs:anyAttribute namespace="##any"/>
          </xs:complexType>
        </xs:element>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

EXEMPLE E-5 Schéma ovf-core.xsd (Suite)

```

    <!-- A comma-separated list of transports that are supported by the virtual machine to
    access the OVF environment. -->
    <xs:attribute name="transport" type="xs:string" use="optional"/>
    <xs:anyAttribute namespace="##any"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- Provides descriptions for the logical networks used within the package. These descriptions are
typically used as an aid when the package is deployed. -->
<xs:complexType name="NetworkSection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>
        <xs:element name="Network" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Description" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
              <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
              <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name" type="xs:string" use="required"/>
            <xs:anyAttribute namespace="##any"/>
          </xs:complexType>
        </xs:element>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Provides meta-information description of the virtual disks in the package -->
<xs:complexType name="DiskSection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>
        <xs:element name="Disk" type="ovf:VirtualDiskDesc_Type" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- Disk -->
<xs:complexType name="VirtualDiskDesc_Type">
  <!-- A logical ID for the virtual disk within this package -->
  <xs:attribute name="diskId" type="xs:string" use="required"/>
  <!-- A file reference to the virtual disk file. If this is not specified a blank virtual disk is
  created of the given size -->
  <xs:attribute name="fileRef" type="xs:string" use="optional"/>
  <!-- Capacity in bytes. The capacity can be specified as either a size or as a reference to a property
  using $(property_name) -->

```

EXEMPLE E-5 Schéma ovf-core.xsd (Suite)

```

<xs:attribute name="capacity" type="xs:string" use="required"/>
<!-- Format of the disk. The format is an URL that identifies the disk type,
     e.g., http://www.vmware.com/format/vmdk.html#sparse -->
<xs:attribute name="format" type="xs:string" use="required"/>
<!-- Populated size of disk. This is an estimation of how much storage the disk needs if backed by
     a non pre-allocated (aka. sparse) disk. This size does not take the meta-data into
     account used by a sparse disk. -->
<xs:attribute name="populatedSize" type="xs:long" use="optional"/>
<!-- Reference to a potential parent disk -->
<xs:attribute name="parentRef" type="xs:string" use="optional"/>
</xs:complexType>

<!-- CPU Architecture requirements for the guest software. -->
<xs:complexType name="CpuCompatibilitySection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>
        <xs:element name="Level" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="level" type="xs:int" use="optional"/>
            <xs:attribute name="eax" type="xs:string" use="optional"/>
            <xs:attribute name="ebx" type="xs:string" use="optional"/>
            <xs:attribute name="ecx" type="xs:string" use="optional"/>
            <xs:attribute name="edx" type="xs:string" use="optional"/>
          </xs:complexType>
        </xs:element>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="Vendor" type="xs:string"/>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Specification of the operating system installed in the guest -->
<xs:complexType name="OperatingSystemSection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>
        <xs:element name="Description" type="ovf:Info_Type" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <!-- The IDs are the enumeration used in CIM_OperatingSystem_Type -->
      <xs:attribute name="id" type="xs:string"/>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- End-User License Agreement -->
<xs:complexType name="EulaSection_Type">
  <xs:complexContent>
    <xs:extension base="ovf:Section_Type">
      <xs:sequence>

```

EXEMPLE E-5 Schéma ovf-core.xsd (Suite)

```

<!-- Contains the license agreement in plain text. Several different locales can be
specified -->
<xs:element name="License" type="ovf:Info_Type" minOccurs="1" maxOccurs="unbounded"/>
<xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute namespace="##any"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- For a VirtualSystemCollection, this section is used to specify the order in which the
contained entities are to be powered on. -->
<xs:complexType name="StartupSection_Type">
<xs:complexContent>
<xs:extension base="ovf:Section_Type">
<xs:sequence>
<xs:element name="item" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<!-- Id of entity in collection -->
<xs:attribute name="id" type="xs:string"/>
<!-- Startup order. Entities are started up starting with lower-numbers first. Items with
same order identifier may be started up concurrently or in any order.
The order is reversed for shutdown. -->
<xs:attribute name="order" type="xs:int"/>
<!-- Delay in seconds to wait for the power on to complete -->
<xs:attribute name="startDelay" type="xs:int"/>
<!-- Whether to resume power-on sequence, once the guest reports ok. -->
<xs:attribute name="waitingForGuest" type="xs:boolean"/>
<!-- Delay in seconds to wait for the power on to complete -->
<xs:attribute name="stopDelay" type="xs:int"/>
<!-- Stop action to use. Valid values are: 'powerOn' (default), 'none'. -->
<xs:attribute name="startAction" type="xs:string"/>
<!-- Stop action to use. Valid values are: 'powerOff' (default), 'guestShutdown',
'suspend'. -->
<xs:attribute name="stopAction" type="xs:string"/>
<xs:anyAttribute namespace="##any"/>
</xs:complexType>
</xs:element>
<xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<!-- A comma-separated list of transports that the virtual machine supports to provide
feedback. -->
<xs:anyAttribute namespace="##any"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- If this section is present, it indicates that the virtual machine needs to be initially
booted to install and configure the software. -->
<xs:complexType name="InstallSection_Type">
<xs:complexContent>
<xs:extension base="ovf:Section_Type">
<xs:sequence>
<xs:any namespace="##targetNamespace" processContents="lax" minOccurs="0"

```

EXEMPLE E-5 Schéma ovf-core.xsd (Suite)

```

        maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <!-- A comma-separated list of transports that the virtual machine supports to provide
         feedback. -->
    <xs:attribute name="transport" type="xs:string"/>
    <xs:anyAttribute namespace="##any"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>

```

Schéma ovf-virtualhardware.xsc

EXEMPLE E-6 Schéma ovf-virtualhardware.xsc

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/envelope"
  xmlns:ovf="/var/opt/SUNWldom/envelope"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:vssd="/var/opt/SUNWldom/CIM_VirtualSystemSettingData"
  xmlns:rasd="/var/opt/SUNWldom/CIM_ResourceAllocationSettingData">

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <xs:include schemaLocation="ovf-section.xsd"/>

  <xs:import namespace="/var/opt/SUNWldom/CIM_VirtualSystemSettingData" schemaLocation="cim-vssd.xsd"/>
  <xs:import namespace="/var/opt/SUNWldom/CIM_ResourceAllocationSettingData"
    schemaLocation="cim-rasd.xsd"/>

  <!-- Specifies the virtual hardware for a virtual machine -->
  <xs:complexType name="VirtualHardwareSection_Type">
    <xs:complexContent>
      <xs:extension base="ovf:Section_Type">
        <xs:sequence>
          <xs:element name="System" type="vssd:CIM_VirtualSystemSettingData_Type" minOccurs="0"/>
          <xs:element name="Item" type="rasd:CIM_ResourceAllocationSettingData_Type"
            minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <!-- Specifies a section for resource constraints on a VirtualSystemCollection -->
  <xs:complexType name="ResourceAllocationSection_Type">
    <xs:complexContent>
      <xs:extension base="ovf:Section_Type">
        <xs:sequence>
          <xs:element name="Item" type="rasd:CIM_ResourceAllocationSettingData_Type"
            minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

EXEMPLE E-6 Schéma ovf-virtualhardware.xsc (Suite)

```

    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>

```

Schéma cim-rasd.xsd

EXEMPLE E-7 Schéma cim-rasd.xsd

```

<?xml version='1.0' encoding='utf-8'?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/CIM_ResourceAllocationSettingData"
  xmlns:class="/var/opt/SUNWldom/CIM_ResourceAllocationSettingData"
  xmlns:cim="/var/opt/SUNWldom/common"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="/var/opt/SUNWldom/common" schemaLocation="cim-common.xsd"/>

  <xs:element name="Caption" nillable="true" type="cim:cimString"/>

  <xs:element name="Description" nillable="true" type="cim:cimString"/>

  <xs:element name="InstanceId" nillable="true" type="cim:cimString"/>

  <xs:element name="ResourceType" nillable="true">
    <xs:complexType>
      <xs:simpleContent>
        <xs:restriction base="xs:anyType">
          <xs:simpleType>
            <xs:union>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:enumeration value="1"/> <!-- Other -->
                  <xs:enumeration value="2"/> <!-- Computer System -->
                  <xs:enumeration value="3"/> <!-- Processor-->
                  <xs:enumeration value="4"/> <!-- Memory-->
                  <xs:enumeration value="5"/> <!-- IDE Controller -->
                  <xs:enumeration value="6"/> <!-- Parallel SCSI HBA -->
                  <xs:enumeration value="7"/> <!-- FC HBA -->
                  <xs:enumeration value="8"/> <!-- iSCSI HBA -->
                  <xs:enumeration value="9"/> <!-- IB HCA -->
                  <xs:enumeration value="10"/> <!-- Ethernet Adapter -->
                  <xs:enumeration value="11"/> <!-- Other Network Adapter -->
                  <xs:enumeration value="12"/> <!-- I/O Slot -->
                  <xs:enumeration value="13"/> <!-- I/O Device -->
                  <xs:enumeration value="14"/> <!-- Floppy Drive -->
                  <xs:enumeration value="15"/> <!-- CD Drive -->
                  <xs:enumeration value="16"/> <!-- DVD drive -->
                  <xs:enumeration value="17"/> <!-- Disk Drive -->
                  <xs:enumeration value="18"/> <!-- Tape Drive -->
                  <xs:enumeration value="19"/> <!-- Storage Extent -->
                  <xs:enumeration value="20"/> <!-- Other storage device -->
                
```

EXEMPLE E-7 Schéma cim-rasd.xsd (Suite)

```

<xs:enumeration value="21"/> <!-- Serial port -->
<xs:enumeration value="22"/> <!-- Parallel port -->
<xs:enumeration value="23"/> <!-- USB Controller -->
<xs:enumeration value="24"/> <!-- Graphics controller -->
<xs:enumeration value="25"/> <!-- IEEE 1394 Controller -->
<xs:enumeration value="26"/> <!-- Partitionable Unit -->
<xs:enumeration value="27"/> <!-- Base Partitionable Unit -->
<xs:enumeration value="28"/> <!-- Power Supply -->
<xs:enumeration value="29"/> <!-- Cooling Device -->
<xs:enumeration value="29"/> <!-- Cooling Device -->
<xs:enumeration value="31"/> <!-- PS2 Controller -->
<xs:enumeration value="32"/> <!-- SIO Controller -->
<xs:enumeration value="33"/> <!-- Keyboard -->
<xs:enumeration value="34"/> <!-- Pointing Device -->
</xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:unsignedShort">
    <xs:minInclusive value="30"/>
    <xs:maxInclusive value="32769"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:unsignedShort">
    <xs:minInclusive value="32768"/>
    <xs:maxInclusive value="65535"/>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any" />
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>

<xs:element name="OtherResourceType" nillable="true" type="cim:cimString"/>
<xs:element name="ResourceSubType" nillable="true" type="cim:cimString"/>
<xs:element name="PoolID" nillable="true" type="cim:cimString"/>
<xs:element name="ConsumerVisibility" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="xs:anyType">
        <xs:simpleType>
          <xs:union>
            <xs:simpleType>
              <xs:restriction base="xs:unsignedShort">
                <xs:enumeration value="0"/>
                <xs:enumeration value="2"/>
                <xs:enumeration value="3"/>
                <xs:enumeration value="4"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```

EXEMPLE E-7 Schéma cim-rasd.xsd (Suite)

```

<xs:simpleType>
  <xs:restriction base="xs:unsignedShort">
    <xs:minInclusive value="5"/>
    <xs:maxInclusive value="32768"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:unsignedShort">
    <xs:minInclusive value="32767"/>
    <xs:maxInclusive value="65535"/>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>

<xs:element name="HostResource" nillable="true" type="xs:anyType"/>
<xs:element name="AllocationUnits" nillable="true" type="cim:cimString"/>
<xs:element name="VirtualQuantity" nillable="true" type="cim:cimUnsignedLong"/>
<xs:element name="Reservation" nillable="true" type="cim:cimUnsignedLong"/>
<xs:element name="Limit" nillable="true" type="cim:cimUnsignedLong"/>
<xs:element name="Weight" nillable="true" type="cim:cimUnsignedInt"/>
<xs:element name="AutomaticAllocation" nillable="true" type="cim:cimBoolean"/>
<xs:element name="AutomaticDeallocation" nillable="true" type="cim:cimBoolean"/>
<xs:element name="Parent" nillable="true" type="cim:cimString"/>
<xs:element name="Connection" nillable="true" type="cim:cimString"/>
<xs:element name="Address" nillable="true" type="cim:cimString"/>
<xs:element name="MappingBehavior" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="xs:anyType">
        <xs:simpleType>
          <xs:union>
            <xs:simpleType>
              <xs:restriction base="xs:unsignedShort">
                <xs:enumeration value="0"/>
                <xs:enumeration value="1"/>
                <xs:enumeration value="2"/>
                <xs:enumeration value="3"/>
                <xs:enumeration value="4"/>
              </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
              <xs:restriction base="xs:unsignedShort">
                <xs:minInclusive value="5"/>
                <xs:maxInclusive value="32768"/>
              </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
              <xs:restriction base="xs:unsignedShort">
                <xs:minInclusive value="32767"/>
                <xs:maxInclusive value="65535"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

```

EXEMPLE E-7 Schéma cim-rasd.xsd (Suite)

```

    </xs:restriction>
  </xs:simpleType>
</xs:union>
  </xs:simpleType>
  <xs:anyAttribute namespace="##any"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="AddressOnParent" nillable="true" type="cim:cimString"/>

<xs:element name="BusNumber" nillable="true" type="cim:cimUnsignedShort"/>

<xs:complexType name="CIM_ResourceAllocationSettingData_Type">
  <xs:sequence>
    <xs:element ref="class:Caption" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="class:Description" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="class:InstanceId" minOccurs="0"/>
    <xs:element ref="class:ResourceType" minOccurs="0"/>
    <xs:element ref="class:OtherResourceType" minOccurs="0"/>
    <xs:element ref="class:ResourceSubType" minOccurs="0"/>
    <xs:element ref="class:PoolID" minOccurs="0"/>
    <xs:element ref="class:ConsumerVisibility" minOccurs="0"/>
    <xs:element ref="class:HostResource" maxOccurs="unbounded" minOccurs="0"/>
    <xs:element ref="class:AllocationUnits" minOccurs="0"/>
    <xs:element ref="class:VirtualQuantity" minOccurs="0"/>
    <xs:element ref="class:Reservation" minOccurs="0"/>
    <xs:element ref="class:Limit" minOccurs="0"/>
    <xs:element ref="class:Weight" minOccurs="0"/>
    <xs:element ref="class:AutomaticAllocation" minOccurs="0"/>
    <xs:element ref="class:AutomaticDeallocation" minOccurs="0"/>
    <xs:element ref="class:Parent" minOccurs="0"/>
    <xs:element ref="class:Connection" maxOccurs="unbounded" minOccurs="0"/>
    <xs:element ref="class:Address" minOccurs="0"/>
    <xs:element ref="class:MappingBehavior" minOccurs="0"/>
    <xs:element ref="class:AddressOnParent" minOccurs="0"/>
    <xs:element ref="class:BusNumber" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any"/>
</xs:complexType>

<xs:element name="CIM_ResourceAllocationSettingData"
  type="class:CIM_ResourceAllocationSettingData_Type"/>
</xs:schema>

```

Schéma cim-vssd.xsd

EXEMPLE E-8 Schéma cim-vssd.xsd

```
<?xml version='1.0' encoding='utf-8'?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/CIM_VirtualSystemSettingData"
  xmlns:class="/var/opt/SUNWldom/CIM_VirtualSystemSettingData"
  xmlns:cim="/var/opt/SUNWldom/common"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import namespace="/var/opt/SUNWldom/common"
    schemaLocation="cim-common.xsd"/>

  <xs:element name="Caption" nillable="true" type="cim:cimString"/>

  <xs:element name="Description" nillable="true" type="cim:cimString"/>

  <xs:element name="InstanceId" nillable="true" type="cim:cimString"/>

  <xs:element name="VirtualSystemIdentifier" nillable="true" type="cim:cimString"/>

  <xs:element name="VirtualSystemType" nillable="true" type="cim:cimString"/>

  <xs:complexType name="CIM_VirtualSystemSettingData_Type">
    <xs:sequence>
      <xs:element ref="class:Caption" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="class:Description" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="class:InstanceId" minOccurs="0"/>
      <xs:element ref="class:VirtualSystemIdentifier" minOccurs="0"/>
      <xs:element ref="class:VirtualSystemType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any"/>
  </xs:complexType>

  <xs:element name="CIM_VirtualSystemSettingData" type="class:CIM_VirtualSystemSettingData_Type"/>
</xs:schema>
```

Schéma cim-common.xsd

EXEMPLE E-9 Schéma cim-common.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/common"
  xmlns:cim="/var/opt/SUNWldom/common"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!-- The following are runtime attribute definitions -->
  <xs:attribute name="Key" type="xs:boolean"/>

  <xs:attribute name="Version" type="xs:string"/>
```

EXEMPLE E-9 Schéma cim-common.xsd (Suite)

```

<!-- The following section defines the extended WS-CIM datatypes -->
<xs:complexType name="cimDateTime">
  <xs:choice>
    <xs:element name="CIM_DateTime" type="xs:string" nillable="true"/>
    <xs:element name="Interval" type="xs:duration"/>
    <xs:element name="Date" type="xs:date"/>
    <xs:element name="Time" type="xs:time"/>
    <xs:element name="Datetime" type="xs:dateTime"/>
  </xs:choice>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<xs:complexType name="cimUnsignedByte">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedByte">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimByte">
  <xs:simpleContent>
    <xs:extension base="xs:byte">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimUnsignedShort">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedShort">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimShort">
  <xs:simpleContent>
    <xs:extension base="xs:short">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimUnsignedInt">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedInt">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimInt">
  <xs:simpleContent>
    <xs:extension base="xs:int">

```

EXEMPLE E-9 Schéma cim-common.xsd (Suite)

```
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimUnsignedLong">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedLong">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimLong">
  <xs:simpleContent>
    <xs:extension base="xs:long">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimString">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimBoolean">
  <xs:simpleContent>
    <xs:extension base="xs:boolean">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimFloat">
  <xs:simpleContent>
    <xs:extension base="xs:float">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimDouble">
  <xs:simpleContent>
    <xs:extension base="xs:double">
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimChar16">
  <xs:simpleContent>
    <xs:restriction base="cim:cimString">
```

EXEMPLE E-9 Schéma cim-common.xsd (Suite)

```

        <xs:maxLength value="1"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:restriction>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimBase64Binary">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimHexBinary">
    <xs:simpleContent>
        <xs:extension base="xs:hexBinary">
            <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="cimReference">
    <xs:sequence>
        <xs:any namespace="##other" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<!-- The following datatypes are used exclusively to define metadata fragments -->
<xs:attribute name="qualifier" type="xs:boolean"/>

<xs:complexType name="qualifierString">
    <xs:simpleContent>
        <xs:extension base="cim:cimString">
            <xs:attribute ref="cim:qualifier" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="qualifierBoolean">
    <xs:simpleContent>
        <xs:extension base="cim:cimBoolean">
            <xs:attribute ref="cim:qualifier" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="qualifierUInt32">
    <xs:simpleContent>
        <xs:extension base="cim:cimUnsignedInt">
            <xs:attribute ref="cim:qualifier" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

```

EXEMPLE E-9 Schéma cim-common.xsd (Suite)

```

<xs:complexType name="qualifierSInt64">
  <xs:simpleContent>
    <xs:extension base="cim:cimLong">
      <xs:attribute ref="cim:qualifier" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<!--
<xs:complexType name="qualifierSArray">
  <xs:complexContent>
    <xs:extension base="cim:qualifierString"/>
  </xs:complexContent>
</xs:complexType>
-->
<!-- The following element is to be used only for defining metadata -->
<xs:element name="DefaultValue" type="xs:anySimpleType"/>
</xs:schema>

```

Schéma XML GenericProperty

Ce schéma est une extension du schéma OVF (Open Virtualization Format).

EXEMPLE E-10 Schéma XML GenericProperty

```

<?xml version='1.0' encoding='utf-8'?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/GenericProperty"
  xmlns:class="/var/opt/SUNWldom/GenericProperty"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="GenericProperty_Type" type="xs:string">
    <xs:attribute name="key" type="xs:string" use="required"/>

  </xs:complexType>
  <xs:element name="GenericProperty" type="class:GenericProperty_Type"/>

</xs:schema>

```

Binding_Type (Schéma XML)

Ce schéma est une extension du schéma OVF (Open Virtualization Format).

EXEMPLE E-11 Binding_Type (Schéma XML)

```

<?xml version='1.0' encoding='utf-8'?>
<xs:schema
  targetNamespace="/var/opt/SUNWldom/Binding"
  xmlns:class="/var/opt/SUNWldom/Binding"
  xmlns:rasd="/var/opt/SUNWldom/CIM_ResourceAllocationSettingData"

```

EXEMPLE E-11 Binding_Type (Schéma XML) (Suite)

```
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import namespace="/var/opt/SUNWldom/CIM_ResourceAllocationSettingData"
  schemaLocation="cim-rasd.xsd"/>
<xs:complexType name="Binding_Type">
  <xs:sequence>
    <xs:element name="Item"
      type="rasd:CIM_ResourceAllocationSettingData_Type"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```


Glossaire

Cette liste définit la terminologie, les abréviations et les acronymes de la documentation de Oracle VM Server for SPARC.

A

API	Interface de programmation d'application
auditreduce	Fusionne et sélectionne des enregistrements d'audit dans les fichiers de piste d'audit (voir la page de manuel auditreduce(1M)).
contrôle	Utilisation du contrôle du SE Oracle Solaris pour identifier la source des modifications de sécurité
autorisation	Définition d'une autorisation à l'aide du RBAC du SE Oracle Solaris

B

bge	Pilote Broadcast Gigabit Ethernet sur périphériques BCM57xx
BSM	Module de sécurité de base
bsmconv	Active le BSM (voir la page de manuel bsmconv(1M)).
bsmunconv	Désactive le BSM (voir la page de manuel bsmunconv(1M)).

C

CD	Disque compact
CLI	Interface de ligne de commande
conformité	Détermine si la configuration d'un système est en conformité avec un profil de sécurité prédéfini

configuration	Nom de la configuration de domaine logique enregistrée sur le processeur de service
CMT	Multithreading de puce
contraintes	Les contraintes du gestionnaire de domaines logiques sont une ou plusieurs ressources que vous voulez assigner à un domaine particulier. Soit vous recevez toutes les ressources dont vous avez demandé l'ajout au domaine, soit vous ne recevez aucune d'entre elles, en fonction des ressources disponibles.
domaine de contrôle	Domaine qui crée et gère les autres domaines logiques et les services
CPU	Unité de calcul centrale
CWQ	CWQ (Control Word Queue - File d'attente de mots de contrôle) ; unité cryptographique des plates-formes Sun UltraSPARC T2 d'Oracle

D

DHCP	DHCP (Dynamic Host Configuration Protocol) Le DMA (Direct Memory Access - Accès direct à la mémoire) est la capacité à transférer directement des données entre la mémoire et un périphérique (par exemple, une carte réseau) sans impliquer le CPU.
DMP	Multiacheminement dynamique (Veritas)
gestionnaire de domaines logiques	CLI permettant de créer et de gérer des domaines logiques et d'allouer les ressources aux domaines
DPS	Date plane software (Logiciel de plan de données)
DR	Reconfiguration dynamique
drd	Démon de reconfiguration dynamique du SE Oracle Solaris 10 pour gestionnaire de domaines logiques (voir la page de manuel drd(1M)).
DS	Module de services du domaine (SE Oracle Solaris 10)
DVD	Disque versatile numérique

E

EFI	Interface de microprogramme extensible
ETM	Module de gestion des tables de codage (SE Oracle Solaris)

F

FC_AL	Boucle arbitrée de canal de fibre
FMA	Architecture de gestion des pannes (Fault Management Architecture)
fmd	Démon du gestionnaire de défauts du SE Oracle Solaris 10 (voir la page de manuel fmd(1M)).
format	Utilitaire de partitionnement et de maintenance de disque (voir la page de manuel format(1M)).
fmthard	Remplit les étiquettes sur les disques durs (voir la page de manuel fmthard(1M)).
FTP	File Transfer Protocol

G

Gb	Gigabit
domaine invité	Utilise les services des domaines d'E/S et de service et est géré par le domaine de contrôle.
GLDv3	Pilote LAN générique version 3.

H

Sécurisation	Modification de la configuration du SE Oracle Solaris pour améliorer la sécurité.
HDD	Unité de disque dur
hyperviseur	Couche de microprogramme interposé entre le système d'exploitation et la couche matérielle

I

Domaine d'E/S	Domaine ayant la propriété directe et l'accès direct aux périphériques d'E/S physiques et qui partage ces périphériques avec d'autres domaines logiques sous la forme de périphériques virtuels
IB	InfiniBand
IDE	Integrated Drive Electronics (Électronique d'unité de disque intégrée)
IDR	Interim Diagnostic Release (Version de diagnostic intermédiaire)
ILOM	Integrated Lights Out Manager

io	Périphériques d'E/S, notamment les disques internes et les contrôleurs PCIe ainsi que leurs adaptateurs périphériques connectés
ioctl	Appel de contrôle d'entrée/sortie
IP	Internet Protocol
IPMP	Internet Protocol Network Multipathing
ISO	Organisation internationale de normalisation

K

kaio	Entrée/sortie de noyau asynchrone
Ko	Kilooctet
KU	Mise à jour du noyau

L

LAN	Réseau local
LDAP	Protocole LDAP
LDC	Canal de domaine logique
ldm	Utilitaire du gestionnaire de domaines logiques (voir la page de manuel ldm(1M)).
ldmd	Démon du gestionnaire de domaines logiques
lofi	Fichier loopback
domaine logique	Machine virtuelle comprenant un groupement logique discret de ressources, qui a son propre système d'exploitation et une identité au sein d'un système informatique unique
LUN	Numéro d'unité logique

M

MAC	Adresse de contrôle d'accès au média, que les domaines logiques peuvent assigner automatiquement ou que vous pouvez assigner manuellement
MAU	Unité arithmétique modulaire

Mo	Méga-octet
MD	Description de la machine dans la base de données du serveur
memmemory	Unité de mémoire - Taille par défaut en octets ou indiquée en giga-octets (G), kilooctets (K) ou méga-octets (M). Mémoire virtualisée du serveur qui peut être allouée aux domaines invités.
metadb	Créer et supprime les répliques de la base de données d'état des métapériphériques Solaris Volume Manager (voir la page de manuel metadb(1M)).
metaset	Configure les ensembles de disques (voir la page de manuel metaset(1M)).
mhd	Opérations de contrôle de disque multihôte (voir la page de manuel mhd(7i)).
MIB	Management Information Base
réduction	Installation du nombre minimum de packages nécessaires du SE Oracle Solaris principal
MMF	Fibre multimode
MMU	Unité de gestion de la mémoire
mpgroup	Nom du groupe multivoie pour le basculement du disque virtuel
mtu	Unité de transmission maximale

N

NAT	Network Address Translation
ndpsldcc	Client de canal de domaine logique DPS Netra <i>Voir aussi vdpcc.</i>
ndpsldcs	Service de canal de domaine logique DPS Netra. <i>Voir aussi vdpcc.</i>
NFS	Network File System
NIS	Network Information Services (Services d'information réseau)
NIU	Unité d'interface réseau (serveurs Sun SPARC Enterprise T5120 et T5220 d'Oracle)
NTS	Serveur de terminal réseau
NVRAM	Mémoire à accès aléatoire non volatile
nxge	Pilote pour un adaptateur Ethernet 10 Gb NIU

O

SE	Systeme d'exploitation
OVF	Format de virtualisation ouvert

P

P2V	Outil de conversion physique-à-virtuel des domaines logiques
PA	Adresse physique
PCI	Bus d'interconnexion de composants périphériques
PCIe	Bus PCI EXPRESS
PCI-X	Bus étendu PCI
pcpu	CPU physique
physio	Entrée/sortie physique
PICL	Informations de plate-forme et bibliothèque de contrôle
picld	Démon PICL (voir la page de manuel picld(1M)).
PM	Gestion de l'alimentation de la mémoire et des CPU virtuels
praudit	Imprime le contenu d'un fichier de piste d'audit (voir la page de manuel praudit(1M)).
PRI	Priorité

R

RA	Adresse réelle
RAID	Redundant Array of Inexpensive Disks
RBAC	Contrôle d'accès basé sur les rôles
RPC	Appel de procédure à distance

S

SASL	Authentification simple et couche de sécurité
SAX	API simple pour l'analyseur syntaxique XML, qui traverse un document XML. L'analyseur syntaxique SAX est basé sur les événements et utilisé principalement pour la transmission de données en continu.
contrôleur système (SC)	Voir également processeur de services
SCSI	Small Computer System Interface
domaine de service	Domaine logique qui fournit des périphériques, notamment des commutateurs virtuels, des connecteurs de consoles virtuelles et des serveurs de disques virtuels aux autres domaines logiques.
SMA	Agent de gestion du système
SMF	Utilitaire de gestion des services
SNMP	Protocole SNMP
processeur de service (SP)	SP, également appelé contrôleur de système (SC), surveille et exécute la machine physique.
SSH	Shell sécurisé
ssh	Commande Shell sécurisé (voir la page de manuel ssh(1)).
sshd	Démon de Shell sécurisé (voir la page de manuel sshd(1M)).
SunVTS	Sun Validation Test Suite
svcadm	Manipule les instances de service (voir la page de manuel svcadm(1M)).

T

TCP	Protocole TCP
TLS	Transport Layer Security

U

UDP	Protocole UDP (User Datagram Protocol)
UFS	Système de fichiers UNIX

unicast	Communication réseau ayant lieu entre un émetteur unique et un récepteur unique
USB	Universal Serial Bus
usci	Interface de commande SCSI utilisateur (voir la page de manuel uscsi(7I)).
UTP	Paire torsadée non blindée

V

var	Variable
VBSC	Contrôleur système de lame virtuelle
vcc, vconscn	Service du concentrateur de console virtuelle avec une plage de ports spécifique pour assigner des domaines invités
vcons, vconsole	Console virtuelle pour accéder aux messages au niveau du système. Une connexion est établie en connectant le service vconscn dans le domaine de contrôle à un port spécifique.
vcpu	Unité de calcul centrale virtuelle Chaque cœur dans un serveur est représenté par un CPU virtuel. Par exemple, un serveur Sun Fire T2000 8 cœurs d'Oracle a 32 CPU virtuels qui peuvent être alloués parmi les domaines logiques.
vdc	Client de disque virtuel
vdisk	Un disque virtuel est un périphérique en mode bloc générique associé à différents types de périphériques physiques, de volumes ou de fichiers.
vdppc	Client de canal de plan de données virtuelles dans un environnement DPS Netra
vdpcs	Client de service de plan de données virtuelles dans un environnement DPS Netra
vds, vdiskserver	Le serveur de disque virtuel vous permet d'importer des disques virtuels dans un domaine logique
vdsdev, vdiskserverdevice	La périphérique de serveur de disque virtuel est exporté par le serveur de disque virtuel. Le périphérique peut être un disque entier, un segment sur le disque, un fichier ou un volume de disque.
VLAN	Réseau local virtuel
vldc	Service de canal de domaine logique virtuel
vldcc	Client de canal de domaine logique virtuel
vnet	Le périphérique de réseau virtuel implémente un périphérique Ethernet virtuel et communique avec les autres périphériques vnet du système à l'aide du commutateur réseau virtuel (<code>vswit ch</code>)
vntsd	Démon du serveur de terminal réseau virtuel du SE Oracle Solaris 10 pour les consoles des domaines logiques (voir la page de manuel vntsd(1M)).
volfs	Système de fichiers de gestion du volume (voir la page de manuel volfs(7FS)).

vsw, vswitch	Commutateur de réseau virtuel qui connecte les périphériques du réseau virtuel au réseau externe et commute également les paquets entre eux
VTOC	Table des matières du volume
VxDMP	Mutiacheminement dynamique Veritas
VxVM	Veritas Volume Manager

W

WAN	Réseau étendu
------------	---------------

X

XFP	eXtreme Fast Path
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

Z

ZFS	Système de fichiers Zettabyte (SE Oracle Solaris 10)
zpool	Pool de stockage ZFS (voir la page de manuel zpool(1M)).
ZVOL	Pilote d'émulation de volume ZFS

Index

A

Assignation

- Bus PCIe à un domaine d'E/S, 66–71
- Périphérique d'extrémité à un domaine d'E/S, 71–81

Autorisation

- ldmSous-commandes, 41
- Lecture, 41
- Lecture et écriture, 41
- Niveaux, 41

B

bus PCI EXPRESS (PCIe), 65–66

C

- Cadres Jumbo, Configuration, 144–147
- canal de domaine logique (LDC), 19
- cancel-operation reconf Sous-commande, 164
- CLI, *Voir* Interface de ligne de commande
- commandes
 - ldm(1M), 21
- Commandes
 - ldmconfig(1M), 23, 227, 229
- commandes
 - ldmp2v(1M), 216
- Configuration
 - Cadres Jumbo, 144–147
- configuration, sélection pour le démarrage, 22

Configuration

- Stockage sur le processeur de service, 164
- Contrôleur système, *Voir* Processeur de service (SP)
- Création d'un domaine d'E/S, Bus PCIe entier, 67

D

- Définition, limite de puissance, 178
- Démon de reconfiguration dynamique (drd), 164
- démon du serveur de terminal réseau virtuel (vntsd), 22
- Démons
 - drd, 164
 - ldmd, 21
- démons, vntsd, 22
- Désactivation des cœurs de CPU, 178
- domaine d'E/S, 65–66
- Domaine d'E/S, 66–71, 71–81
 - Assignation d'un bus PCIe, 66–71
 - Assignation d'un périphérique d'extrémité, 71–81
- domaine d'E/S
 - bus PCI EXPRESS (PCIe), 65–66
- Domaine d'E/S
 - Création, 67
- domaine d'E/S
 - limitations de migration, 66
- Domaine de contrôle, 20
- Domaine de service, 20, 21
- Domaine invité, 20
- Domaine racine, 20

- Domaines
 - Service, 21
 - Types, 20
 - Domaines logiques
 - Définition, 17
 - Rôles, 20
 - DR, *Voir* Reconfiguration dynamique
- E**
- E/S directe (DIO), Planification, 74
- G**
- Gestion de l'alimentation (PM), 178
 - Gestion de l'alimentation (PM) de la mémoire, 178
 - gestionnaire de domaines logiques, 18, 19
 - Démon (ldmd), 21
 - Mécanisme de recherche, 233
 - Schémas XML utilisés avec, 261
 - utilisation du schéma XML, 237
- H**
- Hyperviseur, définition, 17
- I**
- Interface de ligne de commande, 21
 - Interface graphique de l'assistant de configuration, 227
 - IPMP basé sur liaison, utilisation, 130–133
- L**
- LDC, *Voir* canal de domaine logique
 - ldm Sous-commandes
 - cancel-operation reconf, 164
 - ls-dom, 164
 - ldm(1M) page de manuel, 21
 - ldm(1M)commande, 21
 - ldmconfig(1M)Commande, 23, 227, 229
 - ldmd, Démon du gestionnaire de domaines logiques, 21
 - ldmp2v(1M) commande, 216
 - ldmSous-commandes, Autorisations utilisateur, 41
 - Lecture, Autorisation, 41
 - Lecture et écriture, Autorisation, 41
 - limitations de migration, domaine d'E/S, 66
 - limite de puissance, 178
 - ls-dom Sous-commande, 164
- M**
- Machine virtuelle, 19
 - migration, non interactive, 160
 - migration de domaine, non interactive, 160
 - migration de domaine non interactive, 160
 - Multivoie, Disque virtuel, 96
 - Multivoie de disque virtuel, 96
- P**
- packages, SUNWldm, 21
 - Périphériques physiques, 20, 21
 - périphériques virtuels
 - client de disque virtuel (vdc), 22
 - commutateur virtuel (vsw), 21
 - concentrateur de console virtuelle (vcc), 22
 - Périphériques virtuels, E/S, 21
 - périphériques virtuels
 - réseau virtuel (vnet), 21
 - service de disque virtuel (vds), 22
 - Planification
 - E/S directe (DIO), 74
 - Plates-formes
 - Serveur SPARC T3, 21
 - Serveur UltraSPARC T2 Plus, 21
 - primary Domaine, Redémarrage, 75–76
 - primaryDomaine, 20
 - Processeur de service (SP)
 - Ssurveillance et exécution des machines virtuelles, 19
 - Stockage des configurations, 164

R

Reconfiguration dynamique (DR), 163

Reconfiguration dynamique de la mémoire (DR), 169

Reconfiguration retardée, 164

Redémarrage du domaine primary, 75–76

Ressources

Voir aussi Périphériques virtuels

Définition, 19

rôles, Domaines logiques, 20

S

Saut du cycle d'horloge du CPU, 178

Schéma XML

utilisation du gestionnaire de domaines

logiques, 237

Utilisé avec le gestionnaire de domaines

logiques, 261

Serveur SPARC T3, 21

Serveur UltraSPARC T2 Plus, 21

SUNWldm package, 21

U

utilisation de l'IPMP basé sur liaison, 130–133

