

**Administration de la gestion des
ressources dans Oracle® Solaris 11.2**

Brouillon 2015-01-21-10:40:25+01:00

Copyright © 2004, 2014, 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 expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, 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 livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 un risque de 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 des applications dangereuses.

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.

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. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

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.

Cette documentation est fournie en version pré-production et uniquement à des fins de démonstration et d'usage à titre préliminaire de la version finale. Celle-ci n'est pas toujours spécifique du matériel informatique sur lequel vous utilisez ce logiciel. Oracle Corporation et ses affiliés déclinent expressément toute responsabilité ou garantie expresse quant au contenu de cette documentation. Oracle Corporation et ses affiliés ne sauraient en aucun cas être tenus pour responsable des pertes subies, des coûts occasionnés ou des dommages causés par l'utilisation de cette documentation.

Les informations contenues dans ce document sont fournies à titre informatif uniquement et doivent être prises en compte en votre qualité de membre du customer advisory board ou conformément à votre contrat d'essai de version bêta uniquement. Ce document ne constitue en aucun cas un engagement à fournir des composants, du code ou des fonctionnalités et ne doit pas être retenu comme base d'une quelconque décision d'achat. Le développement, la commercialisation et la mise à disposition des fonctions ou fonctionnalités décrites restent à la seule discrétion d'Oracle.

Ce document contient des informations qui sont la propriété exclusive d'Oracle, qu'il s'agisse de la version électronique ou imprimée. Votre accès à ce contenu confidentiel et son utilisation sont soumis aux termes de votre Contrat de Licence et de Services Oracle en vigueur que vous avez signé et que vous vous engagez à respecter. Ce document et son contenu ne peuvent en aucun cas être communiqués, copiés, reproduits ou distribués à une personne extérieure à Oracle sans le consentement écrit d'Oracle. Ce document ne fait pas partie de votre accord de licence. Par ailleurs, il ne peut être intégré à aucun accord contractuel passé avec Oracle ou ses filiales.

Table des matières

Utilisation de la présente documentation	11
1 Introduction à la gestion des ressources	13
Présentation de la gestion des ressources	13
Classifications des ressources	14
Mécanismes de contrôle de la gestion des ressources	15
Configuration de la gestion des ressources	16
Interaction avec les zones non globales	17
Intérêt de la gestion des ressources	17
Consolidation serveur	17
Prise en charge d'une population importante et variée d'utilisateurs	18
Configuration de la gestion des ressources (liste des tâches)	18
2 A propos des projets et tâches	21
Utilitaires de projet et de tâche	21
Identificateurs de projet	22
Détermination du projet par défaut des utilisateurs	22
Définition des attributs utilisateur à l'aide des commandes useradd et usermod	23
Base de données project	23
Sous-système PAM	24
Configuration des services de noms	24
Format de fichier /etc/project local	25
Configuration de projet pour le système d'information réseau NIS	27
Configuration de projet pour le service d'annuaire LDAP	27
Identificateurs des tâches	27
Commandes utilisées avec les projets et les tâches	29
3 Administration des projets et des tâches	31
Administration des projets et des tâches (liste des tâches)	31

Exemples de commandes et d'options de commande	32
Options de commande utilisées avec les projets et les tâches	32
Utilisation des commandes cron et su avec les projets et les tâches	34
Administration des projets	35
▼ Définition d'un projet et affichage du projet actuel	35
▼ Suppression d'un projet du fichier /etc/project	37
Validation du contenu du fichier /etc/project	38
Obtention des informations d'appartenance au projet	39
▼ Création d'une tâche	39
▼ Transfert d'un processus en cours vers une nouvelle tâche	39
Modification et validation des attributs de projet	40
▼ Ajout d'attributs et de valeurs d'attribut à des projets	40
▼ Suppression des valeurs d'attribut des projets	41
▼ Suppression d'un attribut de contrôle de ressource d'un projet	41
▼ Remplacement des attributs et des valeurs d'attribut des projets	42
▼ Suppression des valeurs existantes pour un attribut de contrôle de ressource	42
4 A propos de la comptabilisation étendue	43
Introduction à la comptabilisation étendue	43
Mode de fonctionnement de la comptabilisation étendue	44
Format extensible	46
Enregistrements et formats exact	46
Utilisation de la comptabilisation étendue sur un système Oracle Solaris comportant des zones installées	46
Configuration de la comptabilisation étendue	47
Démarrage et activation durable de la comptabilisation étendue	47
Enregistrements	48
Commandes s'appliquant à la comptabilisation étendue	48
Interface Perl pour libexacct	48
5 Administration des tâches de comptabilisation étendue	51
Administration de l'utilitaire de comptabilisation étendue (liste des tâches)	51
Utilisation de la fonctionnalité de comptabilisation étendue	52
▼ Activation de la comptabilisation étendue des flux, processus, tâches et composants réseau	52
Affichage de l'état de la comptabilisation étendue	53
Affichage des ressources de comptabilisation disponibles	53

▼ Désactivation de la comptabilisation des processus, des tâches, des flux et de gestion de réseau	54
Utilisation de l'interface Perl pour accéder à libexacct	55
Affichage récursif du contenu d'un objet exacct	55
Création et écriture d'un enregistrement de groupe dans un fichier	57
Impression du contenu d'un fichier exacct	57
Exemple de sortie de Sun::Solaris::Exacct::Object->dump()	58
6 A propos des contrôles de ressources	59
Concepts de base sur les contrôles de ressources	59
Limites d'utilisation des ressources et contrôles de ressources	60
Communication interprocessus et contrôles de ressources	60
Mécanismes de contrainte par contrôle des ressources	61
Mécanismes d'attribut d'un projet	61
Configuration des contrôles de ressources et des attributs	61
Contrôles de ressources disponibles	62
Contrôles des ressources à l'échelle d'une zone	65
Prise en charge des unités	66
Valeurs des contrôles de ressources et niveaux de privilège	67
Actions globales et locales applicables aux valeurs de contrôle de ressource	68
Indicateurs et propriétés des contrôles de ressources	70
Mise en oeuvre du contrôle de ressources	72
Surveillance globale des événements de contrôle de ressource	73
Application des contrôles de ressources	73
Mise à jour temporaire des valeurs de contrôle de ressource sur un système en cours d'exécution	73
Mise à jour de l'état de la consignment	73
Mise à jour des contrôles de ressources	74
Commandes utilisées avec les contrôles de ressources	74
7 Administration des contrôles des ressources (tâches)	75
Administration des contrôles des ressources (liste des tâches)	75
Définition des contrôles des ressources	76
▼ Définition du nombre maximum de processus légers (LWP) pour chaque tâche d'un projet	76
▼ Définition de plusieurs contrôles pour un projet	77
Utilisation de la commande prctl	78
▼ Affichage des valeurs de contrôle des ressources par défaut à l'aide de la commande prctl	79

▼ Affichage des informations relatives à un contrôle de ressource précis à l'aide de la commande <code>prctl</code>	81
▼ Modification temporaire d'une valeur à l'aide de la commande <code>prctl</code>	81
▼ Réduction de la valeur de contrôle des ressources à l'aide de la commande <code>prctl</code>	82
▼ Affichage, remplacement et vérification de la valeur d'un contrôle dans un projet à l'aide de la commande <code>prctl</code>	83
Utilisation de la commande <code>rctladm</code>	83
Utilisation de la commande <code>rctladm</code>	83
Utilisation de la commande <code>ipcs</code>	84
Utilisation de la commande <code>ipcs</code>	84
Avertissements relatifs à la capacité	85
▼ Détermination de la capacité CPU allouée à un serveur Web	85
8 A propos de l'ordonnanceur de partage équitable	87
Introduction à l'ordonnanceur FSS	87
Définition d'une part de CPU	88
Parts de la CPU et état des processus	89
Différence entre allocation des parts de CPU et utilisation de la CPU	89
Exemples de parts de CPU	89
Exemple 1 : deux processus tirant parti de la CPU dans chaque projet	90
Exemple 2 : aucune compétition entre les projets	91
Exemple 3 : un projet dans l'incapacité de s'exécuter	92
Configuration de l'ordonnanceur FSS	94
Projets et utilisateurs	94
Configuration des parts de CPU	94
Ordonnanceur FSS et jeux de processeurs	95
Exemples d'utilisation des jeux de processeurs avec l'ordonnanceur FSS	96
Utilisation de l'ordonnanceur FSS avec d'autres classes de programmation	98
Définition de la classe de programmation pour le système	99
Classe de programmation dans un système doté de zones	99
Commandes utilisées avec l'ordonnanceur FSS	99
9 Administration de l'ordonnanceur FSS (tâches)	101
Administration de l'ordonnanceur FSS (liste des tâches)	101
Surveillance de l'ordonnanceur FSS	102
▼ Surveillance de l'utilisation de la CPU par projet	102
▼ Surveillance de l'utilisation de la CPU par projet dans les jeux de processeurs	102

Configuration de l'ordonnanceur FSS	103
Liste des classes de programmation sur le système	103
▼ Sélection de l'ordonnanceur FSS comme classe de programmation par défaut	103
▼ Transfert manuel de processus de la classe TS vers la classe FSS	104
▼ Transfert manuel de toutes les classes de processus vers la classe FSS	104
▼ Transfert manuel des processus d'un projet vers la classe FSS	105
Ajustement des paramètres de l'ordonnanceur	105
10 A propos du contrôle de la mémoire physique à l'aide du démon de limitation des ressources	107
Introduction au démon de limitation des ressources	107
Principe de fonctionnement de la limitation des ressources	108
Attribut permettant de limiter l'utilisation de la mémoire physique pour les projets	109
Configuration de la commande rcapd	109
Utilisation du démon de limitation des ressources sur un système doté de zones	110
Seuil d'allocation restrictive de la mémoire	110
Détermination des valeurs de seuil	111
Intervalles des opérations rcapd	112
Surveillance des ressources à l'aide de rcapstat	114
Commandes utilisées avec rcapd	115
11 Administration du démon de limitation des ressources (tâches)	117
Définition de la limite de taille résidente définie	117
▼ Ajout d'un attribut rcap.max-rss pour un projet	117
▼ Utilisation de la commande projmod pour ajouter un attribut rcap.max-rss pour un projet	118
Configuration et utilisation du démon de limitation des ressources (liste des tâches)	118
Administration du démon de limitation des ressources avec rcapadm	119
▼ Définition du seuil d'allocation restrictive de la mémoire	119
▼ Définition des intervalles de fonctionnement	120
▼ Activation de la limitation des ressources	120
▼ Désactivation de la limitation des ressources	120
▼ Spécification d'une limitation temporaire de ressources pour une zone	121
Etablissement de rapports à l'aide de la commande rcapstat	121
Création d'un rapport sur la limitation des ressources et sur le projet	122
Surveillance de la taille résidente définie d'un projet	122

Détermination de la taille de la charge de travail définie d'un projet	123
Création d'un rapport sur l'utilisation de la mémoire et le seuil d'allocation restrictive	124
12 A propos des pools de ressources	127
Introduction aux pools de ressources	128
Introduction aux pools de ressources dynamiques	129
A propos de l'activation et de la désactivation des pools de ressources et des pools de ressources dynamiques	129
Pools de ressources utilisés dans les zones	130
Intérêt des pools	130
Structure des pools de ressources	131
Contenu du fichier /etc/pooladm.conf	132
Propriétés des pools	133
Implémentation des pools sur un système	133
Attribut project.pool	134
SPARC: opérations de reconfiguration dynamique et pools de ressources	134
Création de configurations de pools	135
Affectation spécifique de CPU, de noyaux et de sockets	136
Manipulation directe de la configuration dynamique	136
Présentation de poold	136
Gestion des pools de ressources dynamiques	137
Contraintes et objectifs de configuration	137
Contraintes de configuration	138
Objectifs de configuration	139
Propriétés poold	142
Fonctionnalités poold configurables	142
Intervalle de surveillance poold	143
Informations de journalisation poold	143
Emplacement de journalisation	145
Gestion du journal avec logadm	145
Mode de fonctionnement de l'allocation dynamique des ressources	146
A propos des ressources disponibles	146
Détermination des ressources disponibles	146
Identification d'un manque de ressources	147
Détermination de l'utilisation des ressources	147
Identification des violations de contrôle	147
Détermination de l'action corrective appropriée	148

Utilisation de poolstat pour surveiller l'utilitaire des pools et l'utilisation des ressources	148
Sortie poolstat	149
Réglage des intervalles d'exécution des opérations poolstat	150
Commandes utilisées avec l'utilitaire des pools de ressources	150
13 Création et administration des pools de ressources (tâches)	153
Administration des pools de ressources (liste des tâches)	153
Activation et désactivation de l'utilitaire Pools	154
▼ Activation du service de pools de ressources à l'aide de svcadm	155
▼ Désactivation du service de pools de ressources à l'aide de svcadm	155
▼ Activation du service de pools de ressources dynamiques à l'aide de svcadm	155
▼ Désactivation du service de pools de ressources dynamiques à l'aide de svcadm	158
▼ Activation des pools de ressources à l'aide de pooladm	158
▼ Désactivation des pools de ressources à l'aide de pooladm	158
Affectation de CPU spécifique	158
Configuration des pools	159
▼ Création d'une configuration statique	159
▼ Modification d'une configuration	161
▼ Association d'un pool avec une classe de programmation	163
▼ Définition des contraintes de configuration	165
▼ Etablissement des objectifs de configuration	165
▼ Définition du niveau de consignation poold	167
▼ Utilisation des fichiers de commandes avec poolcfg	168
Transfert de ressources	169
▼ Transfert de CPU entre les jeux de processeurs	169
Activation et suppression des configurations de pools	169
▼ Activation d'une configuration de pools	170
▼ Test d'une configuration avant sa validation	170
▼ Suppression d'une configuration de pools	170
Définition des attributs des pools et liaison à un pool	171
▼ Liaison des processus à un pool	171
▼ Liaison de tâches ou de projets à un pool	172
▼ Définition de l'attribut project.pool pour un projet	172
▼ Liaison d'un processus à un autre pool grâce aux attributs project	172
Création d'un état statistique pour les ressources liées au pool à l'aide de poolstat	173

Affichage de la sortie poolstat par défaut	173
Création de plusieurs rapports à intervalles spécifiques	173
Création d'un état statistique sur l'ensemble de ressources	174
14 Exemple de configuration de la gestion des ressources	175
Configuration à consolider	175
Configuration de la consolidation	176
Création de la configuration	176
Visualisation de la configuration	177
Index	183

Utilisation de la présente documentation

- **Présentation** – Décrit comment écrire des applications qui partitionnent et gèrent les ressources du système, telles que des ensembles de processeurs et des classes de programmation.
- **Public** – Programmeurs connaissant les interfaces de système d'exploitation.
- **Connaissances requises** – Maîtrise de C et des interfaces de système Solaris (ou de type Unix).

Bibliothèque de documentation produit

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse : <http://www.oracle.com/pls/topic/lookup?ctx=E56338>.

Accès aux services de support Oracle

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Commentaires

Faites part de vos commentaires sur cette documentation à la page suivante : <http://www.oracle.com/goto/docfeedback>.

BETA
DRAFT

◆ ◆ ◆ C H A P I T R E 1

Introduction à la gestion des ressources

La fonction de gestion des ressources d'Oracle Solaris permet de contrôler la façon dont les applications utilisent les ressources système disponibles. Vous pouvez :

- Allouer des ressources informatiques (temps d'occupation du processeur, par exemple)
- Surveiller les allocations, puis les ajuster dans la mesure de vos besoins
- Générer des données de comptabilisation étendue à des fins d'analyse, de facturation et de planification de la capacité

Ce chapitre se compose des sections suivantes :

- [“Présentation de la gestion des ressources” à la page 13](#)
- [“Intérêt de la gestion des ressources” à la page 17](#)
- [“Configuration de la gestion des ressources \(liste des tâches\)” à la page 18](#)

Présentation de la gestion des ressources

Les environnements informatiques modernes se doivent de répondre avec un maximum de souplesse aux charges de travail variables qui résultent des différentes applications sur un système. Une *charge de travail* correspond à la somme de tous les processus mis en jeu par une application ou un groupe d'applications. En l'absence de fonctionnalités de gestion des ressources, le système d'exploitation Oracle Solaris répond aux charges de travail en adaptant de façon dynamique son activité aux nouvelles demandes des applications. Ce comportement par défaut implique généralement un accès équitable aux ressources pour toutes les activités du système. Les fonctions de gestion des ressources permettent d'apporter une réponse personnelle à chaque charge de travail. Vous pouvez :

- Restreindre l'accès à une ressource spécifique
- Accorder de préférence certaines ressources aux charges de travail
- Isoler les charges de travail les unes des autres

La *gestion des ressources* est l'ensemble des moyens permettant d'optimiser les performances des charges de travail et de surveiller la façon dont les ressources sont exploitées.. Elle est mise en oeuvre par le biais de divers algorithmes. Ces algorithmes traitent les séries de demandes de capacité envoyées par une application au cours de son exécution.

Les utilitaires de gestion des ressources permettent d'influer sur le comportement par défaut du système d'exploitation en matière de charges de travail. Le terme *comportement* représente l'ensemble des décisions prises par les algorithmes du système d'exploitation pour faire face aux demandes de ressources émanant d'une application. Les utilitaires de gestion des ressources permettent de :

- Refuser des ressources à une application ou d'accorder de préférence à une application un nombre plus important d'allocations que cela est normalement autorisé
- Traiter certaines allocations de façon collective au lieu de faire appel à des mécanismes isolés

La mise en oeuvre d'une configuration système fonctionnant sur le principe de gestion des ressources offre plusieurs avantages. Vous pouvez :

- Empêcher une application de consommer indifféremment des ressources
- Changer la priorité d'une application en fonction d'événements externes
- Répartir de façon équilibrée les ressources allouées à un ensemble d'applications tout en veillant à optimiser l'utilisation du système

Pour planifier une configuration de ce type, il est nécessaire de prendre en compte plusieurs facteurs clés :

- Identifier les charges de travail en concurrence sur le système (c'est-à-dire celles qui se disputent les mêmes ressources)
- Distinguer les charges de travail qui ne sont pas en conflit de celles pour lesquelles le niveau de performance requis compromet les charges de travail principales

Après avoir identifié les charges de travail qui coopèrent et celles qui sont en conflit, vous pouvez créer une configuration des ressources qui présente le meilleur compromis entre les objectifs de service de l'entreprise et les limites de capacité du système.

Dans la pratique, la gestion des ressources du système Oracle Solaris est fondée sur des mécanismes de contrôle, des mécanismes de notification et des mécanismes de surveillance. Il a fallu, dans la plupart des cas, apporter des améliorations aux mécanismes existants, notamment au système de fichiers [proc\(4\)](#) aux jeux de processeurs et aux classes de programmation. Dans les autres cas, les mécanismes sont propres à la gestion des ressources. Vous trouverez une description détaillée à ce sujet dans les chapitres suivants.

Classifications des ressources

Ce terme désigne un aspect du système informatique pouvant être manipulé afin de modifier le comportement d'une application. Autrement dit, une ressource est une capacité demandée de façon implicite ou explicite par une application. Si la capacité est refusée ou fait l'objet d'une restriction, l'exécution d'une application robuste prendra plus de temps.

La classification des ressources, à l'inverse de l'identification des ressources, peut obéir à certaines règles : implicites ou explicites, dépendantes ou non du facteur temps (temps d'utilisation de la CPU par opposition aux parts de CPU allouées), etc.

En règle générale, la gestion des ressources basée sur l'ordonnancement s'applique aux ressources que l'application peut demander de façon implicite. Pour continuer à fonctionner, par exemple, une application envoie une demande implicite afin de bénéficier de temps de CPU supplémentaire. Pour écrire les données dans un socket de réseau, une application demande de façon implicite une bande passante. Les contraintes peuvent s'appliquer à l'utilisation totale cumulée d'une ressource demandée de façon implicite.

Des interfaces supplémentaires peuvent être présentées pour permettre une négociation explicite de la bande passante ou des niveaux de service de la CPU. Les ressources faisant l'objet d'une demande explicite (demande de thread supplémentaire, par exemple), peuvent être gérées au moyen de contraintes.

Mécanismes de contrôle de la gestion des ressources

Les mécanismes de contrainte, de programmation et de partitionnement sont les trois types de mécanisme de contrôle disponibles dans le système d'exploitation Oracle Solaris.

Mécanismes de contrainte

Les contraintes permettent à l'administrateur ou au développeur d'application de fixer des limites de consommation de ressources spécifiques pour une charge de travail. Lorsque les limites sont connues, la modélisation des scénarios de consommation de ressources est d'autant plus facile. Les limites peuvent également servir à identifier les applications au comportement anormal qui risqueraient de compromettre les performances ou la disponibilité du système par des demandes de ressources non régulées.

Les contraintes représentent des complications pour l'application. Il est possible de changer la relation entre l'application et le système jusqu'au point où l'application ne fonctionne plus. Pour éviter de prendre un risque trop important, le mieux est de réduire progressivement les contraintes qui pèsent sur les applications dont le comportement en matière d'utilisation des ressources est incertain. La fonctionnalité de contrôle des ressources dont il est question au [Chapitre 6, A propos des contrôles de ressources](#) offre un mécanisme de contrainte. Il est possible de tenir compte de ces contraintes lors du développement de nouvelles applications, mais ce n'est pas nécessairement la préoccupation principale de tous les développeurs.

Mécanismes de programmation

Le terme programmation désigne la série de choix effectués à intervalles spécifiques en termes d'allocation des ressources. Les décisions sont basées sur un algorithme prévisible. Une application n'ayant pas besoin de son allocation actuelle laisse les ressources à la disposition d'une autre application. Une gestion programmée des ressources permet de faire le meilleur usage possible d'une configuration sous-utilisée, tout en permettant des allocations contrôlées dans un scénario exigeant un engagement important ou surévalué. L'algorithme sous-jacent définit le mode d'interprétation du terme "contrôlé". Dans certains cas, l'algorithme de programmation peut garantir que toutes les applications ont accès à la ressource. L'ordonnanceur FSS décrit au [Chapitre 8, A propos de l'ordonnanceur de partage équitable](#) gère de cette manière l'accès des applications aux ressources de la CPU.

Mécanismes de partitionnement

Le partitionnement sert à lier une charge de travail à un sous-ensemble des ressources disponibles du système. Cette liaison vous assure qu'une quantité de ressources connue est toujours réservée à la charge de travail. La fonction de pools de ressources décrite au [Chapitre 12, A propos des pools de ressources](#) permet justement de restreindre les charges de travail à des sous-ensembles spécifiques de la machine.

Les configurations tirant parti du partitionnement peuvent éviter un surengagement à l'échelle du système. Vous risquez toutefois de ne plus atteindre les seuils optimaux d'utilisations. Il faut savoir qu'un groupe réservé de ressources, comme des processeurs, n'est pas exploitable par une autre charge de travail lorsque la charge de travail à laquelle elles sont liées est inactive.

Configuration de la gestion des ressources

Il est possible d'intégrer des parties de la configuration de la gestion des ressources dans un service de noms du réseau. L'administrateur sera ainsi capable d'appliquer les contraintes de gestion des ressources à un ensemble de machines, et non pas sur une base individuelle. Les opérations correspondantes peuvent partager le même identificateur et l'utilisation cumulée des ressources peut être ventilée à partir des données de comptabilisation.

La configuration de la gestion des ressources et les identificateurs orientés charges de travail sont traités en détail dans le [Chapitre 2, A propos des projets et tâches](#). L'utilitaire de comptabilisation étendue chargé de lier ces identificateurs à l'utilisation des ressources par l'application est décrit dans le [Chapitre 4, A propos de la comptabilisation étendue](#).

Interaction avec les zones non globales

Il est possible d'utiliser les fonctions de gestion des ressources avec des zones afin d'affiner l'environnement applicatif. Les interactions entre ces fonctions et les zones sont décrites dans les sections correspondantes de ce manuel.

Intérêt de la gestion des ressources

La gestion des ressources permet de s'assurer que les temps de réponse sont satisfaisants pour vos applications.

C'est également un moyen d'optimiser l'utilisation des ressources. En définissant des catégories d'utilisation et en fixant les priorités appropriées, vous pouvez profiter au mieux de la capacité de réserve pendant les périodes creuses et éviter d'investir dans une puissance de traitement supplémentaire. C'est enfin la garantie pour vous qu'aucune ressource précieuse n'est gaspillée inutilement du fait de la variabilité des charges de travail.

Consolidation serveur

La gestion des ressources est idéale pour les environnements consolidant un certain nombre d'applications sur un serveur unique.

La gestion de nombreuses machines est relativement coûteuse et complexe. C'est la raison pour laquelle il est souvent préférable de consolider plusieurs applications sur des serveurs évolutifs de plus grande taille. Au lieu d'exécuter chaque charge de travail sur un système indépendant et d'autoriser un accès complet aux ressources de ce système, vous pouvez vous servir d'un logiciel de gestion des ressources pour séparer les charges de travail à l'intérieur du système. La gestion des ressources permet d'abaisser le coût total de possession en exécutant et en contrôlant des applications de diverses natures sur un même système Oracle Solaris.

Si vous offrez des services Internet et des services applicatifs, la gestion des ressources est un outil précieux pour :

- Héberger plusieurs serveurs Web sur une seule machine. Vous pouvez contrôler la consommation des ressources pour chaque site Web et protéger chaque site des excès potentiels des autres sites.
- Empêcher un script CGI (Common Gateway Interface, interface de passerelle commune) défaillant d'épuiser les ressources de la CPU.
- Eviter qu'une application au comportement anormal accapare toute la mémoire virtuelle disponible.
- S'assurer que les applications d'un client ne sont pas affectées par les applications d'un autre client fonctionnant au niveau du même site.
- Offrir des niveaux différenciés de classes de service sur la même machine.

- Obtenir des données de comptabilisation à des fins de facturation.

Prise en charge d'une population importante et variée d'utilisateurs

Les fonctions de gestion des ressources sont particulièrement recommandées si votre système comporte une base d'utilisateurs importante et diversifiée, par exemple dans le cas d'un établissement d'enseignement. Si vous traitez des charges de travail de différentes natures, n'hésitez pas à configurer le logiciel pour accorder la priorité à certains projets par rapport aux autres.

Par exemple, dans les grandes maisons de courtage, les négociants ont occasionnellement besoin d'un accès rapide afin d'effectuer une interrogation ou un calcul. Les autres utilisateurs du système ont toutefois une charge de travail plus uniforme. Il est conseillé, dans ce cas, d'allouer une quantité de puissance de traitement proportionnellement plus importante aux projets des négociants pour répondre plus efficacement à leurs demandes.

La gestion de ressources est également un outil idéal pour prendre en charge les systèmes à clients légers. Ces plates-formes offrent des consoles sans état équipées de mémoires graphiques et de périphériques d'entrée tels que des cartes à puce. Le calcul est effectué depuis un serveur partagé, ce qui permet de profiter d'un environnement à partage de temps (TS, TimeSharing). Tirez parti des fonctions de gestion des ressources pour isoler les utilisateurs sur le serveur. Un utilisateur générant des charges de travail excessives ne pourra donc pas monopoliser les ressources matérielles au détriment des autres utilisateurs du système.

Configuration de la gestion des ressources (liste des tâches)

La liste des tâches suivante permet d'avoir une idée précise des étapes nécessaires à la mise en place de la gestion des ressources sur votre système.

Tâche	Description	Pour obtenir des instructions
Identification des charges de travail sur votre système et classement de chacune d'elles par projet	Créez des entrées de projet dans le fichier / etc/project, dans la carte NIS ou dans le service d'annuaire LDAP.	“Base de données project” à la page 23
Définition des charges de travail prioritaires sur votre système	Déterminez les applications jouant un rôle primordial. Il est possible que ces charges de travail exigent un accès préférentiel aux ressources.	Se référer aux objectifs de service de l'entreprise
Surveillance de l'activité en temps réel sur votre système	Servez-vous des outils d'évaluation des performances pour connaître la consommation actuelle en ressources des	Pages de manuel cpustat(1M) , iostat(1M) , mpstat(1M) , prstat(1M) , sar(1) , et vmstat(1M)

Tâche	Description	Pour obtenir des instructions
	charges de travail s'exécutant sur votre système. Vous pouvez vérifier ensuite s'il est impératif ou non de restreindre l'accès à une ressource donnée ou d'isoler des charges de travail particulières.	
Modification temporaire des charges de travail s'exécutant sur votre système	Pour déterminer les valeurs qu'il est possible de modifier, reportez-vous aux contrôles de ressources disponibles dans le système Oracle Solaris. Vous pouvez actualiser les valeurs à partir de la ligne de commande pendant l'exécution de la tâche ou du processus.	Pages de manuel " Contrôles de ressources disponibles " à la page 62, " Actions globales et locales applicables aux valeurs de contrôle de ressource " à la page 68, " Mise à jour temporaire des valeurs de contrôle de ressource sur un système en cours d'exécution " à la page 73 et rctladm(1M) et prctl(1) .
Définition des contrôles de ressources et des attributs de projet pour chaque entrée de projet dans la base de données project ou dans la base de données de projet de service de noms	Chaque entrée de projet du fichier /etc/project ou de la base de données de projet de service de noms peut contenir un ou plusieurs contrôles de ressources ou attributs. Les contrôles de ressources appliquent des restrictions aux tâches et processus rattachés à ce projet. Vous pouvez associer une ou plusieurs actions spécifiques à chaque valeur de seuil définie pour un contrôle de ressource. Les actions sont déclenchées dès que le seuil en question est atteint. Il est possible d'instaurer les contrôles de ressources à partir de l'interface de ligne de commande.	" Base de données project " à la page 23, " Format de fichier /etc/project local " à la page 25, " Contrôles de ressources disponibles " à la page 62, " Actions globales et locales applicables aux valeurs de contrôle de ressource " à la page 68, et Chapitre 8 , A propos de l'ordonnanceur de partage équitable
Définition d'une limite maximale d'utilisation de la mémoire physique par les ensembles de processus rattachés à un projet	Le démon de limitation des ressources impose la limite d'utilisation de la mémoire physique fixée pour l'attribut rcap.max-rss du projet dans le fichier /etc/project.	" Base de données project " à la page 23 et Chapitre 10 , A propos du contrôle de la mémoire physique à l'aide du démon de limitation des ressources
Création de configurations de pools de ressources	Les pools de ressources permettent de partitionner les ressources système, telles que les processeurs, et de conserver ces partitions d'une réinitialisation à l'autre. Vous pouvez ajouter un attribut project.pool à chaque entrée du fichier /etc/project.	" Base de données project " à la page 23 et Chapitre 12 , A propos des pools de ressources
Désignation de l'ordonnanceur FSS comme ordonnanceur système par défaut	Assurez-vous que tous les processus utilisateur dans un système doté d'une seule CPU ou dans un jeu de processeurs appartiennent à la même classe de programmation.	Section " Configuration de l'ordonnanceur FSS " à la page 103 et page de manuel dispadm(1M)
Activation de l'utilitaire de comptabilisation étendue pour surveiller et enregistrer la consommation des ressources par tâche ou processus	Servez-vous des données de comptabilisation étendue pour évaluer les contrôles de ressources actuels et planifier les besoins en matière de capacité pour les futures charges de travail. Vous pouvez faire le suivi de l'utilisation cumulée à l'échelle du système. Pour obtenir des statistiques complètes pour les charges de travail	" Activation de la comptabilisation étendue des flux, processus, tâches et composants réseau " à la page 52 et page de manuel acctadm(1M)

Tâche	Description	Pour obtenir des instructions
	connexes s'étendant à plusieurs systèmes, il est possible de partager le nom du projet entre plusieurs machines.	
(Facultatif) Modification des valeurs à partir de la ligne de commande s'il est nécessaire d'effectuer des ajustements supplémentaires à votre configuration. Vous pouvez changer les valeurs pendant l'exécution de la tâche ou du processus.	Les modifications apportées aux tâches existantes peuvent être appliquées de façon temporaire sans redémarrer le projet. Réglez les valeurs jusqu'à ce que vous soyez satisfait des performances. Actualisez ensuite les valeurs actuelles dans le fichier / etc/project ou dans la base de données de projet de service de noms.	Pages de manuel "Mise à jour temporaire des valeurs de contrôle de ressource sur un système en cours d'exécution" à la page 73 et rctladm(1M) et prctl(1)
(Facultatif) Capture des données de comptabilisation étendue	Ecrivez les enregistrements de comptabilisation étendue correspondant aux processus actifs et aux tâches actives. Les fichiers générés vous seront très utiles pour procéder aux opérations de planification, d'imputation et de facturation. Il existe une interface Perl (Practical Extraction and Report Language) prévue spécialement pour l'API libexacct qui permet de développer des scripts de génération de rapports et d'extraction personnalisés.	Page de manuel wracct(1M) et "Interface Perl pour libexacct" à la page 48

♦ ♦ ♦ CHAPITRE 2

A propos des projets et tâches

Ce chapitre présente les utilitaires de *projet* et de *tâche* propres à la gestion de ressources Oracle Solaris. Les projets et les tâches servent à identifier les charges de travail et à les distinguer les unes des autres.

Ce chapitre comprend les sections suivantes :

- “Utilitaires de projet et de tâche” à la page 21
- “Identificateurs de projet” à la page 22
- “Identificateurs des tâches” à la page 27
- “Commandes utilisées avec les projets et les tâches” à la page 29

Pour tirer parti des utilitaires de projet et de tâche, reportez-vous au [Chapitre 3, Administration des projets et des tâches](#).

Utilitaires de projet et de tâche

Pour optimiser le traitement des charges de travail, il est important dans un premier temps d'identifier celles qui sont en cours d'exécution sur le système que vous analysez. Vous ne pouvez pas vous contenter pour cela d'utiliser une méthode entièrement orientée processus ou orientée utilisateur. Le système Oracle Solaris met à votre disposition deux utilitaires supplémentaires très pratiques pour distinguer et reconnaître les charges de travail : le projet et la tâche. Un *projet* fournit un identificateur administratif réseau pour tous les travaux associés. Une *tâche* réunit les processus sous forme d'une entité gérable représentant un composant charge de travail.

Les contrôles spécifiés dans la base de données de service de noms project s'appliquent au processus, à la tâche et au projet. Comme les contrôles de processus et de tâche sont hérités des appels système fork et settaskid, l'ensemble des processus et des tâches créés au sein du projet héritent de ces contrôles. Pour plus d'informations sur ces appels système, reportez-vous aux pages de manuel [fork\(2\)](#) et [settaskid\(2\)](#).

En se basant sur le critère d'appartenance au projet ou à la tâche, il est possible de manipuler les processus en cours avec des commandes Oracle Solaris standard. L'utilitaire de comptabilisation étendue permet d'établir des rapports à la fois sur l'utilisation des processus et sur l'utilisation

des tâches et d'attribuer à chaque enregistrement l'identificateur de projet maître. Cela permet de mettre en corrélation l'analyse des charges de travail hors ligne avec les méthodes de surveillance en ligne. L'identificateur de projet peut être partagé entre plusieurs machines via la base de données de service de noms `project`. Il est donc possible de procéder à une analyse globale de l'utilisation des ressources des charges de travail connexes s'exécutant (ou s'étendant) sur plusieurs machines.

Identificateurs de projet

L'identificateur de projet est un élément administratif prévu pour identifier les charges de travail. Il peut être comparé à une balise de charge de travail à l'échelle des utilisateurs et des groupes. Un utilisateur ou un groupe peut appartenir à un ou plusieurs projets. L'intérêt de ces projets est de représenter les charges de travail auxquelles l'utilisateur (ou le groupe d'utilisateurs) est autorisé à participer. Il est possible de se baser sur ce critère d'appartenance pour imputer les ressources en fonction de leur utilisation ou des allocations initiales. Bien qu'il soit nécessaire d'assigner l'utilisateur à un projet par défaut, les processus lancés par l'utilisateur peuvent être associés à tout projet dont l'utilisateur est membre.

Détermination du projet par défaut des utilisateurs

Pour se connecter au système, l'utilisateur doit être affecté à un projet par défaut. L'utilisateur fait automatiquement partie de ce projet par défaut, même s'il ne figure pas dans la liste d'utilisateurs ou de groupes spécifiée dans ce projet.

Comme chaque processus du système a une appartenance au projet, il est nécessaire de prévoir un algorithme pour attribuer un projet par défaut au processus de connexion ou autre processus initial. L'algorithme est présenté à la page de manuel `getproject(3C)`. Le système suit des étapes dans un ordre précis pour déterminer le projet par défaut. Si aucun projet par défaut n'est trouvé, la connexion utilisateur ou la demande de démarrage d'un processus est refusée.

Le système procède dans l'ordre suivant pour déterminer le projet par défaut d'un utilisateur :

1. Si l'utilisateur possède une entrée pour laquelle l'attribut `project` a été défini dans la base de données étendue des attributs utilisateur `/etc/user_attr`, la valeur de l'attribut `project` correspond au projet par défaut. Voir la page de manuel `user_attr(4)`.
2. Si un projet répondant au nom `user.user-id` est présent dans la base de données `project`, ce projet est considéré comme le projet par défaut. Pour plus d'informations, reportez-vous à la page de manuel `project(4)`.
3. Si un projet intitulé `group.group-name` est présent dans la base de données `project`, où `group-name` représente le nom du groupe par défaut pour l'utilisateur, comme indiqué dans le fichier `passwd`, ce projet est considéré comme le projet par défaut. Pour plus d'informations sur le fichier `passwd`, reportez-vous à la page de manuel `passwd(4)`.

4. Si le projet spécial `default` est présent dans la base de données `project`, ce projet est considéré comme le projet par défaut.

Cette logique est fournie par la fonction de bibliothèque `getdefaultproj`. () Pour plus d'informations, voir la page de manuel [getproject\(3PROJECT\)](#).

Définition des attributs utilisateur à l'aide des commandes `useradd` et `usermod`

Vous pouvez exécuter les commandes suivantes avec l'option `-K` et une paire *key=value* pour définir les attributs utilisateur dans les fichiers locaux :

<code>useradd</code>	Définir le projet par défaut pour l'utilisateur
<code>usermod</code>	Modifier les informations utilisateur

Les fichiers locaux peuvent correspondre aux fichiers suivants :

- `/etc/group`
- `/etc/passwd`
- `/etc/project`
- `/etc/shadow`
- `/etc/user_attr`

En cas d'utilisation d'un service de noms de réseau tel que NIS pour compléter le fichier local, ces commandes ne permettent pas de changer les informations fournies par ce service. Elles se chargent, en revanche, d'effectuer les vérifications suivantes par rapport à la *base de données du service de noms* externe :

- Utilisation d'un nom d'utilisateur (ou rôle) unique
- Utilisation d'un ID d'utilisateur unique
- Existence des noms de groupe indiqués

Pour plus d'informations, reportez-vous aux pages de manuel [useradd\(1M\)](#), [usermod\(1M\)](#), et [user_attr\(4\)](#).

Base de données `project`

Vous pouvez stocker les données de projet dans un fichier local, dans un système de nom de domaine DNS (Domain Name System), dans une correspondance de projet NIS (Network

Information Service) ou dans un service d'annuaire LDAP (Lightweight Directory Access Protocol). Le fichier `/etc/project` ou le service de noms est utilisé au moment de la connexion et par toutes les demandes de gestion de compte émises par le module d'authentification enfichable (PAM) pour lier un utilisateur à un projet par défaut.

Remarque - Les mises à jour des entrées dans la base de données `project`, apportées au fichier `/etc/project` ou à une représentation de la base de données dans un service de noms du réseau, ne sont pas appliquées aux projets actuellement actifs. Elles concernent les nouvelles tâches rejoignant le projet lorsque vous exécutez la commande `login` ou `newtask`. Pour plus d'informations, reportez-vous aux pages de manuel [login\(1\)](#) et [newtask\(1\)](#).

Sous-système PAM

Parmi les opérations qui modifient ou définissent une identité, on trouve la connexion à un système, l'appel des commandes `rcp` et `rsh`, l'utilisation de la commande `ftp`, ou `su`. Pour ce type d'opération, il est nécessaire d'utiliser un ensemble de modules configurables pour assurer l'authentification, la gestion de compte, la gestion des informations d'identification et la gestion de session.

Pour une présentation du PAM, reportez-vous à [Chapitre 1, “ Utilisation de modules d'authentification enfichables ”](#) du manuel “ [Gestion de Kerberos et d'autres services d'authentification dans Oracle Solaris 11.2](#) ”.

Configuration des services de noms

La gestion des ressources prend en charge les bases de données `project` de services de noms. L'emplacement de la base de données `project` est défini dans le fichier `/etc/nsswitch.conf`. Par défaut, `files` figure en premier dans la liste, mais les sources peuvent être répertoriées dans n'importe quel ordre.

```
project: files [nis] [ldap]
```

Si la liste contient plusieurs sources d'informations de projet, le fichier `nsswitch.conf` s'assure que la routine commence par rechercher les informations dans la première source répertoriée avant de s'intéresser aux sources suivantes.

Pour plus d'informations sur le fichier `/etc/nsswitch.conf`, voir [Chapitre 2, “ A propos du commutateur du service de noms ”](#) du manuel “ [Utilisation des services de noms et d'annuaire Oracle Solaris 11.2 : DNS et NIS](#) ” et [nsswitch.conf\(4\)](#).

Format de fichier /etc/project local

Si vous sélectionnez `files` comme source de base de données `project` dans le fichier `nsswitch.conf`, le processus de connexion recherche les informations de projet dans le fichier `/etc/project`. Pour plus d'informations, reportez-vous aux pages de manuel [projects\(1\)](#) et [project\(4\)](#).

Le fichier `project` contient une entrée d'une seule ligne sous la forme suivante pour chaque projet reconnu par le système :

```
projname:projid:comment:user-list:group-list:attributes
```

Les champs sont définis de la manière suivante :

<i>projname</i>	Nom du projet. Le nom doit être une chaîne composée de caractères alphanumériques, de caractères de soulignement (<code>_</code>), de tirets (<code>-</code>) et de points (<code>.</code>). La période, qui est réservé aux projets ayant une signification particulière pour le système d'exploitation, peut être utilisé uniquement dans les noms de projets par défaut des utilisateurs. <i>projname</i> ne doit pas contenir deux-points (<code>:</code>) ni de caractères de saut de ligne.
<i>projid</i>	Identificateur numérique unique du projet (IDPROJ) au sein du système. La valeur maximum du champ <i>projid</i> est <code>UID_MAX</code> (2147483647).
<i>comment</i>	Description du projet.
<i>user-list</i>	Liste des utilisateurs (noms séparés par des virgules) ayant le droit de participer au projet. Ce champ peut contenir des caractères génériques. L'astérisque (*) signifie que tous les utilisateurs rejoignent le projet. Un point d'exclamation suivi d'un astérisque (!*) a pour effet d'exclure tous les utilisateurs du projet. Le point d'exclamation (!) suivi d'un nom d'utilisateur permet d'exclure l'utilisateur spécifié du projet.
<i>group-list</i>	Liste des groupes d'utilisateurs (noms séparés par des virgules) ayant le droit de participer au projet. Ce champ peut contenir des caractères génériques. L'astérisque (*) signifie que tous les groupes rejoignent le projet. Un point d'exclamation suivi d'un astérisque (!*) a pour effet d'exclure tous les groupes du projet. Le point d'exclamation (!) suivi d'un nom de groupe permet d'exclure le groupe spécifié du projet.
<i>attributes</i>	Liste de paires nom-valeur séparées par un point-virgule, tels que les contrôles de ressources (voir le Chapitre 6, A propos des contrôles de

[ressources](#)). *name* représente une chaîne arbitraire qui définit l'attribut ayant trait à l'objet et *value* représente la valeur facultative de cet attribut.

`name[=value]`

Dans la paire nom-valeur, les noms peuvent être composés de lettres, de chiffres, de traits de soulignement et de points. Par convention, le point sert de séparateur entre les catégories et les sous-catégories du contrôle de ressource (rctl). Un nom d'attribut doit impérativement commencer par une lettre. Il fait également la distinction entre les minuscules et les majuscules.

Vous pouvez structurer les valeurs en utilisant des virgules et des parenthèses pour définir l'ordre de priorité.

Le point-virgule sert de caractère de séparation pour les paires nom-valeur. Vous ne pouvez pas utiliser ce symbole dans une définition de valeur. Le signe deux-points sert de caractère de séparation pour les champs du projet. Il est interdit de l'utiliser dans une définition de valeur.

Remarque - Les routines de lecture de ce fichier s'interrompent chaque fois qu'elles rencontrent une entrée mal formée. Les projets spécifiés après l'entrée incorrecte ne sont pas assignés.

L'exemple suivant montre à quoi ressemble le fichier `/etc/project` par défaut :

```
system:0:::  
user.root:1:::  
noproject:2:::  
default:3:::  
group.staff:10:::
```

L'exemple suivant présente le contenu du fichier `/etc/project` par défaut avec des entrées de projet ajoutées à la fin :

```
system:0:::  
user.root:1:::  
noproject:2:::  
default:3:::  
group.staff:10:::  
user.ml:2424:Lyle Personal::  
booksite:4113:Book Auction Project:ml,mp,jtd,kjh::
```

Vous êtes libre également d'ajouter des contrôles de ressources et des attributs au fichier `/etc/project` :

- Pour ajouter des contrôles de ressources pour un projet, reportez-vous à la section [“Définition des contrôles des ressources”](#) à la page 76.
- Pour définir une limitation d'utilisation des ressources de mémoire physique pour un projet à l'aide du démon d'allocation restrictive décrit à la page de manuel [rcapd\(1M\)](#), voir [“Attribut permettant de limiter l'utilisation de la mémoire physique pour les projets”](#) à la page 109.

- Pour ajouter un attribut `project.pool` à l'entrée d'un projet, reportez-vous à la section [“Création de la configuration”](#) à la page 176.

Configuration de projet pour le système d'information réseau NIS

Si vous avez recours au système d'information réseau NIS, vous pouvez configurer le fichier `/etc/nsswitch.conf` dans le but de rechercher les correspondances de projets NIS pour les projets :

```
project: nis files
```

Les cartes NIS, `project.byname` ou `project.bynumber`, se présentent sous la même forme que le fichier `/etc/project` :

```
projname:projid:comment:user-list:group-list:attributes
```

Pour plus d'informations, reportez-vous à [Chapitre 5, “A propos du service d'information réseau”](#) du manuel [“Utilisation des services de noms et d'annuaire Oracle Solaris 11.2 : DNS et NIS”](#).

Configuration de projet pour le service d'annuaire LDAP

Si vous avez recours au service d'annuaire LDAP, vous pouvez configurer le fichier `/etc/nsswitch.conf` dans le but d'effectuer des recherches dans la base de données `project` pour les projets :

```
project: ldap files
```

Pour plus d'informations, reportez-vous à [Chapitre 1, “Introduction au service de noms LDAP”](#) du manuel [“Utilisation des services de noms et d'annuaire Oracle Solaris 11.2 : LDAP”](#).

Pour plus d'informations sur le schéma des entrées de projet dans une base de données LDAP, reportez-vous à [“Schémas Oracle Solaris”](#) du manuel [“Utilisation des services de noms et d'annuaire Oracle Solaris 11.2 : LDAP”](#).

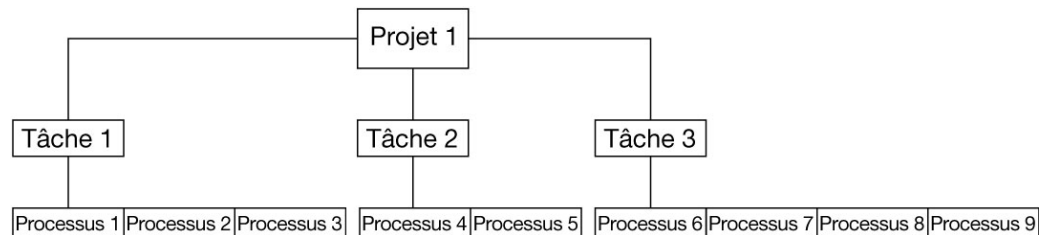
Identificateurs des tâches

Chaque connexion réussie à un projet génère une nouvelle *tâche* dans laquelle s'inscrit le processus de connexion. La tâche correspond à un regroupement de processus représentant un ensemble d'opérations dans le temps. La tâche peut également être considérée comme un

composant charge de travail. Un ID de tâche est attribué automatiquement à chacune des tâches.

Chaque processus est membre d'une tâche et chaque tâche est associée à un projet.

FIGURE 2-1 Vue du projet et des tâches sous forme d'arborescence



Toutes les opérations s'appliquant aux groupes de processus, telles que la distribution du signal, sont également prises en charge au niveau des tâches. Il est possible, en outre, de lier une tâche à un *jeu de processeurs* et de définir un ordre de priorité et une classe pour la tâche, ce qui aura pour effet de modifier tous les processus actuels et suivants dans la tâche.

Le programme crée une tâche chaque fois qu'un utilisateur rejoint un projet. Voici l'ensemble des actions, commandes et fonctions ayant pour effet de générer des tâches :

- login
- cron
- newtask
- setproject
- su

Vous pouvez créer une tâche finalisée à l'aide de l'une des méthodes suivantes. Toute autre tentative échouera.

- Exécutez la commande `newtask` avec l'option `-F`.
- Appliquez l'attribut `task.final` à un projet dans la base de données de services de noms `project`. Toutes les tâches créées dans ce projet par la commande `setproject` sont identifiées par l'indicateur `TASK_FINAL`.

Pour plus d'informations, reportez-vous aux pages de manuel [login\(1\)](#), [newtask\(1\)](#), [cron\(1M\)](#), [su\(1M\)](#), et [setproject\(3PROJECT\)](#).

L'utilitaire de comptabilisation étendue peut fournir des données comptables aux processus. L'agrégation des données est effectuée au niveau de la tâche.

Commandes utilisées avec les projets et les tâches

Les commandes présentées dans le tableau suivant assurent l'interface d'administration principale avec les utilitaires de gestion des projets et des tâches.

Référence aux pages de manuel	Description
projects(1)	Affiche les appartenances au projet pour les utilisateurs. Répertorie les projets de la base de données <code>project</code> . Présente des informations au sujet de projets donnés. Si aucun nom de projet n'est fourni, les informations concernent l'ensemble des projets. Exécutez la commande <code>projects</code> avec l'option <code>-l</code> pour obtenir une sortie détaillée.
newtask(1)	Exécute le shell par défaut de l'utilisateur ou la commande indiquée, en plaçant la commande d'exécution dans une nouvelle tâche appartenant à l'objet spécifié. <code>newtask</code> peut également servir à changer la liaison de la tâche et du projet pour un processus en cours. Associez-la à l'option <code>-F</code> pour créer une tâche finalisée.
projadd(1M)	Ajoute une nouvelle entrée de projet au fichier <code>/etc/project</code>. La commande <code>projadd</code> crée une entrée de projet uniquement sur le système local. <code>projadd</code> ne permet pas de modifier les informations fournies par le service de noms du réseau. Elle peut servir à modifier des fichiers de projet autres que le fichier par défaut, <code>/etc/project</code>. Elle assure la vérification de la syntaxe pour le fichier <code>project</code>. Elle permet de valider et de modifier les attributs de projet. Elle accepte les valeurs scalaires.
projmod(1M)	Modifie les informations relatives à un projet sur le système local. <code>projmod</code> ne permet pas de modifier les informations fournies par le service de noms du réseau. En revanche, la commande se charge de vérifier si le nom du projet et l'ID du projet sont uniques par comparaison avec le service de noms externe. Elle peut servir à modifier des fichiers de projet autres que le fichier par défaut, <code>/etc/project</code>. Elle assure la vérification de la syntaxe pour le fichier <code>project</code>. Elle permet de valider et de modifier les attributs de projet. Elle est pratique pour ajouter un nouvel attribut, ajouter des valeurs à un attribut ou supprimer un attribut. Elle accepte les valeurs scalaires. Exécutez la commande avec l'option <code>-A</code> pour appliquer au projet actif les valeurs de contrôle de ressources définies dans la base de données du projet. Les valeurs qui ne correspondent pas à celles définies dans le fichier <code>project</code> sont supprimées.
projdel(1M)	Supprime un projet du système local. <code>projdel</code> ne permet pas de modifier les informations fournies par le service de noms du réseau.
useradd(1M)	Ajoute des définitions de projet par défaut aux fichiers locaux. Exécutez-la avec l'option <code>-K key=value</code> pour ajouter ou remplacer des attributs.
userdel(1M)	Supprime un compte de l'utilisateur du fichier local.
usermod(1M)	Modifie les informations de connexion d'un utilisateur sur le système. Exécutez-la avec l'option <code>-K key=value</code> pour ajouter ou remplacer des attributs.

BETA
DRAFT

3

♦ ♦ ♦ C H A P I T R E 3

Administration des projets et des tâches

Ce chapitre décrit l'utilisation des utilitaires de projets et de tâches du système de gestion des ressources Oracle Solaris.

Liste des sujets abordés dans ce chapitre :

- [“Exemples de commandes et d'options de commande” à la page 32](#)
- [“Administration des projets” à la page 35](#)

Pour avoir un aperçu des utilitaires de projets et de tâches, reportez-vous au [Chapitre 2, A propos des projets et tâches](#).

Remarque - Si vous utilisez ces utilitaires sur un système Oracle Solaris où des zones sont installées, seuls les processus de la même zone sont visibles par le biais des interfaces d'appel système utilisant des ID de processus lorsque ces commandes sont exécutées dans une zone non globale.

Administration des projets et des tâches (liste des tâches)

Tâche	Description	Pour obtenir des instructions
Affichage d'exemples de commandes et d'options utilisées avec les projets et les tâches	Présentez les ID des tâches et des projets, affichez des statistiques variées au sujet des processus et des projets en cours d'exécution sur votre système.	“Exemples de commandes et d'options de commande” à la page 32
Définition d'un projet.	Ajoutez une entrée de projet dans le fichier <code>/etc/project</code> et modifiez des valeurs pour cette entrée.	“Définition d'un projet et affichage du projet actuel” à la page 35
Supprimer un projet.	Supprimez une entrée de projet du fichier <code>/etc/project</code> .	“Suppression d'un projet du fichier /etc/project” à la page 37
Validation du fichier de projet ou de la base de données du projet	Vérifiez la syntaxe du fichier <code>/etc/project</code> ou du caractère non ambigu du nom et de l'ID de projet par comparaison avec le service de noms externe.	“Validation du contenu du fichier /etc/project” à la page 38

Tâche	Description	Pour obtenir des instructions
Obtention des informations relatives aux données d'appartenance au projet	Affichez les critères d'appartenance actuelle au projet du processus appelant.	“Obtention des informations d'appartenance au projet” à la page 39
Création d'une tâche	Créez une tâche dans un projet particulier à l'aide de la commande newtask.	“Création d'une tâche” à la page 39
Association d'un processus en cours à une autre tâche ou un autre projet	Associez un numéro de processus à un autre ID de tâche dans un projet donné.	“Transfert d'un processus en cours vers une nouvelle tâche” à la page 39
Ajout et traitement d'attributs de projet	Utilisez les commandes d'administration de la base de données du projet pour ajouter, modifier, valider et supprimer des attributs de projet.	“Modification et validation des attributs de projet” à la page 40

Exemples de commandes et d'options de commande

Cette section propose différents exemples illustrant l'utilisation des commandes et des options avec des projets et des tâches.

Options de commande utilisées avec les projets et les tâches

Commande ps

Servez-vous de la commande ps et de l'option -o pour afficher les ID de tâche et de projet. Pour connaître l'ID de projet, par exemple, entrez l'instruction suivante :

```
# ps -o user,pid,uid,projid
USER PID  UID  PROJID
jtd  89430 124  4113
```

Commande id

Associez la commande id à l'option -p pour obtenir l'ID de projet actuel en plus des ID d'utilisateur et de groupe. Il suffit de spécifier l'opérande user pour connaître le projet correspondant à l'identifiant de connexion normal de cet utilisateur :

```
# id -p
uid=124(jtd) gid=10(staff) projid=4113(booksite)
```

Commandes pgrep et pkill

Pour ne sélectionner que les processus correspondant à un ID de projet dans une liste donnée, exécutez les commandes `pgrep` et `pkill` avec l'option `-J` :

```
# pgrep -J projidlist
# pkill -J projidlist
```

Pour ne sélectionner que les processus correspondant à un ID de tâche dans une liste donnée, exécutez les commandes `pgrep` et `pkill` avec l'option `-T` :

```
# pgrep -T taskidlist
# pkill -T taskidlist
```

Commande prstat

Pour obtenir différentes statistiques au sujet des processus et des projets en cours d'exécution sur votre système, exécutez la commande `prstat` avec l'option `-J` :

```
% prstat -J
  PID USERNAME  SIZE  RSS STATE PRI NICE   TIME CPU PROCESS/NLWP
12905 root      4472K 3640K cpu0  59   0  0:00:01 0.4% prstat/1
  829 root         43M   33M sleep  59   0  0:36:23 0.1% Xorg/1
  890 gdm         88M   26M sleep  59   0  0:22:22 0.0% gdm-simple-gree/1
  686 root     3584K 2756K sleep  59   0  0:00:34 0.0% automountd/4
    5 root         0K    0K sleep  99  -20  0:02:43 0.0% zpool-rpool/138
9869 root         44M   17M sleep  59   0  0:02:06 0.0% pool/9
  804 root     7104K 5968K sleep  59   0  0:01:28 0.0% intrd/1
  445 root     7204K 4680K sleep  59   0  0:00:38 0.0% nsd/33
  881 gdm     7140K 5912K sleep  59   0  0:00:06 0.0% gconfd-2/1
  164 root     2572K 1648K sleep  59   0  0:00:00 0.0% pfexecd/3
  886 gdm     7092K 4920K sleep  59   0  0:00:00 0.0% bonobo-activati/2
   45 netcfg    2252K 1308K sleep  59   0  0:00:00 0.0% netcfgd/2
  142 daemon    7736K 5224K sleep  59   0  0:00:00 0.0% kcf/3
   43 root     3036K 2020K sleep  59   0  0:00:00 0.0% dlmgrd/5
  405 root     6824K 5400K sleep  59   0  0:00:18 0.0% hald/5
PROJID  NPROC  SWAP  RSS MEMORY   TIME CPU PROJECT
    1      4 4728K  19M   0.9%  0:00:01 0.4% user.root
    0     111 278M  344M   17%  1:15:02 0.1% system
   10      2 1884K 9132K   0.4%  0:00:00 0.0% group.staff
    3      3 1668K 6680K   0.3%  0:00:00 0.0% default
```

Total: 120 processes, 733 lwps, load averages: 0.01, 0.00, 0.00

Pour obtenir différentes statistiques au sujet des processus et des tâches en cours d'exécution sur votre système, exécutez la commande `prstat` avec l'option `-T` :

```
% prstat -T
  PID USERNAME  SIZE  RSS STATE PRI NICE   TIME CPU PROCESS/NLWP
12907 root      4488K 3588K cpu0  59   0  0:00:00 0.3% prstat/1
```

```

829 root      43M   33M sleep 59    0   0:36:24 0.1% Xorg/1
890 gdm       88M   26M sleep 59    0   0:22:22 0.0% gdm-simple-gree/1
9869 root     44M   17M sleep 59    0   0:02:06 0.0% pool/9
  5 root      0K    0K sleep 99   -20  0:02:43 0.0% zpool-rpool/138
445 root     7204K 4680K sleep 59    0   0:00:38 0.0% nscd/33
881 gdm      7140K 5912K sleep 59    0   0:00:06 0.0% gconfd-2/1
164 root     2572K 1648K sleep 59    0   0:00:00 0.0% pfexecd/3
886 gdm      7092K 4920K sleep 59    0   0:00:00 0.0% bonobo-activati/2
 45 netcfg   2252K 1308K sleep 59    0   0:00:00 0.0% netcfgd/2
142 daemon   7736K 5224K sleep 59    0   0:00:00 0.0% kcf/3
 43 root     3036K 2020K sleep 59    0   0:00:00 0.0% dlmgt/5
405 root     6824K 5400K sleep 59    0   0:00:18 0.0% hald/5
311 root     3488K 2512K sleep 59    0   0:00:00 0.0% picld/4
409 root     4356K 2768K sleep 59    0   0:00:00 0.0% hald-addon-cpuf/1

TASKID  NPROC  SWAP  RSS  MEMORY  TIME  CPU  PROJECT
1401    2 2540K 8120K  0.4%  0:00:00 0.3% user.root
  94    15   84M 162M   7.9%  0:59:37 0.1% system
 561    1   37M 24M    1.2%  0:02:06 0.0% system
  0     2    0K  0K    0.0%  0:02:47 0.0% system
 46     1 4224K 5524K  0.3%  0:00:38 0.0% system
Total: 120 processes, 733 lwps, load averages: 0.01, 0.00, 0.00

```

Remarque - Il est impossible d'utiliser conjointement les options -J et -T.

Utilisation des commandes cron et su avec les projets et les tâches

Commande cron

La commande cron émet une instruction `settaskid` pour s'assurer que chaque opération cron, at, et batch s'exécute dans une tâche distincte et que le projet par défaut de l'initiateur de la commande convient. Les commandes at et batch permettent également de capturer l'ID de projet actuel, ce qui garantit la restauration de l'ID de projet lors de l'exécution d'une opération at.

Commande su

La commande su rejoint le projet par défaut de l'utilisateur cible en créant une tâche lors de la simulation d'une connexion.

Pour passer au projet par défaut de l'utilisateur à l'aide de la commande su, entrez la commande suivante :

```
# su - user
```

Administration des projets

▼ Définition d'un projet et affichage du projet actuel

Cet exemple montre comment utiliser la commande `projadd` pour ajouter une entrée de projet et la commande `projmod` pour modifier cette entrée.

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.
2. Affichez le fichier `/etc/project` par défaut sur votre système à l'aide de l'instruction `projects -l`.

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
noproject
    projid : 2
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
default
    projid : 3
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
group.staff
    projid : 10
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
```

3. Ajoutez un projet intitulé *booksite*. Assignez le projet à un utilisateur nommé *mark* et possédant l'ID 4113.

```
# projadd -U mark -p 4113 booksite
```

4. Affichez à nouveau le fichier /etc/project.

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
noproject
    projid : 2
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
default
    projid : 3
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
group.staff
    projid : 10
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
booksite
    projid : 4113
    comment: ""
    users  : mark
    groups : (none)
    attribs:
```

5. Ajoutez un commentaire de description du projet dans le champ prévu à cet effet.

```
# projmod -c 'Book Auction Project' booksite
```

6. Vérifiez les modifications dans le fichier /etc/project

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
```

```
user.root
  projid : 1
  comment: ""
  users  : (none)
  groups : (none)
  attribs:
noproject
  projid : 2
  comment: ""
  users  : (none)
  groups : (none)
  attribs:
default
  projid : 3
  comment: ""
  users  : (none)
  groups : (none)
  attribs:
group.staff
  projid : 10
  comment: ""
  users  : (none)
  groups : (none)
  attribs:
booksite
  projid : 4113
  comment: "Book Auction Project"
  users  : mark
  groups : (none)
  attribs:
```

Voir aussi Pour lier des projets, des tâches et des processus à un pool, reportez-vous à la section [“Définition des attributs des pools et liaison à un pool”](#) à la page 171.

▼ Suppression d'un projet du fichier /etc/project

Cet exemple montre comment utiliser la commande `projdel` pour supprimer un projet.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Effacez le projet *booksite* à l'aide de la commande `projdel`.**

```
# projdel booksite
```

3. **Affichez le fichier `/etc/project`.**

```
# projects -l
system
  projid : 0
```

```
comment: ""
users : (none)
groups : (none)
attribs:
user.root
projid : 1
comment: ""
users : (none)
groups : (none)
attribs:
noproject
projid : 2
comment: ""
users : (none)
groups : (none)
attribs:
default
projid : 3
comment: ""
users : (none)
groups : (none)
attribs:
group.staff
projid : 10
comment: ""
users : (none)
groups : (none)
attribs:
```

4. **Connectez-vous sous le nom d'utilisateur *mark* et tapez *projects* pour afficher la liste des projets assignés à cet utilisateur.**

```
# su - mark
# projects
default
```

Validation du contenu du fichier /etc/project

En l'absence d'option d'édition, la commande *projmod* valide le contenu du fichier *project*.

Pour valider une carte NIS, entrez l'instruction suivante :

```
# ypcat project | projmod -f -
```

Pour vérifier la syntaxe du fichier /etc/project, entrez l'instruction suivante :

```
# projmod -n
```

Obtention des informations d'appartenance au projet

Associez la commande `id` à l'indicateur `-p` pour afficher l'appartenance actuelle au projet du processus d'appel.

```
$ id -p
uid=100(mark) gid=1(other) projid=3(default)
```

▼ Création d'une tâche

1. Connectez-vous en tant que membre du projet de destination, *booksite* dans cet exemple.
2. Définissez une nouvelle tâche dans le projet *booksite* en exécutant la commande `newtask` avec l'option `-v` (mode détaillé) afin d'obtenir l'ID de la tâche système.

```
machine% newtask -v -p booksite
16
```

L'exécution de la commande `newtask` a pour effet de créer une tâche dans le projet indiqué et de placer le shell par défaut de l'utilisateur dans cette tâche.

3. Affichez les critères d'appartenance actuelle au projet du processus appelant.

```
machine% id -p
uid=100(mark) gid=1(other) projid=4113(booksite)
```

Le processus fait désormais partie du nouveau projet.

▼ Transfert d'un processus en cours vers une nouvelle tâche

Cet exemple montre comment associer un processus en cours d'exécution à une autre tâche et à un nouveau projet. Pour exécuter cette action, vous devez être l'utilisateur `root`, disposer du profil doté des droits requis ou être le propriétaire de la procédure et un membre du nouveau projet.

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.

Remarque - Si vous êtes le propriétaire du processus ou un membre du nouveau projet, vous pouvez ignorer cette étape.

2. **Obtenez l'ID du processus** *book_catalog*.

```
# pgrep book_catalog
8100
```

3. **Associez le processus** *8100* **à un nouvel ID de tâche dans le projet** *booksite*.

```
# newtask -v -p booksite -c 8100
17
```

L'option -c indique que la commande newtask s'applique au processus nommé existant.

4. **Confirmez la tâche pour procéder à la correspondance de l'ID.**

```
# pgrep -T 17
8100
```

Modification et validation des attributs de projet

Vous pouvez vous servir des commandes d'administration de la base de données de projet (projadd et projmod) pour modifier les attributs du projet.

L'option -K affiche une liste d'attributs de remplacement. Les attributs sont délimités par un point-virgule (;). Le fait d'associer l'option -K à l'option -a permet d'ajouter l'attribut ou la valeur de l'attribut. En revanche, associer l'option -K à l'option -r a pour effet de retirer l'attribut ou la valeur d'attribut. Combiner l'option -K à l'option -s permet de remplacer l'attribut ou la valeur d'attribut.

▼ Ajout d'attributs et de valeurs d'attribut à des projets

Exécutez la commande projmod avec les options -a et -K pour ajouter des valeurs à un attribut de projet. Si l'attribut n'existe pas, il est créé.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Ajoutez un attribut de contrôle de ressource** *task.max-lwps* **sans valeur au projet** *myproject*. **Une tâche accédant au projet possède uniquement la valeur système en guise d'attribut.**

```
# projmod -a -K task.max-lwps myproject
```

3. **Insérez ensuite une valeur à l'attribut `task.max-lwps` dans le projet *myproject*. La valeur correspond à un niveau de privilège, à une valeur de seuil et à une action associée au seuil.**

```
# projmod -a -K "task.max-lwps=(priv,100,deny)" myproject
```

4. **Etant donné que les contrôles de ressource peuvent être définis par plusieurs valeurs, vous êtes libre de compléter la liste de valeurs actuelle en utilisant les mêmes options.**

```
# projmod -a -K "task.max-lwps=(priv,1000,sigkill=KILL)" myproject
```

Les différentes valeurs sont séparées par des virgules. L'entrée `task.max-lwps` se présente désormais comme suit :

```
task.max-lwps=(priv,100,deny),(priv,1000,sigkill=KILL)
```

▼ Suppression des valeurs d'attribut des projets

Cette procédure utilise les valeurs suivantes :

```
task.max-lwps=(priv,100,deny),(priv,1000,sigkill=KILL)
```

1. **Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.**
2. **Pour retirer une valeur d'attribut du contrôle de ressource `task.max-lwps` dans le projet *myproject*, exécutez la commande `projmod` avec les options `-r` et `-K`.**

```
# projmod -r -K "task.max-lwps=(priv,100,deny)" myproject
```

Si l'attribut `task.max-lwps` possède plusieurs valeurs telles que :

```
task.max-lwps=(priv,100,deny),(priv,1000,sigkill=KILL)
```

Le programme élimine la première valeur qui correspond. Vous obtenez le résultat suivant :

```
task.max-lwps=(priv,1000,sigkill=KILL)
```

▼ Suppression d'un attribut de contrôle de ressource d'un projet

Pour retirer le contrôle de ressource `task.max-lwps` dans le projet *myproject*, exécutez la commande `projmod` avec les options `-r` et `-K`.

1. **Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.**

2. **Supprimez l'attribut `task.max-lwps` et toutes ses valeurs du projet *myproject* :**

```
# projmod -r -K task.max-lwps myproject
```

▼ Remplacement des attributs et des valeurs d'attribut des projets

Pour remplacer une valeur de l'attribut `task.max-lwps` dans le projet *myproject*, exécutez la commande `projmod` avec les options `-s` et `-K`. Si l'attribut n'existe pas, il est créé.

1. **Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.**
2. **Remplacez les valeurs `task.max-lwps` actuelles par les nouvelles valeurs affichées :**

```
# projmod -s -K "task.max-lwps=(priv,100,none),(priv,120,deny)" myproject
```

Vous obtenez le résultat suivant :

```
task.max-lwps=(priv,100,none),(priv,120,deny)
```

▼ Suppression des valeurs existantes pour un attribut de contrôle de ressource

1. **Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.**
2. **Pour supprimer les valeurs actuelles de l'attribut `task.max-lwps` dans le projet *myproject*, entrez l'instruction suivante :**

```
# projmod -s -K task.max-lwps myproject
```

◆ ◆ ◆ 4 C H A P I T R E 4

A propos de la comptabilisation étendue

En tirant parti des utilitaires de projet et de tâche décrits au [Chapitre 2, A propos des projets et tâches](#) pour étiqueter et séparer les charges de travail, vous avez la possibilité de surveiller la façon dont les ressources sont exploitées par chaque charge de travail. Le sous-système de *comptabilisation étendue* vous aide à recueillir un ensemble détaillé de statistiques sur l'utilisation des ressources tant pour les processus que pour les tâches.

Ce chapitre comprend les sections suivantes :

- “Introduction à la comptabilisation étendue” à la page 43
- “Mode de fonctionnement de la comptabilisation étendue” à la page 44
- “Configuration de la comptabilisation étendue” à la page 47
- “Commandes s'appliquant à la comptabilisation étendue” à la page 48
- “Interface Perl pour libexacct” à la page 48

Pour commencer à utiliser la fonction de comptabilisation étendue, passez à la section “[Activation de la comptabilisation étendue des flux, processus, tâches et composants réseau](#)” à la page 52.

Introduction à la comptabilisation étendue

Le sous-système de comptabilisation étendue permet d'identifier les enregistrements de chaque utilisation du projet correspondant à un travail précis. Vous pouvez également vous servir de la comptabilisation étendue, en parallèle avec le module de comptabilisation de flux IPQoS (Internet Protocol Quality of Service) pour capturer des informations sur les flux de réseau sur un système.

Avant de mettre en oeuvre les mécanismes de gestion des ressources, vous devez d'abord analyser les demandes d'utilisation de ressources émanant des différentes charges de travail d'un système. L'utilitaire de comptabilisation étendue du système d'exploitation Oracle Solaris permet d'enregistrer de façon souple la consommation des ressources système et réseau pour les éléments suivants :

- Tâches.

- Processus.
- Sélecteurs fournis par le module `flowacct` d'IPQoS. Pour plus d'informations, reportez-vous à `ipqos(7IPP)`.
- Gestion réseau. Voir `dladm(1M)` et `flowadm(1M)`.

A la différence des outils de surveillance en ligne qui permettent de mesurer l'utilisation du système en temps réel, la comptabilisation étendue produit un historique de toutes les opérations effectuées. Vous pouvez alors évaluer plus facilement les besoins en termes de capacité des futures charges de travail.

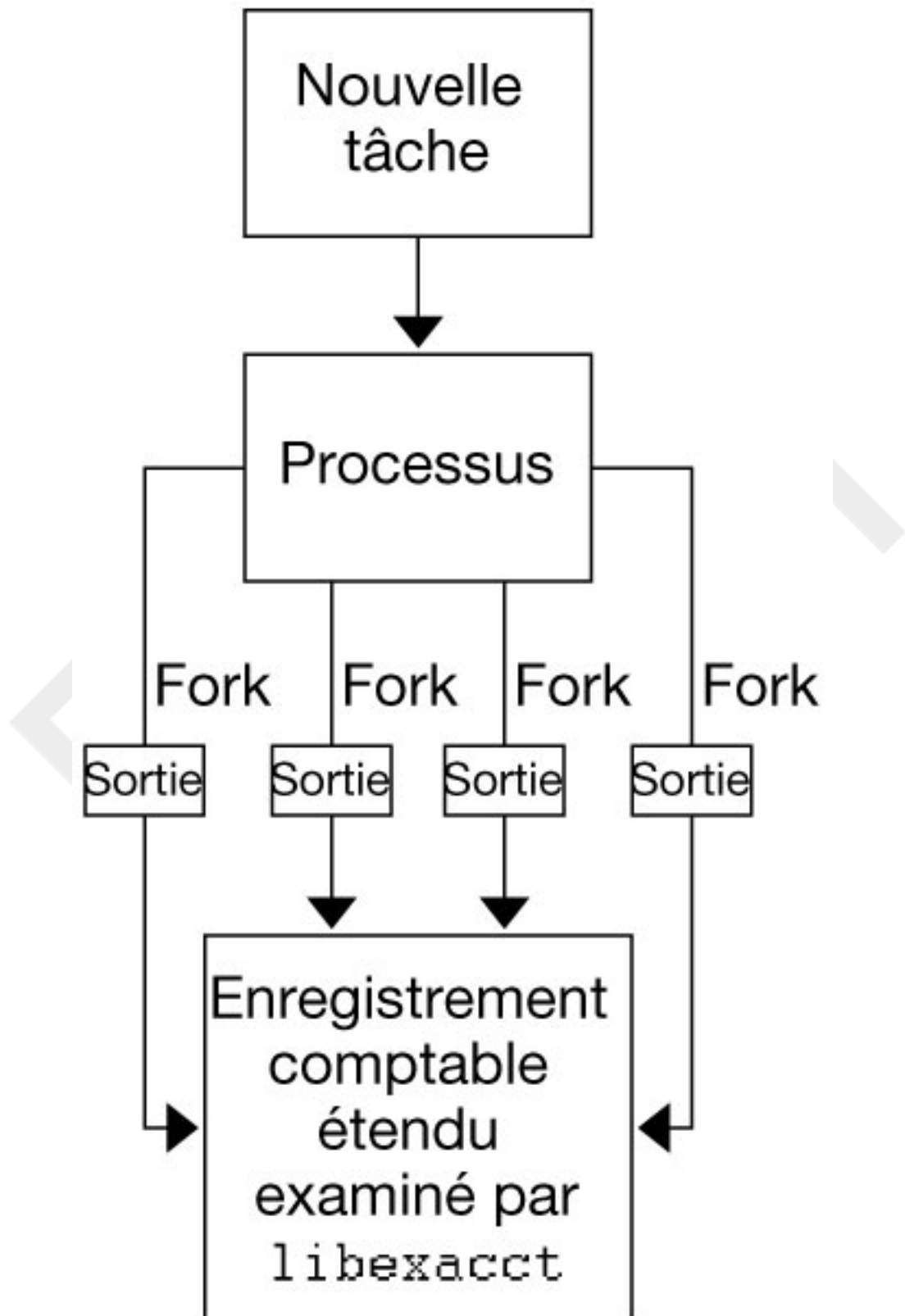
Grâce aux données de comptabilisation étendue disponibles, vous pouvez développer ou acheter des logiciels prévus spécialement pour imputer les ressources, surveiller la charge de travail ou planifier la capacité.

Mode de fonctionnement de la comptabilisation étendue

L'utilitaire de comptabilisation étendue du système d'exploitation Oracle Solaris stocke les données de comptabilisation sous un format de fichier extensible à version multiple. Pour accéder aux fichiers disponibles sous ce format ou créer ce type de fichier, servez-vous de l'API proposée dans la bibliothèque fournie, `libexacct` (voir `libexacct(3LIB)`). Il suffit ensuite d'analyser ces fichiers à partir de la plate-forme sur laquelle la comptabilisation étendue a été activée et d'exploiter leurs données à des fins de planification de capacité et d'imputation des ressources.

Lorsque la fonction de comptabilisation étendue est active, les statistiques compilées sont consultables au moyen de l'API `libexacct`. `libexacct` permet d'examiner les fichiers `exacct` en amont ou en aval. L'API prend en charge les fichiers tiers générés par `libexacct` ainsi que les fichiers créés par le noyau. Il existe une interface Perl (Practical Extraction and Report Language) prévue spécialement pour l'API `libexacct` qui permet de développer des scripts de génération de rapports et d'extraction personnalisés. Reportez-vous à la section “[Interface Perl pour libexacct](#)” à la page 48.

Par exemple, lorsque la fonction de comptabilisation étendue est active, la tâche fait le cumul des ressources utilisées par ses processus membres. L'enregistrement correspondant est écrit au moment de l'exécution de la tâche. Le programme peut également créer des enregistrements intermédiaires relatifs aux processus et aux tâches en cours. Pour plus d'informations sur les tâches, reportez-vous à [Chapitre 2, A propos des projets et tâches](#).



Format extensible

Le format de comptabilisation étendue est considérablement plus extensible que le format du logiciel de comptabilisation de l'ancien système. La comptabilisation étendue permet d'ajouter et de supprimer des mesures comptables du système entre chaque version, voire même pendant le fonctionnement du système.

Remarque - La fonction de comptabilisation étendue et le logiciel de comptabilisation de l'ancien système peuvent être actifs en même temps sur votre système.

Enregistrements et formats exact

Les routines permettant de créer des enregistrements exact ont un double intérêt :

- Créer des fichiers tiers exact.
- Permettre la création d'enregistrements de balisage en vue de les incorporer dans le fichier de comptabilisation du noyau via l'appel système `putacct` (voir [getacct\(2\)](#)).

Remarque - L'appel système `putacct` est également disponible depuis l'interface Perl.

Le format permet de prendre en compte différentes formes d'enregistrements de comptabilisation sans demander un changement de version explicite après chaque modification. Les applications bien conçues exploitant les données de comptabilisation doivent ignorer les enregistrements non reconnus.

La bibliothèque `libexact` convertit et génère des fichiers au format exact. Il s'agit de la seule interface compatible pour les fichiers au format exact.

Remarque - Les appels système `getacct`, `putacct`, et `wracct` ne s'appliquent pas aux flux. Le noyau crée des enregistrements de flux et les écrit dans le fichier lors de la configuration de comptabilisation de flux IPQoS.

Utilisation de la comptabilisation étendue sur un système Oracle Solaris comportant des zones installées

Le sous-système de comptabilisation étendue effectue la collecte de données et produit des rapports pour le système entier (y compris les zones non globales) en cas d'exécution

dans la zone globale. L'administrateur global ou un utilisateur disposant des autorisations adéquates peut également déterminer la consommation des ressources par zone par le biais de l'utilitaire `zonecfg`. Pour plus d'informations, reportez-vous au [Chapitre 2, “ Présentation de la configuration des zones non globales ”](#) du manuel “ [Présentation d'Oracle Solaris Zones](#) ”.

Configuration de la comptabilisation étendue

L'annuaire `/var/adm/exacct` est l'emplacement standard réservé aux données de comptabilisation étendue. Vous pouvez faire appel à la commande `acctadm` pour changer l'emplacement où sont stockés les fichiers des données de comptabilisation des processus et des tâches. Pour plus d'informations, voir [acctadm\(1M\)](#).

Démarrage et activation durable de la comptabilisation étendue

La commande `acctadm` décrite dans [acctadm\(1M\)](#) lance la comptabilisation étendue par le biais de l'utilitaire de gestion des services (SMF) d'Oracle Solaris décrit dans [smf\(5\)](#).

La configuration de la comptabilisation étendue est stockée dans le référentiel SMF. La configuration est restaurée à l'initialisation par une instance de service, une pour chaque type de comptabilisation. Chaque type de comptabilisation étendue est représenté par une instance distincte du service SMF :

<code>svc:/system/ extended- accounting:flow</code>	Comptabilisation du flux
<code>svc:/system/ extended- accounting:process</code>	Comptabilisation des processus
<code>svc:/system/ extended- accounting:task</code>	Comptabilisation des tâches
<code>svc:/system/ extended- accounting:net</code>	Comptabilisation du réseau

L'activation de la comptabilisation étendue à l'aide de [acctadm\(1M\)](#) entraîne l'activation de l'instance de service correspondante, de sorte que la configuration de la comptabilisation

étendue puisse être restaurée à l'initialisation suivante. De la même manière, si la configuration entraîne la désactivation de la comptabilisation pour un service, l'instance de service est désactivée. Les instances sont activées ou désactivées à l'aide de la commande `acctadm` en fonction des besoins.

Pour activer la comptabilisation étendue pour une ressource de façon permanente, exécutez la commande suivante :

```
# acctadm -e resource_list
```

`resource_list` est une liste séparée par des virgules de ressources ou de groupes de ressources.

Enregistrements

La commande `acctadm` ajoute de nouveaux enregistrements à un fichier existant dans `/var/adm/exacct`.

Commandes s'appliquant à la comptabilisation étendue

Référence des commandes	Description
acctadm(1M)	Modifie divers attributs de l'utilitaire de comptabilisation étendue, arrête et démarre la comptabilisation étendue et sert à sélectionner les attributs de comptabilisation permettant d'assurer le suivi des processus, des tâches, des flux et du réseau.
wracct(1M)	Écrit les enregistrements de comptabilisation étendue correspondant aux tâches et processus actifs.
lastcomm(1)	Affiche les commandes précédemment appelées. <code>lastcomm</code> peut utiliser des données de comptabilisation standard ou des données de comptabilisation étendue pour les processus.

Pour plus d'informations sur les commandes associées aux tâches et aux projets, reportez-vous à la section [“Exemples de commandes et d'options de commande”](#) à la page 32.

Pour plus d'informations sur la comptabilisation des flux IPQoS, voir la page de manuel [ipqosconf\(1M\)](#).

Interface Perl pour libexacct

L'interface Perl permet de créer des scripts Perl capables de lire les fichiers de comptabilisation générés par la structure `exacct`. Vous pouvez également créer des scripts Perl qui écrivent des fichiers `exacct`.

Cette interface présente une fonctionnalité équivalente à l'API sous-jacente C. Lorsque cela est possible, les données obtenues à partir de cette API sont présentées en tant que types de données Perl. Cette interface simplifie l'accès aux données et permet d'éviter les opérations de compression et de décompression du tampon. De plus, toutes les opérations liées à la gestion de la mémoire sont réalisées par la bibliothèque Perl.

Les différentes fonctions de projet et de tâche, et les fonctions ayant trait à la structure `exacct` sont réparties par groupe. Chaque groupe de fonctions est placé dans un module Perl spécifique. Chaque module commence par le préfixe Oracle Solaris standard de package Perl `Sun::Solaris::`. Toutes les classes fournies par la bibliothèque Perl `exacct` se trouvent dans le module `Sun::Solaris::Exacct`.

La bibliothèque `libexacct(3LIB)` sous-jacente assure des opérations s'appliquant aux fichiers de format `exacct`, aux balises de catalogue et aux objets `exacct`. Les objets `exacct` sont classés en deux types :

- Les éléments qui correspondent à des valeurs de données uniques (scalaires)
- Les groupes qui constituent des listes d'éléments

Le tableau suivant présente chacun des modules.

Module (sans aucun espace)	Description	Pour en savoir plus
<code>Sun::Solaris::Project</code>	Ce module fournit des fonctions pour accéder aux fonctions de manipulation de projet <code>getprojid(2)</code> , <code>endproject(3PROJECT)</code> , <code>fgetproject(3PROJECT)</code> , <code>getdefaultproj(3PROJECT)</code> , <code>getprojbyid(3PROJECT)</code> , <code>getprojbyname(3PROJECT)</code> , <code>getproject(3PROJECT)</code> , <code>getprojidbyname(3PROJECT)</code> , <code>inproj(3PROJECT)</code> , <code>project_walk(3PROJECT)</code> , <code>setproject(3PROJECT)</code> , et <code>setproject(3PROJECT)</code> .	<code>Project(3PERL)</code>
<code>Sun::Solaris::Task</code>	Ce module fournit des fonctionnalités permettant d'accéder aux fonctions de manipulation de tâche <code>gettaskid(2)</code> et <code>settaskid(2)</code> .	<code>Task(3PERL)</code>
<code>Sun::Solaris::Exacct</code>	Il s'agit du module <code>exacct</code> de niveau supérieur. Ce module fournit des fonctions pour accéder aux appels système de type <code>exacct</code> <code>getacct(2)</code> , <code>putacct(2)</code> , et <code>wracct(2)</code> . Ce module fournit également des fonctions pour accéder à la fonction bibliothèque <code>libexacct(3LIB) ea_error(3EXACCT)</code> . Les constantes s'appliquant à toutes les macros <code>exacct</code> <code>EO_*</code> , <code>EW_*</code> , <code>EXR_*</code> , <code>P_*</code> et <code>TASK_*</code> sont également disponibles dans ce module.	<code>Exacct(3PERL)</code>

Module (sans aucun espace)	Description	Pour en savoir plus
Sun::Solaris::Exacct:: Catalog	Ce module fournit des méthodes orientées objet pour accéder aux champs de bits d'une balise de catalogue exacct. Il donne également accès aux constantes destinées aux macros EXC_*, EXD_* et EXD_*.	Exacct::Catalog(3PERL)
Sun::Solaris::Exacct:: File	Ce module fournit des méthodes orientées objet pour accéder aux fonctions de fichier de comptabilisation libexacct ea_open(3EXACCT) , ea_close(3EXACCT) , ea_get_creator(3EXACCT) , ea_get_hostname(3EXACCT) , ea_next_object(3EXACCT) , ea_previous_object(3EXACCT) , et ea_write_object(3EXACCT) .	Exacct::File(3PERL)
Sun::Solaris::Exacct:: Object	Ce module fournit des méthodes orientées objet pour accéder à un objet de fichier de comptabilisation exacct. Un objet exacct est représenté sous forme d'une référence opaque incorporée à la sous-classe Sun::Solaris::Exacct::Object appropriée. Ce module est subdivisé à son tour en types d'objet Élément et Groupe. A ce niveau, il existe des méthodes pour accéder aux fonctions ea_match_object_catalog(3EXACCT) et ea_attach_to_object(3EXACCT)	Exacct::Object(3PERL)
Sun::Solaris::Exacct:: Object::Item	Ce module fournit des méthodes orientées objet pour accéder à un élément de fichier de comptabilisation exacct. Les objets de ce type héritent de Sun::Solaris::Exacct::Object.	Exacct::Object::Item(3PERL)
Sun::Solaris::Exacct:: Object::Group	Ce module fournit des méthodes orientées objet pour accéder à un groupe de fichiers de comptabilisation exacct. Les objets de ce type héritent de Sun::Solaris::Exacct::Object. Ces objets donnent accès à la fonction ea_attach_to_group(3EXACCT) . Les éléments contenus dans le groupe sont présentés sous forme de tableau Perl.	Exacct::Object::Group(3PERL)
Sun::Solaris::Kstat	Ce module fournit une interface de hachage Perl à l'utilitaire kstat. Vous trouverez un exemple d'utilisation de ce module dans /bin/kstat, écrit en langage Perl.	Kstat(3PERL)

Pour obtenir des exemples d'utilisation des modules décrits dans le tableau précédent, reportez-vous à la section [“Utilisation de l'interface Perl pour accéder à libexacct”](#) à la page 55..

5

♦ ♦ ♦ C H A P I T R E 5

Administration des tâches de comptabilisation étendue

Ce chapitre décrit comment gérer le sous-système de comptabilisation étendue.

Pour avoir un aperçu du sous-système de comptabilisation étendue, reportez-vous au [Chapitre 4, A propos de la comptabilisation étendue](#).

Administration de l'utilitaire de comptabilisation étendue (liste des tâches)

Tâche	Description	Pour obtenir des instructions
Activation de l'utilitaire de comptabilisation étendue	Servez-vous de la comptabilisation étendue pour surveiller l'utilisation des ressources par chaque projet en cours d'exécution sur votre système. Vous pouvez tirer parti du sous-système de <i>comptabilisation étendue</i> pour capturer des données d'historique pour les tâches, les processus et les flux.	“Activation de la comptabilisation étendue des flux, processus, tâches et composants réseau” à la page 52
Affichage de l'état de la comptabilisation étendue	Déterminez l'état de l'utilitaire de comptabilisation étendue.	“Affichage de l'état de la comptabilisation étendue” à la page 53
Affichage des ressources de comptabilisation disponibles	Affichez les ressources de comptabilisation disponibles sur votre système.	“Affichage des ressources de comptabilisation disponibles” à la page 53
Désactivation des instances de comptabilisation des flux, des processus, des tâches et du réseau	Désactivez la fonction de comptabilisation étendue.	“Désactivation de la comptabilisation des processus, des tâches, des flux et de gestion de réseau” à la page 54
Utilisation de l'interface Perl pour tirer parti de l'utilitaire de comptabilisation étendue	Servez-vous de l'interface Perl pour développer des scripts de génération de rapports et d'extraction personnalisés.	“Utilisation de l'interface Perl pour accéder à libexacct” à la page 55

Utilisation de la fonctionnalité de comptabilisation étendue

Les utilisateurs peuvent gérer la comptabilisation étendue (démarrer la comptabilisation, l'arrêter et modifier ses paramètres de configuration) s'ils disposent d'un profil avec droits correspondants au type de comptabilisation étendue à gérer :

- Gestion des flux de comptabilisation étendue
- Gestion des processus
- Gestion des tâches
- Gestion du réseau

▼ Activation de la comptabilisation étendue des flux, processus, tâches et composants réseau

Pour activer l'utilitaire de comptabilisation étendue pour les processus, les tâches, les flux et les composants réseau, servez-vous de la commande `acctadm`. Le paramètre final facultatif pour `acctadm` indique si la commande doit s'appliquer aux composants de comptabilisation des flux, des processus, des tâches système ou du réseau de l'utilitaire.

Remarque - Les rôles contiennent des autorisations et des commandes privilégiées. Pour obtenir plus d'informations sur la création du rôle et son affectation à un utilisateur via la fonction de contrôle d'accès basé sur les rôles (RBAC) d'Oracle Solaris, reportez-vous au manuel “ [Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2](#) ”.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Activez la comptabilisation étendue pour les processus.**

`# acctadm -e extended -f /var/adm/exacct/proc process`
3. **Activez la comptabilisation étendue pour les tâches.**

`# acctadm -e extended,mstate -f /var/adm/exacct/task task`
4. **Activez la comptabilisation étendue pour les flux.**

`# acctadm -e extended -f /var/adm/exacct/flow flow`
5. **Activez la comptabilisation étendue pour le réseau.**

`# acctadm -e extended -f /var/adm/exacct/net net`

Exécutez `acctadm` sur les liens et les flux administrés par les commandes `dladm` et `flowadm`.

Voir aussi Pour plus d'informations, voir [acctadm\(1M\)](#).

Affichage de l'état de la comptabilisation étendue

Entrez `acctadm` sans argument pour afficher l'état actuel de l'utilitaire de comptabilisation étendue.

```
machine% acctadm
      Task accounting: active
      Task accounting file: /var/adm/exacct/task
      Tracked task resources: extended
      Untracked task resources: none
      Process accounting: active
      Process accounting file: /var/adm/exacct/proc
      Tracked process resources: extended
      Untracked process resources: host
      Flow accounting: active
      Flow accounting file: /var/adm/exacct/flow
      Tracked flow resources: extended
      Untracked flow resources: none
```

Dans l'exemple précédent, la comptabilisation des tâches système est active en mode étendu et en mode `mstate`. La comptabilisation des processus et des flux est active en mode étendu.

Remarque - Dans le cadre de la comptabilisation étendue, microstate (`mstate`) fait référence aux données étendues (associées aux transitions de processus microstate) disponibles dans le fichier d'utilisation de processus (voir la page de manuel [proc\(4\)](#)). Ces données permettent d'obtenir un plus grand nombre de détails au sujet des activités du processus que les enregistrements de base ou étendus.

Affichage des ressources de comptabilisation disponibles

Les ressources disponibles peuvent varier d'un système à un autre et d'une plate-forme à une autre. Exécutez la commande `acctadm` avec l'option `-r` pour afficher les groupes de ressources de comptabilisation de votre système.

```
machine% acctadm -r
process:
extended pid,uid,gid,cpu,time,command,tty,projid,taskid,ancpid,wait-status,zone,flag,
memory,mstate    displays as one line
basic      pid,uid,gid,cpu,time,command,tty,flag
task:
extended taskid,projid,cpu,time,host,mstate,anctaskid,zone
basic     taskid,projid,cpu,time
```

```

flow:
extended
saddr,daddr,sport,dport,proto,dsfield,nbytes,npkts,action,ctime,lseen,projid,uid
basic    saddr,daddr,sport,dport,proto,nbytes,npkts,action
net:
    extended name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
    priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpks \
    oerrpks,saddr,daddr,sport,dport,protocol,dsfield
    basic    name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
    priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpks \
    oerrpks

```

▼ Désactivation de la comptabilisation des processus, des tâches, des flux et de gestion de réseau

Pour désactiver la comptabilisation des processus, des tâches, des flux et du réseau, désactivez chacun de ces éléments en exécutant la commande `acctadm` avec l'option `-x`.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**

2. **Désactivez la comptabilisation des processus.**

```
# acctadm -x process
```

3. **Désactivez la comptabilisation des tâches.**

```
# acctadm -x task
```

4. **Désactivez la comptabilisation des flux.**

```
# acctadm -x flow
```

5. **Désactivez la comptabilisation de la gestion de réseau.**

```
# acctadm -x net
```

6. **Assurez-vous que la fonction de comptabilisation a été désactivée pour l'ensemble des éléments.**

```

# acctadm
    Task accounting: inactive
    Task accounting file: none
    Tracked task resources: none
    Untracked task resources: extended
    Process accounting: inactive
    Process accounting file: none
    Tracked process resources: none
    Untracked process resources: extended

```

```
Flow accounting: inactive
Flow accounting file: none
Tracked flow resources: none
Untracked flow resources: extended
Net accounting: inactive
Net accounting file: none
Tracked Net resources: none
Untracked Net resources: extended
```

Utilisation de l'interface Perl pour accéder à libexacct

Affichage récursif du contenu d'un objet exacct

Servez-vous du code suivant pour afficher de façon récursive le contenu d'un objet exacct. Notez que cette fonctionnalité est assurée par la bibliothèque en tant que fonction `Sun::Solaris::Exacct::Object::dump()`. Elle est également disponible par le biais de la fonction de convenance `ea_dump_object()`.

```
sub dump_object
{
    my ($obj, $indent) = @_;
    my $istr = ' ' x $indent;

    #
    # Retrieve the catalog tag. Because we are
    # doing this in an array context, the
    # catalog tag will be returned as a (type, catalog, id)
    # triplet, where each member of the triplet will behave as
    # an integer or a string, depending on context.
    # If instead this next line provided a scalar context, e.g.
    #   my $cat = $obj->catalog()->value();
    # then $cat would be set to the integer value of the
    # catalog tag.
    #
    my @cat = $obj->catalog()->value();

    #
    # If the object is a plain item
    #
    if ($obj->type() == &EO_ITEM) {
        #
        # Note: The '%s' formats provide s string context, so
        # the components of the catalog tag will be displayed
        # as the symbolic values. If we changed the '%s'
        # formats to '%d', the numeric value of the components
        # would be displayed.
        #
        printf("%sITEM\n%s  Catalog = %s|%s|%s\n",
```

```
        $istr, $istr, @cat);
    $indent++;

    #
    # Retrieve the value of the item. If the item contains
    # in turn a nested exacct object (i.e., an item or
    # group), then the value method will return a reference
    # to the appropriate sort of perl object
    # (Exacct::Object::Item or Exacct::Object::Group).
    # We could of course figure out that the item contained
    # a nested item or group by examining the catalog tag in
    # @cat and looking for a type of EXT_EXACCT_OBJECT or
    # EXT_GROUP.
    #
    my $val = $obj->value();
    if (ref($val)) {
        # If it is a nested object, recurse to dump it.
        dump_object($val, $indent);
    } else {
        # Otherwise it is just a 'plain' value, so
        # display it.
        printf("%s Value = %s\n", $istr, $val);
    }

    #
    # Otherwise we know we are dealing with a group. Groups
    # represent contents as a perl list or array (depending on
    # context), so we can process the contents of the group
    # with a 'foreach' loop, which provides a list context.
    # In a list context the value method returns the content
    # of the group as a perl list, which is the quickest
    # mechanism, but doesn't allow the group to be modified.
    # If we wanted to modify the contents of the group we could
    # do so like this:
    #   my $grp = $obj->value(); # Returns an array reference
    #   $grp->[0] = $newitem;
    # but accessing the group elements this way is much slower.
    #
    } else {
        printf("%sGROUP\n%s Catalog = %s|s|s\n",
            $istr, $istr, @cat);
        $indent++;
        # 'foreach' provides a list context.
        foreach my $val ($obj->value()) {
            dump_object($val, $indent);
        }
        printf("%sENDGROUP\n", $istr);
    }
}
```

Création et écriture d'un enregistrement de groupe dans un fichier

Servez-vous de ce script pour définir un nouvel enregistrement de groupe et l'écrire dans un fichier nommé /tmp/exacct.

```
#!/usr/bin/perl

use strict;
use warnings;
use Sun::Solaris::Exacct qw(:EXACCT_ALL);
# Prototype list of catalog tags and values.
my @items = (
    [ &EXT_STRING | &EXC_DEFAULT | &EXD_CREATOR    => "me"      ],
    [ &EXT_UINT32  | &EXC_DEFAULT | &EXD_PROC_PID   => $$        ],
    [ &EXT_UINT32  | &EXC_DEFAULT | &EXD_PROC_UID   => $<        ],
    [ &EXT_UINT32  | &EXC_DEFAULT | &EXD_PROC_GID   => $(         ],
    [ &EXT_STRING  | &EXC_DEFAULT | &EXD_PROC_COMMAND => "/bin/rec" ],
);

# Create a new group catalog object.
my $cat = ea_new_catalog(&EXT_GROUP | &EXC_DEFAULT | &EXD_NONE)

# Create a new Group object and retrieve its data array.
my $group = ea_new_group($cat);
my $ary = $group->value();

# Push the new Items onto the Group array.
foreach my $v (@items) {
    push(@$ary, ea_new_item(ea_new_catalog($v->[0]), $v->[1]));
}

# Open the exacct file, write the record & close.
my $f = ea_new_file('/tmp/exacct', &O_RDWR | &O_CREAT | &O_TRUNC)
    || die("create /tmp/exacct failed: ", ea_error_str(), "\n");
$f->write($group);
$f = undef;
```

Impression du contenu d'un fichier exacct

Servez-vous du code Perl suivant pour afficher le contenu d'un fichier exacct.

```
#!/usr/bin/perl

use strict;
use warnings;
use Sun::Solaris::Exacct qw(:EXACCT_ALL);

die("Usage is dumpexacct <exacct file>\n") unless (@ARGV == 1);
```

```
# Open the exacct file and display the header information.
my $ef = ea_new_file($ARGV[0], &O_RDONLY) || die(error_str());
printf("Creator:  %s\n", $ef->creator());
printf("Hostname: %s\n\n", $ef->hostname());

# Dump the file contents
while (my $obj = $ef->get()) {
    ea_dump_object($obj);
}

# Report any errors
if (ea_error() != EXR_OK && ea_error() != EXR_EOF) {
    printf("\nERROR: %s\n", ea_error_str());
    exit(1);
}
exit(0);
```

Exemple de sortie de **Sun::Solaris::Exacct::Object->dump()**

Voici un exemple de la sortie obtenue en appliquant `Sun::Solaris::Exacct::Object->dump()` au fichier créé dans la section [“Création et écriture d'un enregistrement de groupe dans un fichier” à la page 57](#).

```
Creator:  root
Hostname: localhost
GROUP
    Catalog = EXT_GROUP|EXC_DEFAULT|EXD_NONE
    ITEM
        Catalog = EXT_STRING|EXC_DEFAULT|EXD_CREATOR
        Value = me
    ITEM
        Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_PID
        Value = 845523
    ITEM
        Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_UID
        Value = 37845
    ITEM
        Catalog = EXT_UINT32|EXC_DEFAULT|EXD_PROC_GID
        Value = 10
    ITEM
        Catalog = EXT_STRING|EXC_DEFAULT|EXD_PROC_COMMAND
        Value = /bin/rec
ENDGROUP
```

♦ ♦ ♦ CHAPITRE 6

A propos des contrôles de ressources

Après avoir déterminé la façon dont les ressources sont utilisées par les charges de travail sur votre système, tel que décrit au [Chapitre 4, A propos de la comptabilisation étendue](#), vous pouvez fixer des limites concernant l'utilisation des ressources, pour éviter une surexploitation des ressources. L'utilitaire de *contrôle des ressources* permet justement de mettre en place un mécanisme de contraintes.

Ce chapitre se compose des sections suivantes :

- “Concepts de base sur les contrôles de ressources” à la page 59
- “Configuration des contrôles de ressources et des attributs” à la page 61
- “Application des contrôles de ressources” à la page 73
- “Mise à jour temporaire des valeurs de contrôle de ressource sur un système en cours d'exécution” à la page 73
- “Commandes utilisées avec les contrôles de ressources” à la page 74

Pour plus d'informations sur le mode d'administration des contrôles de ressources, reportez-vous au [Chapitre 7, Administration des contrôles des ressources \(tâches\)](#).

Concepts de base sur les contrôles de ressources

Dans le système d'exploitation Oracle Solaris, le concept de limitation de ressources par processus a été étendu aux entités tâche et projet décrites au [Chapitre 2, A propos des projets et tâches](#). Ces améliorations sont liées à l'utilitaire de contrôle des ressources (rctl). De plus, les allocations définies par le biais des paramètres réglables `/etc/system` sont désormais automatiques ou configurées également via le mécanisme de contrôle des ressources.

Un contrôle de ressource est identifié par le préfixe `zone`, `project`, `task`, ou `process`. Les contrôles de ressources peuvent être observés à l'échelle d'un système. Il est possible de mettre à jour les valeurs des contrôles de ressources sur un système en cours d'exécution.

Pour connaître les contrôles de ressources standard disponibles dans cette version, reportez-vous à la section “[Contrôles de ressources disponibles](#)” à la page 62. Pour plus d'informations sur les contrôles de ressources à l'échelle d'une zone disponible, reportez-vous à la section “[Contrôles de ressources disponibles](#)” à la page 62.

Limites d'utilisation des ressources et contrôles de ressources

Les systèmes UNIX proposent, par tradition, un utilitaire de limitation des ressources (*rlimit*). L'utilitaire *rlimit* permet aux administrateurs de prévoir une ou plusieurs limites numériques quant à l'utilisation des ressources mises à la disposition d'un processus. Ces limites concernent aussi bien le temps CPU par processus que la taille du dump noyau par processus et la taille du tas maximum par processus. La *taille du tas* correspond à la quantité de mémoire de travail allouée au segment de données du processus.

L'utilitaire de contrôle des ressources fournit des interfaces de compatibilité à l'utilitaire de limitation des ressources. Les applications existantes auxquelles des contraintes d'utilisation des ressources ont déjà été appliquées continuent de fonctionner normalement. Ces applications peuvent être observées de la même manière que les applications modifiées dans le but de tirer parti de l'utilitaire de contrôle des ressources.

Communication interprocessus et contrôles de ressources

Les processus peuvent communiquer entre eux en utilisant l'un des différents types de communication interprocessus (IPC). Le protocole de communication interprocessus (IPC) assure le transfert ou la synchronisation des informations entre les processus. L'utilitaire de contrôle des ressources propose des contrôles de ressources qui régissent le comportement des utilitaires IPC du noyau. Ces contrôles de ressources remplacent les paramètres réglables */etc/system*.

Les paramètres obsolètes qui sont utilisés pour initialiser les valeurs par défaut des contrôles de ressources peuvent être inclus dans le fichier */etc/system* de ce système Oracle Solaris. Toutefois, l'utilisation des paramètres obsolètes est déconseillée.

Pour identifier les objets IPC qui participent au projet, exécutez la commande *ipcs* avec l'option *-J*. Pour obtenir un exemple, reportez-vous à la section “[Utilisation de la commande *ipcs*](#)” à la page 84 Pour plus d'informations au sujet de la commande *ipcs*, voir la page de manuel [ipcs\(1\)](#).

Pour plus d'informations sur le réglage du système Oracle Solaris, reportez-vous au “[Manuel de référence des paramètres réglables d'Oracle Solaris 11.2](#)”.

Mécanismes de contrainte par contrôle des ressources

Les contrôles de ressources offrent un mécanisme de contrainte des ressources système. Vous pouvez empêcher les processus, les tâches, les projets et les zones d'utiliser une quantité donnée des ressources système spécifiées. Ce mécanisme permet de gérer plus efficacement le système en empêchant une surexploitation des ressources.

Les mécanismes de contrainte peuvent servir à gérer les processus à planification de la capacité. La contrainte rencontrée peut fournir des indications sur les besoins en ressource de l'application sans refuser nécessairement la ressource à l'application.

Mécanismes d'attribut d'un projet

Les contrôles de ressources peuvent également faire office de simple mécanisme d'attribut pour les utilitaires de gestion des ressources. Par exemple, le nombre de parts de CPU mises à la disposition d'un projet dans la classe de programmation de l'ordonnanceur FSS est défini par le contrôle de ressource `project.cpu-shares`. Etant donné qu'un nombre fixe de parts est attribué au projet par le contrôle, les diverses actions qui entrent en jeu en cas de dépassement d'un contrôle ne sont pas applicables. Dans ce contexte, la valeur actuelle du contrôle `project.cpu-shares` est considérée comme un attribut pour le projet spécifié.

Un autre type d'attribut de projet est utilisé pour régler la façon dont les ensembles de processus associés à un projet utilisent la mémoire physique. Ces attributs sont reconnaissables à leur préfixe `rcap`, par exemple, `rcap.max-rss`. Comme pour un contrôle de ressource, ce type d'attribut est configuré dans la base de données `project`. Cependant, à la différence des contrôles de ressources qui sont synchronisés par le noyau, l'allocation restrictive des ressources est appliquée de façon asynchrone au niveau utilisateur par le démon de limitation des ressources `rcapd`. Pour plus d'informations sur la commande `rcapd`, reportez-vous à [Chapitre 10, A propos du contrôle de la mémoire physique à l'aide du démon de limitation des ressources](#) et à `rcapd(1M)`.

L'attribut `project.pool` sert à spécifier une liaison de pool pour un projet. Pour plus d'informations sur les pools de ressources, reportez-vous au [Chapitre 12, A propos des pools de ressources](#).

Configuration des contrôles de ressources et des attributs

L'utilitaire des contrôles de ressources est configuré via la base de données `project`. Reportez-vous au [Chapitre 2, A propos des projets et tâches](#). Les contrôles de ressources et les autres attributs sont définis dans le dernier champ de l'entrée de la base de données `project`. Les

valeurs associées à chaque contrôle de ressource sont mises entre parenthèses et affichées sous forme de texte simple séparé par des virgules. Les valeurs entre parenthèses intègrent une "clause d'action". Chaque clause d'action est définie par un niveau de privilège, une valeur de seuil et une action associée au seuil en question. A chaque contrôle de ressource peuvent correspondre plusieurs clauses d'action (séparées également par des virgules). L'entrée suivante fixe une valeur limite de processus léger par tâche et le temps CPU maximum par processus au niveau d'une entité de projet. Dans cette configuration, le contrôle `process.max-cpu-time` se chargerait d'envoyer un signal `SIGTERM` au processus au bout d'une heure d'exécution et un signal `SIGKILL` si le processus avait continué à fonctionner pendant une durée totale de 1 heure et 1 minute. Reportez-vous au [Tableau 6-3, "Signaux disponibles pour les valeurs des contrôles de ressources"](#).

```
development:101:Developers:::task.max-lwps=(privileged,10,deny);  
process.max-cpu-time=(basic,3600,signal=TERM),(priv,3660,signal=KILL)  
typed as one line
```

Remarque - Sur les systèmes dont les zones sont activées, les contrôles de ressources à l'échelle des zones sont spécifiés dans la configuration de zone sous un format légèrement différent. Pour plus d'informations, reportez-vous à la section "[Paramétrage des contrôles de ressources à l'échelle de la zone](#)" du manuel "[Présentation d'Oracle Solaris Zones](#)".

La commande `rctladm` permet de procéder à des interrogations d'exécution et à des modifications de l'utilitaire de contrôle des ressources, en fonction d'un *champ d'application global*. La commande `prctl` permet de procéder à des interrogations d'exécution et à des modifications de l'utilitaire de contrôle des ressources, en fonction d'un *champ d'application local*.

Pour plus d'informations, reportez-vous à "[Actions globales et locales applicables aux valeurs de contrôle de ressource](#)" à la page 68, `rctladm(1M)` et `prctl(1)`.

Remarque - Sur un système sur lequel des zones sont installées, il n'est pas possible d'appliquer la commande `rctladm` à une zone non globale pour modifier des paramètres. Vous pouvez vous servir de la commande `rctladm` dans une zone non globale pour afficher l'état de consignation globale de chaque contrôle de ressource.

Contrôles de ressources disponibles

Le tableau suivant présente la liste des contrôles de ressources standard disponibles dans cette version.

Il décrit la ressource à laquelle chaque contrôle s'applique. Il identifie également les unités par défaut utilisées par la base de données `project` pour cette ressource. On distingue deux types d'unité :

- Les quantités représentent une valeur limite.

- Les index représentent l'identificateur maximum valide.

`project.cpu-shares` indique, par exemple, le nombre de parts auquel le projet a droit.
`process.max-file-descriptor` spécifie le numéro de fichier le plus élevé attribuable à un processus par l'appel système [open\(2\)](#).

TABLEAU 6-1 Contrôles de ressources standard de projet, de tâche et de processus

Nom de la commande	Description	Unité par défaut
<code>project.cpu-cap</code>	Quantité absolue des ressources de la CPU pouvant être consommée par un projet. La valeur 100 représente 100 % d'une CPU pour le paramètre <code>project.cpu-cap</code> . La valeur 125 équivaut à 125 %, car 100 % correspond à une capacité de CPU complète sur le système.	Quantité (nombre de CPU)
<code>project.cpu-shares</code>	Nombre de parts de CPU accordées à ce projet et susceptibles d'être utilisées par l'ordonnanceur FSS (voir FSS(7)).	Quantité (partages)
<code>project.max-crypto-memory</code>	Quantité totale de mémoire du noyau utilisable par <code>libpks11</code> pour l'accélération matérielle cryptographique. Les valeurs prévues pour les tampons du noyau et les structures liées à la session sont allouées conformément à ce contrôle de ressource.	Taille (octets)
<code>project.max-locked-memory</code>	Quantité totale de mémoire physique verrouillée autorisée. Lorsque le contrôle <code>priv_proc_lock_memory</code> est appliqué à un utilisateur, pensez également à définir ce contrôle de ressource pour empêcher cet utilisateur de verrouiller toute la mémoire. Notez que ce contrôle de ressource remplace <code>project.max-device-locked-memory</code> , qui a été supprimé.	Taille (octets)
<code>project.max-msg-ids</code>	Nombre maximum d'ID de file d'attente des messages autorisé pour ce projet.	Quantité (ID de file d'attente des messages)
<code>project.max-port-ids</code>	Nombre maximum autorisé de ports pairs.	Quantité (nombre de ports pairs)
<code>project.max-processes</code>	Nombre maximum d'emplacements de table de processus simultanément disponibles pour ce projet. Notez que, dans la mesure où les processus standard et zombie utilisent tous deux les emplacements de table de processus, le contrôle <code>max-processes</code> offre une protection contre les zombies qui épuisent la table de processus. Les processus zombie ne possédant, par définition, aucun LWP, le	Quantité (emplacements de table de processus)

Nom de la commande	Description	Unité par défaut
	contrôle max-lwps n'offre aucune protection contre cette possibilité.	
project.max-sem-ids	Nombre maximum d'ID de sémaphore autorisé pour ce projet.	Quantité (ID de sémaphore)
project.max-shm-ids	Nombre maximum d'ID de mémoire partagée autorisé pour ce projet.	Quantité (ID de mémoire partagée)
project.max-shm-memory	Quantité totale de mémoire partagée System V autorisée pour ce projet.	Taille (octets)
project.max-lwps	Nombre maximum de LWP accessibles simultanément par ce projet.	Quantité (LWP)
project.max-tasks	Nombre maximum de tâches autorisé dans ce projet.	Quantité (nombre de tâches)
project.max-contracts	Nombre maximum de contrats autorisé dans ce projet.	Quantité (contrats)
task.max-cpu-time	Temps CPU maximum disponible pour les processus de cette tâche.	Temps (secondes)
task.max-lwps	Nombre maximum de LWP accessibles simultanément par les processus de cette tâche.	Quantité (LWP)
task.max-processes	Nombre maximum d'emplacements de table de processus simultanément disponibles pour les processus de cette tâche.	Quantité (emplacements de table de processus)
process.max-cpu-time	Temps CPU maximum disponible pour ce processus.	Temps (secondes)
process.max-file-descriptor	Index de descripteur de fichier maximum disponible pour ce processus.	Index (descripteur de fichier maximum)
process.max-file-size	Décalage de fichier maximum accessible en écriture par ce processus.	Taille (octets)
process.max-core-size	Taille maximum d'un dump noyau créé par ce processus.	Taille (octets)
process.max-data-size	Mémoire du tas maximum disponible pour ce processus.	Taille (octets)
process.max-stack-size	Segment de mémoire du tas maximum disponible pour ce processus.	Taille (octets)
process.max-address-space	Quantité d'espace d'adressage maximum (résultant de la somme des tailles de segment), disponible pour ce processus.	Taille (octets)
process.max-port-events	Nombre maximum d'événements autorisé par port pair.	Quantité (nombre d'événements)
process.max-sem-nsems	Nombre maximum de sémaphores autorisé par jeu de sémaphores.	Quantité (sémaphores par jeu)
process.max-sem-ops	Nombre maximum d'opérations de sémaphore autorisé par appel semop (valeur copiée à partir du contrôle de ressource au moment semget()).	Quantité (nombre d'opérations)
process.max-msg-qbytes	Nombre maximum d'octets de messages dans une file d'attente de messages (valeur	Taille (octets)

Nom de la commande	Description	Unité par défaut
	copiée à partir du contrôle de ressource au moment <code>msgget()</code> .	
<code>process.max-msg-messages</code>	Nombre maximum de messages dans une file d'attente de messages (valeur copiée à partir du contrôle de ressource au moment <code>msgget()</code>).	Quantité (nombre de messages)

Vous pouvez afficher les valeurs par défaut des contrôles de ressources sur un système pour lequel aucun contrôle de ressource n'a été défini ou modifié. Un tel système contient les entrées autres que celles par défaut dans le fichier `/etc/system` ou dans la base de données `project`. Pour afficher les valeurs, servez-vous de la commande `prctl`.

Contrôles des ressources à l'échelle d'une zone

L'intérêt de ces contrôles est de limiter l'utilisation des ressources totales pour l'ensemble des entités processus à l'intérieur d'une zone. Les contrôles de ressources à l'échelle d'une zone peuvent également être définis à l'aide des noms de propriétés globales comme décrit dans la section “ [Composants des zones](#) ” du manuel “ [Présentation d'Oracle Solaris Zones](#) ”.

TABLEAU 6-2 Contrôles de ressources des zones

Nom de la commande	Description	Unité par défaut
<code>zone.cpu-cap</code>	Quantité absolue des ressources de la CPU pouvant être consommée par une zone non globale. La valeur 100 représente 100 % d'une CPU pour le paramètre <code>project.cpu-cap</code> . La valeur 125 équivaut à 125 %, car 100 % correspond à une capacité de CPU complète sur le système.	Quantité (nombre de CPU)
<code>zone.cpu-shares</code>	Nombre de partages CPU de l'ordonnanceur FSS pour cette zone.	Quantité (partages)
<code>zone.max-lofi</code>	Nombre maximum de périphériques <code>lofi</code> qui peuvent être créés par une zone. La valeur permet de limiter l'utilisation de l'espace de noms de noeud inférieur de chaque zone.	Quantité (nombre de périphériques <code>lofi</code>)
<code>zone.max-locked-memory</code>	Quantité totale de mémoire physique verrouillée accessible par une zone. Lorsque le contrôle <code>priv_proc_lock_memory</code> est appliqué à une zone, pensez également à définir ce contrôle de ressource pour empêcher cette zone de verrouiller toute la mémoire.	Taille (octets)

Nom de la commande	Description	Unité par défaut
zone.max-lwps	Nombre maximum de LWP accessibles simultanément par cette zone.	Quantité (LWP)
zone.max-msg-ids	Nombre maximum d'ID de file d'attente des messages autorisé pour cette zone.	Quantité (ID de file d'attente des messages)
zone.max-processes	Nombre maximum d'emplacements de table de processus simultanément disponibles pour cette zone. Dans la mesure où les processus standard et zombie utilisent tous deux les emplacements de table de processus, le contrôle max-processes offre une protection contre les zombies qui épuisent la table de processus. Les processus zombie ne possédant, par définition, aucun LWP, le contrôle max-lwps n'offre aucune protection contre cette possibilité.	Quantité (emplacements de table de processus)
zone.max-sem-ids	Nombre maximum d'ID de sémaphore autorisé pour cette zone.	Quantité (ID de sémaphore)
zone.max-shm-ids	Nombre maximum d'ID de mémoire partagée autorisé pour cette zone.	Quantité (ID de mémoire partagée)
zone.max-shm-memory	Quantité totale de mémoire partagée System V autorisée pour cette zone.	Taille (octets)
zone.max-swap	Quantité totale de swap utilisable par les mappages d'espace d'adressage des processus utilisateur et les montages tmpfs pour cette zone.	Taille (octets)

Pour plus d'informations sur la configuration des contrôles de ressources à l'échelle d'une zone, reportez-vous à la section [“Configuration des contrôles de ressources et des attributs”](#) à la page 61 et à la section [“Paramétrage des contrôles de ressources à l'échelle de la zone”](#) du manuel [“Présentation d'Oracle Solaris Zones”](#).

Notez qu'il est possible d'appliquer à la zone globale un contrôle de ressource à l'échelle d'une zone.

Prise en charge des unités

Les indicateurs globaux permettant d'identifier les types de contrôles de ressources sont définis pour l'ensemble des contrôles de ressources. Le système utilise ces indicateurs pour communiquer des informations de type de base aux applications telles que la commande `prctl`. Les applications se servent de ces informations pour déterminer :

- Les chaînes d'unité qui conviennent à chaque contrôle de ressource
- L'échelle à utiliser lors de l'interprétation des valeurs d'échelle

Voici l'ensemble des indicateurs globaux disponibles :

Indicateur global	Chaîne du type de contrôle de ressource	Modificateur	Echelle
RCTL_GLOBAL_BYTES	octets	octets	1
		Ko	2^{10}
		Mo	2^{20}
		Go	2^{30}
		To	2^{40}
		Po	2^{50}
		Eo	2^{60}
RCTL_GLOBAL_SECONDS	secondes	s	1
		Ks	10^3
		Ms	10^6
		Gs	10^9
		Ts	10^{12}
		Ps	10^{15}
		Es	10^{18}
RCTL_GLOBAL_COUNT	nombre	aucun	1
		K	10^3
		M	10^6
		G	10^9
		T	10^{12}
		P	10^{15}
		E	10^{18}

Les valeurs d'échelle peuvent être employées avec des contrôles de ressources. L'exemple suivant illustre une valeur de seuil à l'échelle :

```
task.max-lwps=(priv,1K,deny)
```

Remarque - Les modificateurs d'unité sont acceptés par les commandes `prctl`, `projadd`, `etprojmod`. Vous ne pouvez pas utiliser des modificateurs d'unité dans la base de données `project` elle-même.

Valeurs des contrôles de ressources et niveaux de privilège

Une valeur de seuil appliquée à un contrôle de ressource constitue le seuil de déclenchement d'actions locales ou de mise en oeuvre d'actions globales telles que la consignation.

Chaque valeur de seuil d'un contrôle de ressource doit être associé à un niveau de privilège. Trois niveaux de privilège sont autorisés.

- De base (niveau modifiable par le propriétaire du processus d'appel)
- Privilégié (niveau modifiable uniquement par des appelants privilégiés ou root)
- Système (niveau fixé pour toute la durée de l'instance du système d'exploitation)

A chaque contrôle de ressource correspond une seule et même valeur système définie par le système ou par le fournisseur de ressource. La valeur système représente la quantité de ressource susceptible d'être fournie par l'implémentation actuelle du système d'exploitation.

Il est possible de définir un nombre quelconque de valeurs privilégiées, mais une seule valeur de base est autorisée. Un privilège de base par défaut est assigné aux opérations pour lesquelles aucune valeur privilégiée n'a été prévue.

Le niveau de privilège pour une valeur de contrôle de ressource est défini dans le champ correspondant du bloc de contrôle de ressource sous la forme `RCTL_BASIC`, `RCTL_PRIVILEGED` ou `RCTL_SYSTEM`. Pour plus d'informations, reportez-vous à [setrctl\(2\)](#). Vous pouvez faire appel à la commande `prctl` pour modifier les valeurs associées au niveau de base et au niveau privilégié.

Actions globales et locales applicables aux valeurs de contrôle de ressource

Il existe deux catégories d'actions applicables aux valeurs de contrôle de ressource : globale et locale.

Actions globales applicables aux valeurs de contrôle de ressource

Les actions globales s'appliquent aux valeurs de chaque contrôle de ressource du système. Vous pouvez utiliser la commande `rctladm` décrite à la page de manuel [rctladm\(1M\)](#) pour réaliser les actions suivantes :

- Afficher l'état global des contrôles de ressources système actifs
- Définir les actions de consignation globales

Vous avez la possibilité d'activer ou de désactiver l'action de consignation globale pour les contrôles de ressources. Vous pouvez configurer l'action `syslog` jusqu'à un degré spécifique en assignant un niveau de gravité, `syslog=level`. Voici les différents paramètres sélectionnables pour `level` :

- `debug`
- `info`

- notice
- warning
- err
- crit
- alert
- emerg

Par défaut, les violations de contrôle de ressource ne font pas l'objet d'une consignation globale. Le niveau n/a indique les contrôles de ressources sur lesquels aucune action globale ne peut être configurée.

Actions locales applicables aux valeurs de contrôle de ressource

Des actions locales sont accomplies sur un processus essayant de dépasser la valeur de contrôle. Il est possible d'associer une ou plusieurs actions à chaque valeur de seuil appliquée à un contrôle de ressource. Il existe trois types d'action locale : none, deny et signal=. Ces trois actions sont utilisées dans les conditions suivantes :

none	Les demandes de ressources d'une quantité supérieure au seuil fixé ne sont suivies d'aucune action. Cela est pratique pour surveiller l'utilisation des ressources sans perturber le déroulement des applications en cours. Vous pouvez également prévoir l'affichage d'un message global en cas de dépassement du contrôle de ressource, même si cela n'a aucune incidence sur le processus concerné.
deny	Les demandes de ressources d'une quantité supérieure au seuil fixé sont refusées. Ainsi, un contrôle de ressource <code>task.max-lwps</code> pour lequel vous avez choisi l'action deny provoque l'échec de l'appel système <code>fork</code> si le nouveau processus dépasse la valeur de contrôle. Reportez-vous à la page de manuel fork(2) .
signal=	Vous pouvez demander l'émission d'un signal en cas de dépassement du contrôle de ressource. Le signal est alors transmis au processus. Aucun autre signal n'est envoyé si le processus utilise des ressources supplémentaires. Les signaux disponibles sont répertoriés dans le Tableau 6-3, "Signaux disponibles pour les valeurs des contrôles de ressources" .

Ces actions ne sont pas toutes applicables à chaque contrôle de ressource. Un processus n'est pas en mesure, par exemple, de dépasser le nombre de parts de CPU allouées au projet dont il est membre. L'action deny n'est, par conséquent, pas applicable au contrôle de ressource `project.cpu-shares`.

En raison des restrictions liées à l'implémentation, les propriétés globales de chaque contrôle peuvent limiter le champ d'actions programmables pour la valeur de seuil. (Reportez-vous à la page de manuel [rctladm\(1M\)](#).) Le tableau suivant récapitule les actions ayant pour effet d'émettre un signal. Pour plus d'informations au sujet des signaux, voir la page de manuel [signal\(3HEAD\)](#).

TABEAU 6-3 Signaux disponibles pour les valeurs des contrôles de ressources

Signal	Description	Remarques
SIGABRT	Met fin au processus.	
SIGHUP	Envoie un signal de déconnexion. Cela se produit en cas d'abandon de la porteuse sur une ligne. Le signal est transmis au groupe de processus qui contrôle le terminal.	
SIGTERM	Met fin au processus. Signal d'interruption envoyé par le logiciel.	
SIGKILL	Met fin au processus et interrompt le programme.	
SIGSTOP	Arrête le processus. Signal de contrôle du travail.	
SIGXRES	Signal envoyé en cas de dépassement de la limite de contrôle de ressource. Généré par l'utilitaire de contrôle des ressources.	
SIGXFSZ	Met fin au processus. Signal envoyé en cas de dépassement de la limite de taille de fichier.	Concerne uniquement les contrôles de ressources possédant la propriété <code>RCTL_GLOBAL_FILE_SIZE</code> (<code>process.max-file-size</code>). Pour plus d'informations, voir rctlblk_set_value(3C) .
SIGXCPU	Met fin au processus. Signal envoyé en cas de dépassement de la limite du temps CPU.	Concerne uniquement les contrôles de ressources possédant la propriété <code>RCTL_GLOBAL_CPU_TIME</code> (<code>process.max-cpu-time</code>). Pour plus d'informations, voir rctlblk_set_value(3C) .

Indicateurs et propriétés des contrôles de ressources

Chaque contrôle de ressource sur le système est associé à un jeu de propriétés bien précis. Ce jeu de propriétés correspond à un ensemble d'indicateurs auxquels sont associés à toutes les instances gérées de cette ressource. Vous ne pouvez pas modifier les indicateurs globaux, mais il est possible de récupérer les indicateurs à l'aide de l'appel système `rctladm` ou `getrctl`.

Les indicateurs locaux définissent le comportement et la configuration par défaut pour une valeur de seuil donnée du contrôle de ressource appliqué à un processus particulier ou à un ensemble de processus. Les indicateurs locaux prévus pour une valeur de seuil n'ont pas

d'incidence sur le comportement des autres valeurs de seuil définies pour le même contrôle de ressource. En revanche, les indicateurs globaux changent le comportement de chacune des valeurs associées à un contrôle particulier. Il est possible de modifier les indicateurs locaux (en respectant les contraintes fournies par les indicateurs globaux correspondants) au moyen de la commande `prctl` ou de l'appel système `setrctl`. Voir [setrctl\(2\)](#).

Pour obtenir la liste complète des indicateurs locaux, des indicateurs globaux et de leurs définitions, voir la page de manuel [rctlblk_set_value\(3C\)](#).

Pour déterminer le comportement du système lorsqu'une valeur de seuil d'un contrôle de ressource est atteinte, exécutez la commande `rctladm` pour afficher les indicateurs globaux du contrôle de ressource qui vous intéresse. Pour afficher, par exemple, les valeurs de `process.max-cpu-time`, entrez l'instruction suivante :

```
$ rctladm process.max-cpu-time
process.max-cpu-time syslog=off [ lowerable no-deny cpu-time inf seconds ]
```

Les indicateurs globaux ont la signification suivante.

lowerable	Il n'est pas utile de posséder des privilèges de superutilisateur pour abaisser les valeurs privilégiées pour ce contrôle.
no-deny	Même en cas de dépassement des valeurs de seuil, l'accès à la ressource n'est jamais refusé.
cpu-time	Le signal SIGXCPU peut être envoyé lorsque les valeurs de seuil de cette ressource sont atteintes.
seconds	Valeur temporelle pour le contrôle de ressource.
no-basic	Il est impossible de définir une valeur de contrôle de ressource avec un privilège de type basic. Seules les valeurs de privilège élevé sont autorisées.
no-signal	Il est impossible de définir une action locale de signal pour les valeurs de contrôle de ressource.
no-syslog	L'action globale de message sys log ne peut pas être définie pour ce contrôle de ressource.
deny	Les demandes de ressources sont refusées dès lors que les valeurs de seuil sont atteintes.
count	Valeur numérique (entier) pour le contrôle de ressource.
bytes	Unité de taille pour le contrôle de ressource.

Servez-vous de la commande `prctl` pour afficher les valeurs et actions locales pour le contrôle de ressource.

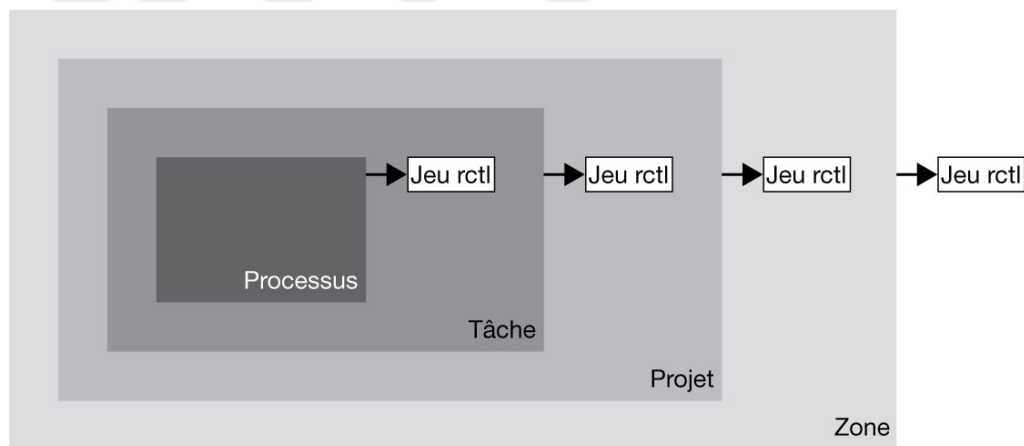
```
$ prctl -n process.max-cpu-time $$
process 353939: -ksh
NAME    PRIVILEGE  VALUE    FLAG    ACTION    RECIPIENT
process.max-cpu-time
  privileged  18.4Es    inf     signal=XCPU    -
  system      18.4Es    inf     none
```

L'indicateur `max` (`RCTL_LOCAL_MAXIMAL`) est défini pour les deux valeurs de seuil et l'indicateur `inf` (`RCTL_GLOBAL_INFINITE`) s'applique à ce contrôle de ressource. Une valeur `inf` représente une quantité infinie. La valeur n'est jamais imposée. Telles qu'elles sont configurées, les deux quantités limites représentent des valeurs infinies qui ne sont jamais dépassées.

Mise en oeuvre du contrôle de ressources

Il est possible d'appliquer plusieurs contrôles de ressources à une même ressource. Vous pouvez définir un contrôle de ressource à chaque niveau d'imbrication dans le modèle de processus. Si des contrôles de ressources sont appliqués à la même ressource à différents niveaux d'imbrication, c'est le contrôle du plus petit conteneur qui est mis en oeuvre en premier. Ainsi, l'action serait appliquée à `process.max-cpu-time` avant d'être appliquée à `task.max-cpu-time` si les deux contrôles sont découverts en même temps.

FIGURE 6-1 Collections de processus, relations entre conteneurs et jeux de contrôles de ressources correspondants



Surveillance globale des événements de contrôle de ressource

La quantité de ressources utilisée par les processus est bien souvent inconnue. Pour obtenir des informations supplémentaires, essayez d'utiliser les actions globales de contrôle de ressource auxquelles la commande `rctladm` donne accès. Exécutez la commande `rctladm` pour établir une action `syslog` sur un contrôle de ressource. Si une entité gérée par ce contrôle de ressource rencontre une valeur de seuil, un message système est consigné dans le journal au niveau de consignation configuré. Pour plus d'informations, reportez-vous au [Chapitre 7, Administration des contrôles des ressources \(tâches\)](#) et à la page de manuel `rctladm(1M)`.

Application des contrôles de ressources

Chaque contrôle de ressource répertorié dans le [Tableau 6-1, “Contrôles de ressources standard de projet, de tâche et de processus”](#) peut être assigné à un projet au moment de la connexion ou lors de l'appel de la commande `newtask`, `su`, ou des autres lanceurs `at`, `batch`, ou `cron`. Chaque commande démarrée est exécutée dans une tâche distincte avec le projet par défaut de l'utilisateur à l'origine de l'appel. Pour plus d'informations, reportez-vous aux pages de manuel `login(1)`, `newtask(1)`, `at(1)`, `cron(1M)`, et `su(1M)`.

Les mises à jour des entrées dans la base de données `project`, apportées au fichier `/etc/project` ou à une représentation de la base de données dans un service de noms du réseau, ne sont pas appliquées aux projets actuellement actifs. Elles le sont dès lors qu'une nouvelle tâche rejoint le projet via une connexion ou la commande `newtask`.

Mise à jour temporaire des valeurs de contrôle de ressource sur un système en cours d'exécution

Les valeurs modifiées dans la base de données `project` prennent effet uniquement pour les nouvelles tâches démarrées dans un projet. Vous pouvez, cependant, vous servir des commandes `rctladm` et `prctl` pour actualiser les contrôles de ressources sur un système en cours d'exécution.

Mise à jour de l'état de la consignation

La commande `rctladm` a une incidence sur l'état de consignation global de chaque contrôle de ressource à l'échelle d'un système. Elle est pratique pour afficher l'état global et configurer le niveau de consignation de `syslog` en cas de dépassement des contrôles.

Mise à jour des contrôles de ressources

Il est possible d'afficher et de modifier temporairement les valeurs des contrôles de ressources et les actions par processus, par tâche ou par projet en utilisant la commande `prctl`. La commande, basée sur l'entrée fournie par un ID de projet, de tâche ou de processus, s'applique au contrôle de ressource au niveau où le contrôle a été défini.

Les modifications apportées aux valeurs et aux actions prennent effet immédiatement. Cependant, ces modifications s'appliquent uniquement au processus, à la tâche ou au projet en cours. Elles ne sont pas consignées dans la base de données `project`. Elles sont donc automatiquement perdues au redémarrage du système. Pour qu'ils aient un caractère permanent, les changements des contrôles de ressources doivent être enregistrés dans la base de données `project`.

Tous les paramètres des contrôles de ressource susceptibles d'être changés dans la base de données `project` peuvent également être modifiés avec la commande `prctl`. Il est possible d'ajouter ou de supprimer aussi bien les valeurs de base que les valeurs privilégiées. Leurs actions peuvent aussi être modifiées. Par défaut, le type de base s'applique à toutes les opérations définies, mais les processus et les utilisateurs dotés des privilèges `root` sont également à même de modifier les contrôles de ressources privilégiés. Il est impossible de changer, en revanche, les contrôles de ressources système.

Commandes utilisées avec les contrôles de ressources

Les commandes s'appliquant aux contrôles de ressources sont présentées dans le tableau suivant.

Référence des commandes	Description
ipcs(1)	Permet d'identifier les objets IPC contribuant à l'utilisation d'un projet.
prctl(1)	Permet de procéder à des interrogations d'exécution et à des modifications de l'utilitaire de contrôle des ressources, en fonction d'un champ d'application local.
rctladm(1M)	Permet de procéder à des interrogations d'exécution et à des modifications de l'utilitaire de contrôle des ressources, en fonction d'un champ d'application global.

La page de manuel [resource_controls\(5\)](#) décrit les contrôles de ressources disponibles par l'intermédiaire de la base de données `project`, y compris les unités et les facteurs d'échelle.

♦ ♦ ♦ CHAPITRE 7

Administration des contrôles des ressources (tâches)

Ce chapitre décrit comment gérer l'utilitaire de contrôle des ressources.

Pour avoir un aperçu de cet utilitaire des contrôles de ressources, reportez-vous au [Chapitre 6, A propos des contrôles de ressources](#).

Administration des contrôles des ressources (liste des tâches)

Tâche	Description	Pour obtenir des instructions
Définition de contrôles des ressources.	Définissez les contrôles des ressources pour un projet dans le fichier <code>/etc/project</code> .	“Définition des contrôles des ressources” à la page 76
Obtention ou révision des valeurs de contrôle des ressources pour les processus, tâches ou projets actifs au niveau local	Procédez à des interrogations d'exécution et à des modifications des contrôles des ressources associés à un processus, une tâche ou un projet actif sur le système.	“Utilisation de la commande <code>prctl</code>” à la page 78
Affichage ou mise à jour de l'état global des contrôles des ressources sur un système en cours de fonctionnement	Vérifiez l'état de connexion global de chaque contrôle de ressource à l'échelle du système. Configurez également le niveau de consignation <code>syslog</code> en cas de dépassement des contrôles.	“Utilisation de la commande <code>rctladm</code>” à la page 83
Détermination de l'état des utilitaires IPC (Interprocess Communication, communication interprocessus) actifs	Affichez des informations au sujet des utilitaires IPC actifs. Notez les objets IPC contribuant à l'utilisation d'un projet.	“Utilisation de la commande <code>ipcs</code>” à la page 84
Evaluation de la capacité CPU allouée à un serveur Web pour déterminer si elle est suffisante	Définissez une action globale pour un contrôle de ressource. Cette action permet d'être averti lorsque la valeur de contrôle de ressource d'une entité est trop faible.	“Détermination de la capacité CPU allouée à un serveur Web” à la page 85

Définition des contrôles des ressources

▼ Définition du nombre maximum de processus légers (LWP) pour chaque tâche d'un projet

Cette procédure a pour objet d'ajouter un projet intitulé x-files dans le fichier /etc/project et de fixer un nombre maximum de LWP pour une tâche créée dans le projet.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Exécutez la commande `projadd` avec l'option `-K` pour créer un projet nommé x-files. Définissez le nombre maximum de LWP (3) pour chaque tâche créée dans le projet.**

```
# projadd -K 'task.max-lwps=(privileged,3,deny)' x-files
```

3. **Vérifiez l'entrée dans le fichier /etc/project en utilisant l'une des méthodes suivantes :**

- Entrez :

```
# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
.
.
.
x-files
    projid : 100
    comment: ""
    users  : (none)
    groups : (none)
    attribs: task.max-lwps=(privileged,3,deny)
```

- Entrez :

```
# cat /etc/project
system:0:System:::
.
.
.
```

```
x-files:100::::task.max-lwps=(privileged,3,deny)
```

Exemple 7-1 Exemple de session

Après avoir effectué toutes les étapes de cette procédure, lorsque l'utilisateur root créera une tâche dans le projet x-files en rejoignant le projet à l'aide de la commande `newtask`, il ne sera pas possible de créer plus de trois nouveaux processus légers (LWP) pendant l'exécution de cette tâche. Cela est illustré dans la session annotée suivante.

```
# newtask -p x-files csh

# prctl -n task.max-lwps $$
process: 111107: csh
NAME  PRIVILEGE  VALUE  FLAG  ACTION  RECIPIENT
task.max-lwps
      usage      3
      privileged  3      -  deny
      system    2.15G  max  deny
# id -p
uid=0(root) gid=1(other) projid=100(x-files)

# ps -o project,taskid -p $$
PROJECT TASKID
x-files  73

# csh /* creates second LWP */

# csh /* creates third LWP */

# csh /* cannot create more LWPs */
Vfork failed
#
```

▼ Définition de plusieurs contrôles pour un projet

Le fichier `/etc/project` peut contenir plusieurs paramètres de contrôle de ressources pour chaque projet ainsi que diverses valeurs de seuil pour chaque contrôle. Les valeurs de seuils sont définies dans des clauses `action` et doivent être séparées par des virgules.

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Exécutez la commande `projmod` avec les options `-s` et `-k` pour appliquer les contrôles de ressources au projet x-files :

```
# projmod -s -K 'task.max-lwps=(basic,10,none),(privileged,500,deny);
process.max-file-descriptor=(basic,128,deny)' x-files one line in file
```

Les contrôles suivants sont définis :

- Contrôle basic sans action pour le nombre maximum de LWP par tâche.
- Contrôle deny privilégié quant au nombre maximum de LWP par tâche. Ce contrôle provoque l'échec de la création d'un LWP en cas de dépassement du nombre maximum autorisé, comme le montre l'exemple proposé dans la section [“Définition du nombre maximum de processus légers \(LWP\) pour chaque tâche d'un projet”](#) à la page 76.
- Nombre limité de descripteurs de fichier par processus au niveau basic, ce qui provoque l'échec d'un appel `open` en cas de dépassement du seuil maximum.

3. Vérifiez l'entrée dans le fichier en utilisant l'une des méthodes suivantes :

- Entrez :

```
# projects -l
.
.
.
x-files
    projid : 100
    comment: ""
    users  : (none)
    groups : (none)
    attrbs: process.max-file-descriptor=(basic,128,deny)
           task.max-lwps=(basic,10,none),(privileged,500,deny)    one line in file
```

- Entrez :

```
# cat /etc/project
.
.
.
x-files:100:::process.max-file-descriptor=(basic,128,deny);
task.max-lwps=(basic,10,none),(privileged,500,deny)    one line in file
```

Utilisation de la commande `prctl`

Servez-vous de la commande `prctl` pour procéder à des interrogations d'exécution et à des modifications des contrôles des ressources associés à un processus, une tâche ou un projet actif sur le système. Pour plus d'informations, reportez-vous à la page de manuel [prctl\(1\)](#).

▼ Affichage des valeurs de contrôle des ressources par défaut à l'aide de la commande `prctl`

Appliquez cette procédure à un système pour lequel aucun contrôle de ressource n'a été prévu ou modifié. Le fichier `/etc/system` et la base de données du projet ne peuvent contenir que des entrées non définies par défaut.

- Appliquez la commande `prctl` au processus de votre choix, par exemple le shell actuellement en cours d'exécution.

```
# prctl $$
process: 3320: bash
NAME      PRIVILEGE       VALUE    FLAG    ACTION                               RECIPIENT
process.max-port-events
  privileged 65.5K          -    deny                               -
  system    2.15G          max   deny                               -
process.max-msg-messages
  privileged 8.19K          -    deny                               -
  system    4.29G          max   deny                               -
process.max-msg-qbytes
  privileged 64.0KB         -    deny                               -
  system    16.0EB         max   deny                               -
process.max-sem-ops
  privileged 512            -    deny                               -
  system    2.15G          max   deny                               -
process.max-sem-nsems
  privileged 512            -    deny                               -
  system    32.8K          max   deny                               -
process.max-address-space
  privileged 16.0EB         max   deny                               -
  system    16.0EB         max   deny                               -
process.max-file-descriptor
  basic      256            -    deny                               3320
  privileged 65.5K          -    deny                               -
  system    2.15G          max   deny                               -
process.max-core-size
  privileged 8.00EB         max   deny                               -
  system    8.00EB         max   deny                               -
process.max-stack-size
  basic      10.0MB         -    deny                               3320
  privileged 32.0TB         -    deny                               -
  system    32.0TB         max   deny                               -
process.max-data-size
  privileged 16.0EB         max   deny                               -
  system    16.0EB         max   deny                               -
process.max-file-size
  privileged 8.00EB         max   deny,signal=XFSZ                 -
  system    8.00EB         max   deny                               -
process.max-cpu-time
  privileged 18.4Es         inf   signal=XCPU                       -
  system    18.4Es         inf   none                               -
```

task.max-cpu-time					
usage	0s				
system	18.4Es	inf	none		-
task.max-processes					
usage	2				
system	2.15G	max	deny		-
task.max-lwps					
usage	3				
system	2.15G	max	deny		-
project.max-contracts					
privileged	10.0K	-	deny		-
system	2.15G	max	deny		-
project.max-locked-memory					
usage	0B				
system	16.0EB	max	deny		-
project.max-port-ids					
privileged	8.19K	-	deny		-
system	65.5K	max	deny		-
project.max-shm-memory					
privileged	510MB	-	deny		-
system	16.0EB	max	deny		-
project.max-shm-ids					
privileged	128	-	deny		-
system	16.8M	max	deny		-
project.max-msg-ids					
privileged	128	-	deny		-
system	16.8M	max	deny		-
project.max-sem-ids					
privileged	128	-	deny		-
system	16.8M	max	deny		-
project.max-crypto-memory					
usage	0B				
privileged	510MB	-	deny		-
system	16.0EB	max	deny		-
project.max-tasks					
usage	2				
system	2.15G	max	deny		-
project.max-processes					
usage	4				
system	2.15G	max	deny		-
project.max-lwps					
usage	11				
system	2.15G	max	deny		-
project.cpu-cap					
usage	0				
system	4.29G	inf	deny		-
project.cpu-shares					
usage	1				
privileged	1	-	none		-
system	65.5K	max	none		-
zone.max-lofi					
usage	0				
system	18.4E	max	deny		-
zone.max-swap					

usage	180MB				
system	16.0EB	max	deny		-
zone.max-locked-memory					
usage	0B				
system	16.0EB	max	deny		-
zone.max-shm-memory					
system	16.0EB	max	deny		-
zone.max-shm-ids					
system	16.8M	max	deny		-
zone.max-sem-ids					
system	16.8M	max	deny		-
zone.max-msg-ids					
system	16.8M	max	deny		-
zone.max-processes					
usage	73				
system	2.15G	max	deny		-
zone.max-lwps					
usage	384				
system	2.15G	max	deny		-
zone.cpu-cap					
usage	0				
system	4.29G	inf	deny		-
zone.cpu-shares					
usage	1				
privileged	1	-	none		-
system	65.5K	max	none		

▼ Affichage des informations relatives à un contrôle de ressource précis à l'aide de la commande `prctl`

- Affichez le nombre maximum de descripteurs de fichier pour le shell actuellement en cours d'exécution.

```
# prctl -n process.max-file-descriptor $$
process: 110453: -sh
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
process.max-file-descriptor
basic           256        -    deny    11731
privileged    65.5K        -    deny    -
system        2.15G       max    deny
```

▼ Modification temporaire d'une valeur à l'aide de la commande `prctl`

Cet exemple de procédure fait appel à la commande `prctl` pour ajouter temporairement une nouvelle valeur privilégiée afin d'interdire l'utilisation de plus de trois LWP par projet pour

le projet `x-files`. Le résultat est comparable à celui obtenu dans la section [“Définition du nombre maximum de processus légers \(LWP\) pour chaque tâche d'un projet”](#) à la page 76.

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.
2. Utilisez la commande `newtask` pour rejoindre le projet `x-files`.
3. Exécutez la commande `id` avec l'option `-p` pour vous assurer que le projet rejoint est celui qui convient.

```
# newtask -p x-files
```

```
# id -p
uid=0(root) gid=1(other) projid=101(x-files)
```

4. Ajoutez une nouvelle valeur privilégiée pour `project.max-lwps` afin de limiter le nombre de LWP à trois.

```
# prctl -n project.max-lwps -t privileged -v 3 -e deny -i project x-files
```

5. Vérifiez le résultat obtenu.

```
# prctl -n project.max-lwps -i project x-files
process: 111108: csh
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
project.max-lwps
usage          203
privileged    1000        -    deny    -
system        2.15G      max    deny    -
```

▼ Réduction de la valeur de contrôle des ressources à l'aide de la commande `prctl`

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.
2. Exécutez la commande `prctl` avec l'option `-r` pour changer la valeur minimale du contrôle de ressource `process.max-file-descriptor`.

```
# prctl -n process.max-file-descriptor -r -v 128 $$
```

▼ Affichage, remplacement et vérification de la valeur d'un contrôle dans un projet à l'aide de la commande `prctl`

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.
2. Vérifiez la valeur de `project.cpu-shares` dans le projet `group.staff`.

```
# prctl -n project.cpu-shares -i project group.staff
project: 2: group.staff
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
project.cpu-shares
  usage          1
  privileged      1      -    none      -
  system        65.5K    max    none
```

3. Remplacez la valeur actuelle 1 de `project.cpu-shares` par la valeur 10.

```
# prctl -n project.cpu-shares -v 10 -r -i project group.staff
```

4. Vérifiez la valeur de `project.cpu-shares` dans le projet `group.staff`.

```
# prctl -n project.cpu-shares -i project group.staff
project: 2: group.staff
NAME    PRIVILEGE    VALUE    FLAG    ACTION    RECIPIENT
project.cpu-shares
  usage          1
  privileged      1      -    none      -
  system        65.5K    max    none
```

Utilisation de la commande `rctladm`

Utilisation de la commande `rctladm`

Servez-vous de la commande `rctladm` pour procéder à des interrogations d'exécution et à des modifications de l'état global de l'utilitaire de contrôle des ressources. Pour plus d'informations, reportez-vous à la page de manuel [rctladm\(1M\)](#).

Vous pouvez, par exemple, utiliser la commande `rctladm` avec l'option `-e` pour activer l'attribut `syslog` global d'un contrôle de ressource. En cas de dépassement du contrôle, une notification est consignée dans le journal au niveau `syslog` spécifié. Pour activer l'attribut `syslog` global de `process.max-file-descriptor`, entrez l'instruction suivante :

```
# rctladm -e syslog process.max-file-descriptor
```

En l'absence d'argument, la commande `rctladm` affiche les indicateurs globaux, y compris l'indicateur de type global, pour chaque contrôle de ressource.

```
# rctladm
process.max-port-events      syslog=off [ deny count ]
process.max-msg-messages    syslog=off [ deny count ]
process.max-msg-qbytes       syslog=off [ deny bytes ]
process.max-sem-ops          syslog=off [ deny count ]
process.max-sem-nsems        syslog=off [ deny count ]
process.max-address-space    syslog=off [ lowerable deny no-signal bytes ]
process.max-file-descriptor  syslog=off [ lowerable deny count ]
process.max-core-size        syslog=off [ lowerable deny no-signal bytes ]
process.max-stack-size       syslog=off [ lowerable deny no-signal bytes ]
.
.
.
```

Utilisation de la commande `ipcs`

Utilisation de la commande `ipcs`

Servez-vous de la commande `ipcs` pour afficher des informations au sujet des utilitaires IPC actifs. Pour plus d'informations, reportez-vous à la page de manuel [ipcs\(1\)](#).

Vous pouvez exécuter la commande `ipcs` avec l'option `-J` pour connaître la limite d'un projet prévue pour un objet IPC.

```
# ipcs -J
IPC status from <running system> as of Wed Mar 26 18:53:15 PDT 2003
T      ID      KEY      MODE      OWNER      GROUP      PROJECT
Message Queues:
Shared Memory:
m      3600     0      --rw-rw-rw-  uname      staff      x-files
m      201      0      --rw-rw-rw-  uname      staff      x-files
m      1802     0      --rw-rw-rw-  uname      staff      x-files
m      503      0      --rw-rw-rw-  uname      staff      x-files
m      304      0      --rw-rw-rw-  uname      staff      x-files
m      605      0      --rw-rw-rw-  uname      staff      x-files
m      6        0      --rw-rw-rw-  uname      staff      x-files
m      107     0      --rw-rw-rw-  uname      staff      x-files
Semaphores:
s      0        0      --rw-rw-rw-  uname      staff      x-files
```

Avertissements relatifs à la capacité

Une action globale appliquée à un contrôle de ressource vous permet d'être averti lorsqu'une entité entre en conflit avec une valeur de contrôle de ressource trop réduite.

Supposons, par exemple, que vous souhaitiez déterminer si un serveur Web dispose d'une capacité CPU suffisante pour assurer sa charge de travail habituelle. Vous pourriez analyser les données `sar` correspondant au temps d'inactivité CPU et à la charge moyenne. Vous pourriez également examiner les données de comptabilisation étendue afin de déterminer le nombre de processus simultanés exécutés pour le processus serveur Web.

Le plus simple, cependant, serait d'intégrer le serveur Web à une tâche. Il suffirait, ensuite, de définir une action globale à l'aide de la commande `syslog`, afin d'être averti si une tâche dépasse le nombre de LWP programmé qui convient aux capacités de la machine.

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

▼ Détermination de la capacité CPU allouée à un serveur Web

1. **Servez-vous de la commande `prctl` pour intégrer un contrôle de ressource privilégié (possédé par un utilisateur `root`) aux tâches contenant un processus `httpd`. Limitez à 40 le nombre total de LWP de chaque tâche et désactivez toutes les actions locales.**

```
# prctl -n task.max-lwps -v 40 -t privileged -d all `pgrep httpd`
```

2. **Appliquez une action globale de consignation système au contrôle de ressource `task.max-lwps`.**

```
# rctladm -e syslog task.max-lwps
```

3. **Observez si la charge de travail perturbe le contrôle de ressource.**

Le cas échéant, vous obtenez des messages `/var/adm/messages` de type :

```
Jan  8 10:15:15 testmachine unix: [ID 859581 kern.notice]  
NOTICE: privileged rctl task.max-lwps exceeded by task 19
```

BETA
DRAFT

8

♦ ♦ ♦ C H A P I T R E 8

A propos de l'ordonnanceur de partage équitable

L'analyse des données de la charge de travail permet parfois d'identifier la charge de travail ou le groupe de charges de travail qui monopolise les ressources de la CPU. S'il n'existe aucune violation de contrainte en termes d'utilisation de la CPU, vous pouvez changer la stratégie d'allocation du temps CPU sur le système. La classe de programmation FSS (Fair Share Scheduler) décrite dans ce chapitre permet d'allouer du temps CPU en tenant compte des parts et non du plan de priorité de la classe de programmation TS (TimeSharing).

Ce chapitre se compose des sections suivantes :

- “Introduction à l'ordonnanceur FSS” à la page 87
- “Définition d'une part de CPU” à la page 88
- “Parts de la CPU et état des processus” à la page 89
- “Différence entre allocation des parts de CPU et utilisation de la CPU” à la page 89
- “Exemples de parts de CPU” à la page 89
- “Configuration de l'ordonnanceur FSS” à la page 94
- “Ordonnanceur FSS et jeux de processeurs” à la page 95
- “Utilisation de l'ordonnanceur FSS avec d'autres classes de programmation” à la page 98
- “Définition de la classe de programmation pour le système” à la page 99
- “Classe de programmation dans un système doté de zones” à la page 99
- “Commandes utilisées avec l'ordonnanceur FSS” à la page 99

Pour apprendre à utiliser l'ordonnanceur FSS, reportez-vous au [Chapitre 9, Administration de l'ordonnanceur FSS \(tâches\)](#).

Introduction à l'ordonnanceur FSS

L'une des tâches fondamentales du système d'exploitation consiste à décider à quels processus attribuer l'accès aux ressources système. Le programme chargé de la programmation des processus, appelé ordonnanceur ou dispatcheur, est la portion du noyau qui gère l'allocation des ressources de la CPU aux processus. L'ordonnanceur fonctionne sur le principe de classes de

programmation. Chaque classe définit une stratégie de programmation qui sert à planifier les processus au sein de la classe. L'ordonnanceur par défaut dans le système d'exploitation Oracle Solaris, l'ordonnanceur TS, essaie d'accorder à chaque processus un temps d'accès équivalent aux CPU disponibles. Il peut être souhaitable, cependant, d'allouer plus de ressources à certains processus qu'à d'autres.

Vous pouvez vous servir de l'*ordonnanceur FSS* pour contrôler la répartition des ressources disponibles des CPU entre les différentes charges de travail, en fonction de leur importance. Celle-ci se traduit par le nombre de *parts* de ressources CPU que vous assignez à chaque charge.

Assignez des parts de CPU à chacun des projets pour contrôler leur droit aux ressources CPU. L'ordonnanceur FSS garantit une répartition équitable des ressources CPU entre les projets en fonction des parts assignées, indépendamment du nombre de processus rattachés à un projet. Pour ce faire, il réduit les droits du projet en termes d'utilisation intensive de la CPU et augmente ses droits pour une utilisation légère, en conformité avec les autres projets.

L'ordonnanceur FSS se compose d'un module de classe de programmation de noyau et de versions spécifiques à la classe des commandes [dispadm\(1M\)](#) et [prioctl\(1\)](#). Les parts de projet utilisées par l'ordonnanceur FSS sont définies par le biais de la propriété `project.cpu-shares` dans la base de données [project\(4\)](#).

Remarque - Si vous utilisez le contrôle des ressources `project.cpu-shares` sur un système Oracle Solaris avec des zones installées, reportez-vous à la section “ [Paramétrage des contrôles de ressources à l'échelle de la zone](#) ” du manuel “ [Présentation d'Oracle Solaris Zones](#) ” et à la section “ [Utilisation de l'ordonnanceur FSS sur un système Oracle Solaris doté de zones](#) ” du manuel “ [Création et utilisation des zones Oracle Solaris](#) ”.

Définition d'une part de CPU

Le terme "part" désigne la partie des ressources de CPU système allouée à un projet. Si vous assignez un nombre de parts de CPU élevé à un projet, par rapport aux autres projets, l'ordonnanceur FSS alloue plus de ressources de CPU à ce projet.

Les parts de CPU ne doivent pas être considérées comme un pourcentage des ressources de CPU. Elles servent à définir l'importance relative d'une charge par rapport à une autre charge. Lorsque vous attribuez des parts de CPU à un projet, ce n'est pas tant le nombre qui est important, mais sa quantité respective par rapport aux autres projets. Il faut également prendre en compte le nombre de projets qui se "disputeront" les ressources de la CPU.

Remarque - Les processus de projets qui ne disposent d'aucune part ont la priorité système la plus basse (0). Ces processus sont uniquement exécutés lorsque les projets dotés de parts supérieures à zéro n'utilisent pas les ressources de CPU.

Parts de la CPU et état des processus

Dans le système Oracle Solaris, la charge de travail d'un projet implique, en principe, plusieurs processus. Du point de vue de l'ordonnanceur FSS, une charge de travail de projet est soit à l'état *passif*, soit à l'état *actif*. Un projet est considéré comme inactif lorsque aucun de ses processus n'utilise les ressources de la CPU. Cela signifie généralement que ces processus sont *en sommeil* (en attente d'exécution d'E/S) ou arrêtés. Un projet est considéré comme actif si au moins un de ses processus exploite les ressources de la CPU. La somme des parts de tous les projets actifs permet de calculer la portion des ressources de la CPU à attribuer aux projets.

Au fur et à mesure que des projets deviennent actifs, l'allocation de la CPU est réduite en conséquence pour chaque projet, mais la répartition entre les différents projets reste la même en proportion.

Différence entre allocation des parts de CPU et utilisation de la CPU

Il ne faut confondre allocation et utilisation des parts de CPU. L'utilisation moyenne de la CPU par un projet peut très bien avoisiner les 20 % même si 50 % des ressources de la CPU lui ont été alloués. En outre, les parts ont pour intérêt de limiter l'utilisation de la CPU uniquement en cas de compétition avec d'autres projets. Quelle que soit la part des ressources qui lui a été attribuée, un projet reçoit systématiquement 100 % de la puissance de traitement s'il est le seul projet en cours d'exécution sur le système. Les cycles de CPU disponibles ne sont jamais gaspillés. Ils sont répartis entre les projets.

L'allocation d'une petite part à une charge de travail intensive risque de réduire ses performances. Cela ne l'empêche pas, en revanche, de terminer ses opérations si le système n'est pas surchargé.

Exemples de parts de CPU

Supposons que vous disposiez d'un système avec deux CPU exécutant deux charges de travail en parallèle nommées respectivement A et B. Chaque charge de travail est exécutée dans un projet indépendant. Les projets ont été configurés de manière à allouer S_A parts au projet A, et S_B parts au projet B.

En moyenne, avec l'ordonnanceur TS habituel, chacune des charges de travail s'exécutant sur le système aurait droit à la même quantité de ressources de la CPU, c'est-à-dire 50 % de la capacité du système.

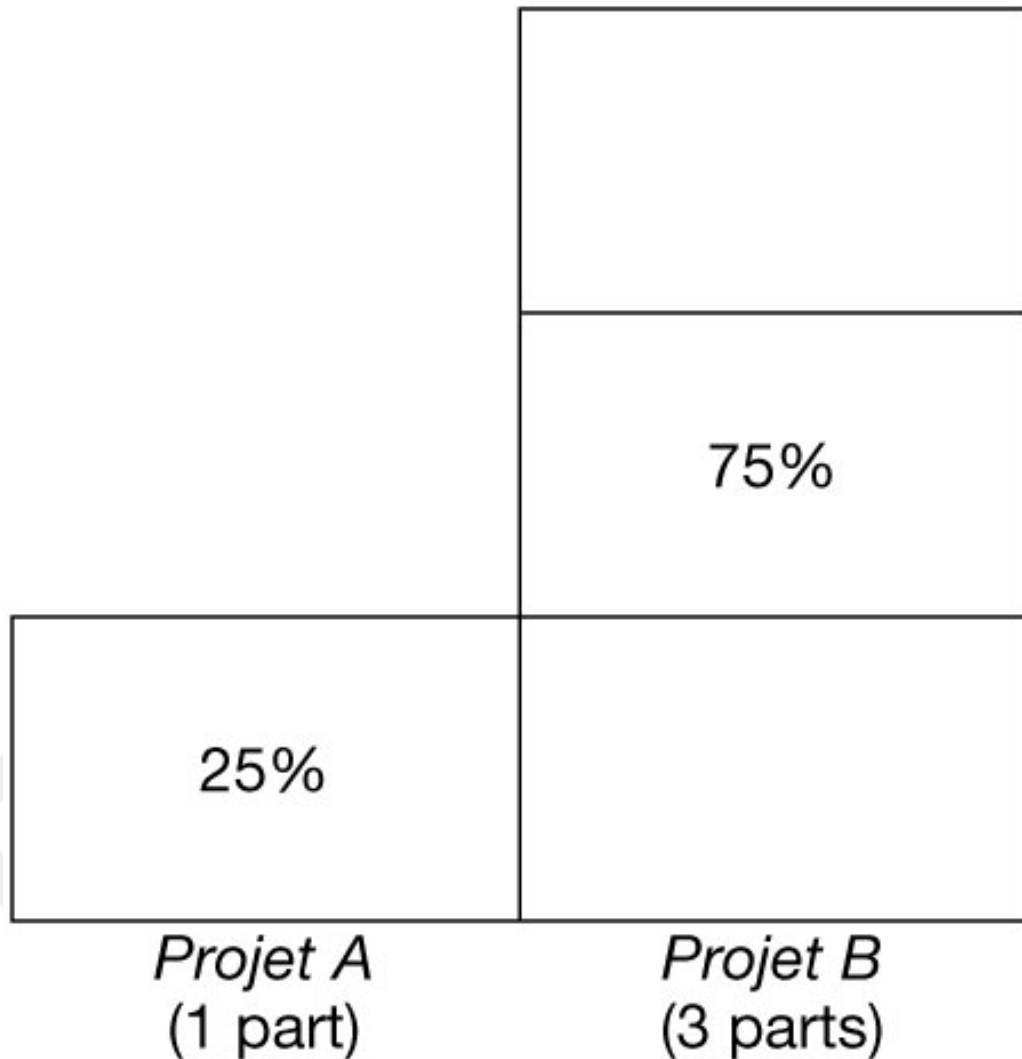
Lorsque vous les exécutez sous le contrôle de l'ordonnanceur FSS avec $S_A=S_B$, ces projets obtiennent approximativement les mêmes quantités de ressources de CPU. En revanche, si le

nombre de parts accordé à ces projets est différent, les allocations des ressources de la CPU ne seront pas identiques.

Les trois exemples qui suivent illustrent le principe de fonctionnement des parts dans différentes configurations. Ces exemples démontrent que le raisonnement mathématique des parts n'a de sens que si la demande atteint ou dépasse les ressources disponibles.

Exemple 1 : deux processus tirant parti de la CPU dans chaque projet

Si A et B disposent chacun de deux processus utilisant la CPU, et que $S_A = 1$ et $S_B = 3$, le nombre total de parts est alors $1 + 3 = 4$. Dans cette configuration, à condition que la demande de la CPU soit suffisante, 25 et 75 % des ressources de la CPU sont attribués respectivement aux projets A et B.



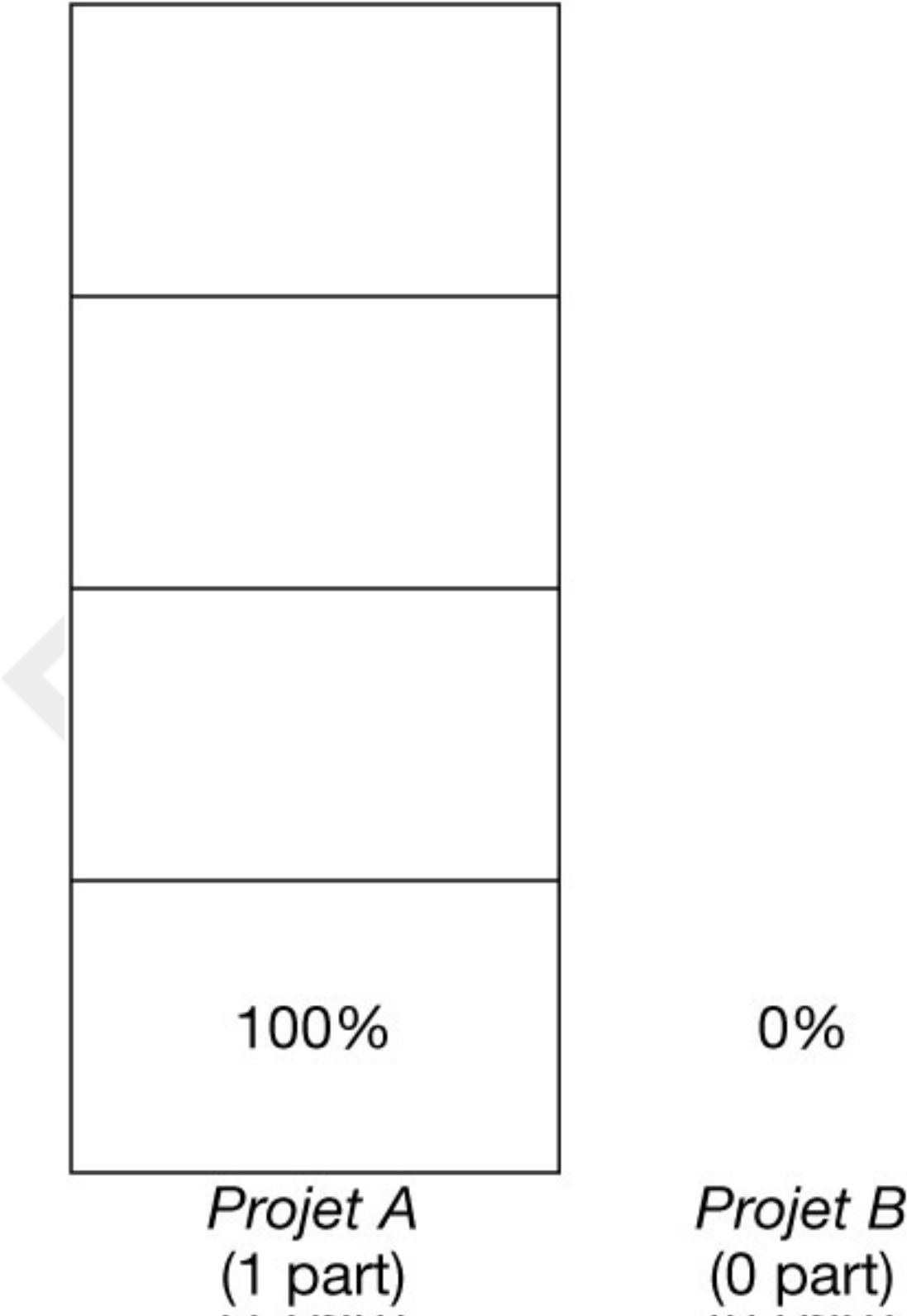
Exemple 2 : aucune compétition entre les projets

Si A et B ne disposent que d'un processus utilisant la CPU, et que $S_A = 1$ et $S_B = 100$, le nombre total de parts est alors de 101. Chaque projet ne peut utiliser plus d'une CPU car chacun n'a qu'un processus en cours d'exécution. Comme il n'existe aucun conflit de partage des ressources de la CPU dans cette configuration, les projets A et B se voient attribuer chacun 50 % de toutes les ressources de la CPU. Les parts de CPU n'ont donc aucune importance dans ce cas. Les allocations des projets seraient identiques (50/50), même si vous ne leur aviez attribué aucune part .

50%	50%
(1 ^{ère} CPU)	(2 ^e CPU)
<i>Projet A</i> (1 part)	<i>Projet B</i> (100 parts)

Exemple 3 : un projet dans l'incapacité de s'exécuter

Si A et B disposent chacun de deux processus ayant besoin de la puissance de traitement de la CPU et si 1 part est attribuée au projet A et 0 part au projet B, toutes les ressources sont allouées au projet A et aucune ressource au projet B. Les processus du projet B seront systématiquement définis au niveau de priorité 0, ce qui signifie qu'il sera impossible de les exécuter dans la mesure où le projet A possède en permanence le niveau de priorité supérieur.



Configuration de l'ordonnanceur FSS

Projets et utilisateurs

Les projets sont les conteneurs de charges de travail dans l'ordonnanceur FSS. Les groupes d'utilisateurs affectés à un projet sont traités comme de simples blocs contrôlables. Sachez qu'il est possible de créer un projet avec son propre nombre de parts pour chaque utilisateur.

Les utilisateurs peuvent faire partie de plusieurs projets possédant un nombre différent de parts. En déplaçant les processus d'un projet à un autre, il est possible de faire varier les quantités de ressources CPU qui leur sont allouées.

Pour plus d'informations sur la base de données et le service de noms `project(4)`, voir “[Base de données project](#)” à la page 23.

Configuration des parts de CPU

La configuration des parts de CPU est gérée par le service de noms en tant que propriété de la base de données `project`.

Lors de la création de la première tâche (ou du premier processus) associée à un projet au moyen de la fonction de bibliothèque `setproject(3PROJECT)`, le nombre de parts de CPU définies comme contrôle de ressource `project.cpu-shares` dans la base de données `project` est communiqué au noyau. Les projets pour lesquels aucun contrôle de ressource `project.cpu-shares` n'a été défini se voient attribuer une part.

Dans l'exemple qui suit, cette entrée du fichier `/etc/project` fixe le nombre de parts pour le projet `x-files` à 5 :

```
x-files:100:::project.cpu-shares=(privileged,5,none)
```

Si vous changez le nombre de parts de CPU alloué à un projet dans la base de données alors que des processus sont déjà en cours d'exécution, ce nombre n'est pas modifié à ce stade. Il est nécessaire, en effet, de redémarrer le projet pour que le changement prenne effet.

Si vous souhaitez modifier temporairement le nombre de parts attribué à un projet sans changer les attributs du projet dans la base de données `project`, utilisez la commande `prctl`. Par exemple, pour remplacer la valeur du contrôle des ressources `project.cpu-shares` du projet `x-files` par 3 lorsque les processus associés à ce projet sont exécutés, tapez ce qui suit :

```
# prctl -r -n project.cpu-shares -v 3 -i project x-files
```

Pour plus d'informations, reportez-vous à la page de manuel `prctl(1)`.

<code>-r</code>	Remplace la valeur actuelle pour le contrôle de ressource nommé.
<code>-n name</code>	Définit le nom du contrôle de ressource.
<code>-v val</code>	Spécifie la valeur du contrôle de ressource.
<code>-i typeid</code>	Précise le type d'ID de l'argument suivant.
<code>x-files</code>	Indique l'objet concerné par le changement. Dans cet exemple, il s'agit du projet <i>x-files</i> .

Le projet `system` avec pour ID de projet 0 comprend l'ensemble des démons système démarrés par les scripts d'initialisation lancés au démarrage. `system` peut être considéré comme un projet avec un nombre illimité de parts. Cela signifie que `system` est toujours programmé en premier, quel que soit le nombre de parts alloué aux autres projets. Si vous ne souhaitez pas accorder au projet `system` un nombre illimité de parts, il suffit de définir le nombre de parts de ce projet dans la base de données `project`.

Comme indiqué précédemment, les processus appartenant aux projets disposant d'aucune part possèdent la priorité système la plus basse (zéro). Les projets dotés d'une ou plusieurs parts s'exécutent aux niveaux de priorité un ou plus. Par conséquent, les projets dépourvus de part sont programmés à condition qu'aucun projet avec des parts ne demande de ressources de CPU.

Le nombre maximum de parts susceptible d'être alloué à un projet est de 65535.

Ordonnanceur FSS et jeux de processeurs

Il est possible d'utiliser l'ordonnanceur FSS conjointement aux jeux de processeurs. Cela permet de gérer les répartitions des ressources de la CPU entre les projets s'exécutant sur chaque jeu de processeurs de façon plus précise que si vous procédiez processeur par processeur. L'ordonnanceur FSS considère les jeux de processeurs comme des partitions parfaitement distinctes, chacun d'eux étant géré indépendamment des ressources de la CPU allouées.

Les allocations de la CPU prévues pour les projets s'exécutant sur un jeu de processeurs ne sont pas affectées par les parts de CPU ou l'activité des projets s'exécutant sur un autre jeu de processeurs, car les projets ne se disputent pas les mêmes ressources. Les projets sont en concurrence uniquement lorsqu'ils fonctionnent sur le même jeu de processeurs.

Le nombre de parts allouées à un projet s'applique à l'ensemble du système. Quel que soit le jeu de processeurs de destination, chaque partie d'un projet possède le même nombre de parts.

En cas d'utilisation de jeux de processeurs, les allocations des ressources de la CPU sont calculées pour les projets actifs au sein de chaque jeu.

Les partitions des projets s'exécutant sur divers jeux de processeurs risquent d'avoir des allocations de CPU différentes. L'allocation des ressources de la CPU pour chaque partition de projet dans un jeu de processeurs dépend uniquement des allocations définies pour les autres projets appartenant au même jeu.

Les performances et la disponibilité des applications s'exécutant dans les limites de leurs jeux de processeurs ne sont pas concernées par l'ajout de nouveaux jeux de processeurs. Les changements apportés aux parts de CPU des projets s'exécutant sur d'autres jeux de processeurs n'ont aucune incidence sur les applications.

Les jeux de processeurs vides (jeux sans processeur) ou les jeux de processeurs auxquels aucun processus n'est rattaché n'ont pas d'impact sur le comportement de l'ordonnanceur FSS.

Exemples d'utilisation des jeux de processeurs avec l'ordonnanceur FSS

Supposons qu'un serveur à huit CPU exécute plusieurs applications utilisant la CPU dans le cadre des projets *A*, *B*, et *C*. Une, deux et trois parts sont allouées respectivement aux projets *A*, *B* et *C*.

Le projet *A* n'est exécuté que sur le jeu de processeurs 1. Le projet *B* est exécuté sur les jeux de processeurs 1 et 2. Le projet *C* est exécuté sur les jeux de processeurs 1, 2 et 3. Présumons que chaque projet dispose de suffisamment de processus pour tirer parti de l'ensemble de la puissance disponible de la CPU. Dans un tel cas de figure, les projets sont en concurrence permanente pour utiliser les ressources de la CPU sur chaque jeu de processeurs.

Projet A 16.66% (1/6)	Projet B 40% (2/5)	Projet C 100% (3/3)
Projet B 33.33% (2/6)		
Projet C 50% (3/6)	Projet C 60% (3/5)	
Jeu de processeurs n° 1 2 CPU 25 % du système	Jeu de processeurs n° 2 4 CPU 50 % du système	Jeu de processeurs n° 3 2 CPU 25 % du système

La valeur totale est récapitulée dans le tableau suivant.

Projet	Allocation
Projet A	$4\% = (1/6 \times 2/8)_{\text{jeuproc1}}$
Projet B	$28\% = (2/6 \times 2/8)_{\text{jeuproc1}} + (2/5 \times 4/8)_{\text{jeuproc2}}$
Projet C	$67\% = (3/6 \times 2/8)_{\text{jeuproc1}} + (3/5 \times 4/8)_{\text{jeuproc2}} + (3/3 \times 2/8)_{\text{jeuproc3}}$

Ces pourcentages ne correspondent pas aux nombres de parts de CPU attribuées aux projets. Cependant, à l'intérieur de chaque jeu de processeurs, les rapports d'allocation de CPU par projet sont proportionnels à leurs parts respectives.

Sur le même système *sans* jeu de processeurs, la répartition des ressources de la CPU serait différente, comme le montre le tableau suivant.

Projet	Allocation
Projet A	$16,66\% = (1/6)$

Projet	Allocation
Projet B	33,33 % = (2/6)
Projet C	50 % = (3/6)

Utilisation de l'ordonnanceur FSS avec d'autres classes de programmation

Par défaut, la classe de programmation FSS utilise la même plage de priorités (0 à 59) que les classes de programmation TS (partage de temps), AI (interactive) et FX (priorité fixe). Il vaut donc mieux éviter que des processus de ces classes de programmation partagent *le même* jeu de processeurs. Le fait de combiner des processus des classes FSS, TS, AI et FX risquerait de provoquer un comportement inattendu.

En revanche, si vous utilisez des jeux de processeurs, il est possible de combiner des classes TS, AI et FX avec l'ordonnanceur FSS sur un même système. Tous les processus s'exécutant sur chaque jeu de processeurs doivent, cependant, faire partie d'*une seule* classe de programmation, pour éviter qu'ils entrent en compétition pour les mêmes CPU. Il faut veiller notamment à ne pas utiliser l'ordonnanceur FX en parallèle avec la classe de programmation FSS, sauf si vous tirez parti des jeux de processeurs. Cela empêche les applications de la classe FX d'utiliser les priorités les plus élevées au détriment des applications de la classe FSS.

Vous pouvez combiner des processus des classes TS et AI au sein du même jeu de processeurs ou sur le même système sans jeu de processeurs.

Le système Oracle Solaris propose également un ordonnanceur en temps réel (RT) aux utilisateurs dotés des privilèges root. Par défaut, la classe de programmation RT utilise les priorités système d'une plage différente (entre 100 et 159, en général) de celle de la classe FSS. Comme les classes RT et FSS utilisent des plages de priorités *distinctes* (sans risque de se chevaucher), il est possible de les faire cohabiter au sein du même jeu de processeurs. La classe de programmation FSS n'offre, cependant, aucun contrôle sur les processus s'exécutant dans la classe RT.

Sur un système quadriprocesseur, par exemple, un processus RT à simple thread peut accaparer un processeur entier si le processus est lié à la CPU. Si le système exécute également la classe FSS, des processus utilisateur standard se disputent les trois CPU restantes. Il faut noter que le processus RT n'utilise pas nécessairement la CPU en continu. Lorsqu'il est inactif, la classe FSS tire parti des quatre processeurs.

Vous pouvez saisir la commande suivante pour identifier les classes de programmation sur lesquelles s'exécutent les jeux de processeurs et vous assurer que chaque jeu de processeurs est configuré pour fonctionner en mode TS, AI, FX ou FSS.

```
$ ps -ef -o pset,class | grep -v CLS | sort | uniq
1 FSS
1 SYS
2 TS
```

2 RT
3 FX

Définition de la classe de programmation pour le système

Pour définir la classe de programmation par défaut pour le système, reportez-vous à la section [“Sélection de l'ordonnanceur FSS comme classe de programmation par défaut”](#) à la page 103, [“ Utilisation de l’ordonnanceur FSS sur un système Oracle Solaris doté de zones ”](#) du manuel [“ Création et utilisation des zones Oracle Solaris ”](#), et à [dispadmin\(1M\)](#). Pour déplacer des processus en cours dans une autre classe de programmation, reportez-vous à [“Configuration de l'ordonnanceur FSS”](#) à la page 103 et [prioctl\(1\)](#).

Classe de programmation dans un système doté de zones

Les zones non globales utilisent la classe de programmation par défaut pour le système. En cas de mise à jour du système avec un nouveau paramètre de classe de programmation par défaut, le paramètre est appliqué aux zones non globales au moment où vous les initialisez ou les redémarrez.

Dans ce cas, il est préférable de définir FSS comme classe de programmation par défaut du système à l'aide de la commande `dispadmin`. Toutes les zones disposent ainsi d'une part équitable des ressources CPU du système. Pour plus d'informations sur la programmation de classe lorsque des zones sont en cours d'utilisation, reportez-vous à la section [“ Utilisation de l’ordonnanceur FSS sur un système Oracle Solaris doté de zones ”](#) du manuel [“ Création et utilisation des zones Oracle Solaris ”](#).

Pour savoir comment transférer des processus en cours d'exécution vers une autre classe de programmation sans changer la classe de programmation par défaut et sans réinitialiser, reportez-vous à la page de manuel [prioctl\(1\)](#).

Commandes utilisées avec l'ordonnanceur FSS

Les commandes présentées dans le tableau suivant assurent l'interface d'administration principale avec l'ordonnanceur FSS.

Référence des commandes	Description
prioctl(1)	Affiche ou définit les paramètres de programmation des processus indiqués, transfère les processus en cours d'exécution vers une autre classe de programmation.

Référence des commandes	Description
ps(1)	Récapitule les informations au sujet des processus en cours d'exécution, identifie les classes de programmation dans lesquelles les jeux de processeurs s'exécutent.
dispadmin(1M)	Répertorie les ordonnanceurs du système. Configure l'ordonnanceur par défaut pour le système. Permet également d'examiner et de régler la valeur de quantum de l'ordonnanceur FSS.
FSS(7).	Affiche une description de l'ordonnanceur FSS.

9

♦ ♦ ♦ C H A P I T R E 9

Administration de l'ordonnanceur FSS (tâches)

Ce chapitre explique comment utiliser l'ordonnanceur FSS (Fair Share Scheduler).

Pour avoir un aperçu de l'ordonnanceur FSS, reportez-vous au [Chapitre 8, A propos de l'ordonnanceur de partage équitable](#). Pour plus d'informations sur la planification de classe lorsque les zones sont en cours d'utilisation, reportez-vous à la section “ [Ordonnanceur FSS sur un système doté de zones](#) ” du manuel “ [Création et utilisation des zones Oracle Solaris](#) ”.

Administration de l'ordonnanceur FSS (liste des tâches)

Tâche	Description	Pour plus d'informations
Surveillance de l'utilisation de la CPU.	Surveillez l'utilisation de la CPU pour les projets indépendants et les projets des jeux de processeurs.	“Surveillance de l'ordonnanceur FSS” à la page 102
Définition de la classe de programmation par défaut	Choisissez l'ordonnanceur FSS comme ordonnanceur par défaut pour le système.	“Sélection de l'ordonnanceur FSS comme classe de programmation par défaut” à la page 103
Transfert de processus en cours d'exécution vers une autre classe de programmation (vers la classe FSS, par exemple)	Déplacez manuellement des processus d'une classe de programmation vers une autre, sans changer la classe de programmation par défaut et sans réinitialiser.	“Transfert manuel de processus de la classe TS vers la classe FSS” à la page 104
Transfert de l'intégralité des processus en cours d'exécution vers une autre classe de programmation (vers la classe FSS, par exemple)	Déplacez manuellement des processus dans toutes les classes de programmation vers une autre, sans changer la classe de programmation par défaut et sans réinitialiser.	“Transfert manuel de toutes les classes de processus vers la classe FSS” à la page 104
Transfert des processus d'un projet vers une autre classe de programmation (vers la classe FSS, par exemple)	Déplacez manuellement les processus d'un projet de leur classe de programmation actuelle vers une autre classe de programmation.	“Transfert manuel des processus d'un projet vers la classe FSS” à la page 105
Examen et réglage des paramètres FSS	Ajustez la valeur de quantum de l'ordonnanceur. Le <i>quantum</i> est le délai pendant lequel un thread est autorisé à s'exécuter avant de devoir abandonner le processeur.	“Ajustement des paramètres de l'ordonnanceur” à la page 105

Surveillance de l'ordonnanceur FSS

Vous pouvez faire appel à la commande `prstat` décrite dans la page de manuel [prstat\(1M\)](#) pour surveiller la façon dont les projets actifs utilisent la CPU.

Il peut être intéressant également de tirer parti des données de comptabilisation étendue pour les tâches afin d'obtenir des statistiques par projet sur la quantité de ressources CPU utilisée sur des périodes plus longues. L'ensemble complet d'objectifs est documenté dans le [Chapitre 4, A propos de la comptabilisation étendue](#).

▼ Surveillance de l'utilisation de la CPU par projet

- Pour surveiller les ressources CPU utilisées par les projets exécutés sur le système, utilisez la commande `prstat` avec l'option `-J`.

```
# prstat -J
  PID USERNAME  SIZE  RSS STATE PRI NICE   TIME    CPU PROCESS/NLWP
5107 root        4556K 3268K cpu0  59   0   0:00:00 0.0% prstat/1
4570 root         83M  47M sleep  59   0   0:00:25 0.0% java/13
5105 bobbyc      3280K 2364K sleep  59   0   0:00:00 0.0% su/1
5106 root        3328K 2580K sleep  59   0   0:00:00 0.0% bash/1
   5 root           0K   0K sleep  99  -20   0:00:14 0.0% zpool-rpool/138
333 daemon      7196K 2896K sleep  59   0   0:00:07 0.0% rcapd/1
  51 netcfg      4436K 3460K sleep  59   0   0:00:01 0.0% netcfgd/5
2685 root        3328K 2664K sleep  59   0   0:00:00 0.0% bash/1
 101 netadm      4164K 2824K sleep  59   0   0:00:01 0.0% ipmgmt/6
 139 root        6940K 3016K sleep  59   0   0:00:00 0.0% syseventd/18
5082 bobbyc      2236K 1700K sleep  59   0   0:00:00 0.0% csh/1
  45 root         15M  7360K sleep  59   0   0:00:01 0.0% dlmgmt/7
  12 root         23M   22M sleep  59   0   0:00:45 0.0% svc.configd/22
  10 root         15M   13M sleep  59   0   0:00:05 0.0% svc.startd/19
337 netadm      6768K 5620K sleep  59   0   0:00:01 0.0% nwamd/9
PROJID  NPROC  SWAP  RSS  MEMORY   TIME    CPU PROJECT
    1      6   25M   18M    0.9%   0:00:00 0.0% user.root
    0     73  479M  284M   14%   0:02:31 0.0% system
    3      4   28M   24M    1.1%   0:00:26 0.0% default
   10      2   14M  7288K    0.3%   0:00:00 0.0% group.staff
```

Total: 85 processes, 553 lwps, load averages: 0.00, 0.00, 0.00

▼ Surveillance de l'utilisation de la CPU par projet dans les jeux de processeurs

- Pour contrôler les ressources CPU utilisées par les projets dans une liste de jeux de processeurs, entrez l'instruction suivante :

```
% prstat -J -C pset-list
```

où *pset-list* représente la liste des ID de jeux de processeurs séparés par des virgules.

Configuration de l'ordonnanceur FSS

Vous pouvez appliquer à l'ordonnanceur FSS les commandes utilisées pour les autres classes de programmation dans le système Oracle Solaris. Vous êtes libre de définir la classe de programmation, de configurer les paramètres ajustables de l'ordonnanceur et de paramétrer les propriétés des différents processus.

Sachez qu'il est possible d'utiliser la commande `svcadm restart` pour redémarrer le service de l'ordonnanceur. Pour plus d'informations, voir [svcadm\(1M\)](#).

Liste des classes de programmation sur le système

Pour afficher les classes de programmation du système, utilisez la commande `dispadmin` avec l'option `-l`.

```
$ dispadmin -l
CONFIGURED CLASSES
=====
SYS      (System Class)
TS       (Time Sharing)
SDC      (System Duty-Cycle Class)
FSS      (Fair Share)
FX       (Fixed Priority)
IA       (Interactive)
```

▼ Sélection de l'ordonnanceur FSS comme classe de programmation par défaut

Il est indispensable de désigner l'ordonnanceur FSS comme ordonnanceur par défaut sur votre système pour que les affectations de parts de CPU prennent effet.

Il suffit de combiner les commandes `priocntl` et `dispadmin` pour s'assurer que l'ordonnanceur FSS soit adopté par défaut immédiatement et après la réinitialisation.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**

2. Sélectionnez FSS comme ordonnanceur par défaut du système.

```
# dispadmin -d FSS
```

Ce changement sera effectif lors de la réinitialisation suivante. Après la réinitialisation, tous les processus du système s'exécutent dans la classe de programmation FSS.

3. Faites en sorte que cette configuration prenne effet immédiatement, sans avoir à réinitialiser.

```
# priocntl -s -c FSS -i all
```

▼ Transfert manuel de processus de la classe TS vers la classe FSS

Vous pouvez déplacer manuellement des processus d'une classe de programmation à une autre sans changer la classe de programmation par défaut et sans réinitialiser. Cette procédure montre comment déplacer manuellement des processus de la classe de programmation TS vers la classe de programmation FSS.

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.

2. Déplacez le processus init (pid 1) vers la classe de programmation FSS.

```
# priocntl -s -c FSS -i pid 1
```

3. Transférez tous les processus de la classe de programmation TS vers la classe de programmation FSS.

```
# priocntl -s -c FSS -i class TS
```

Remarque - Tous les processus s'exécutent à nouveau dans la classe de programmation TS après la réinitialisation.

▼ Transfert manuel de toutes les classes de processus vers la classe FSS

Il est possible que vous utilisiez une classe par défaut différente de la classe TS. Admettons, par exemple, que votre système exécute un environnement de multifenêtrage faisant appel à la classe AI par défaut. Vous avez la possibilité de transférer tous les processus dans la classe de programmation FSS sans avoir à changer la classe de programmation par défaut et sans avoir à réinitialiser.

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Déplacez le processus `init` (pid 1) vers la classe de programmation FSS.
3. Transférez tous les processus de leur classe de programmation actuelle vers la classe de programmation FSS.

```
# priocntl -s -c FSS -i pid 1
```

```
# priocntl -s -c FSS -i all
```

Remarque - Tous les processus s'exécutent à nouveau dans la classe de programmation par défaut après la réinitialisation.

▼ Transfert manuel des processus d'un projet vers la classe FSS

Vous pouvez déplacer manuellement les processus d'un projet depuis leur classe de programmation actuelle vers la classe de programmation FSS.

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Déplacez les processus s'exécutant dans l'ID de projet `10` vers la classe de programmation FSS.

```
# priocntl -s -c FSS -i projid 10
```

Les processus du projet s'exécutent à nouveau dans la classe de programmation par défaut après la réinitialisation.

Ajustement des paramètres de l'ordonnanceur

Vous pouvez vous servir de la commande `dispadmin` pour afficher ou changer les paramètres de programmation des processus lorsque le système est en cours d'exécution. Utilisez, par exemple, la commande `dispadmin` pour examiner et peaufiner la valeur de quantum de l'ordonnanceur FSS. Le *quantum* est le délai pendant lequel un thread est autorisé à s'exécuter avant de devoir abandonner le processeur.

Pour afficher le quantum actuel pour l'ordonnanceur FSS lorsque le système est en cours d'exécution, entrez l'instruction suivante :

```
$ dispadmin -c FSS -g  
#
```

```
# Fair Share Scheduler Configuration
#
RES=1000
#
# Time Quantum
#
QUANTUM=110
```

Lorsque vous utilisez l'option -g, vous pouvez également lui associer l'option -r afin de spécifier la résolution d'affichage des valeurs de quantum. En l'absence de résolution, les valeurs de quantum sont affichées en millisecondes par défaut.

```
$ dispadmin -c FSS -g -r 100
#
# Fair Share Scheduler Configuration
#
RES=100
#
# Time Quantum
#
QUANTUM=11
```

Pour définir les paramètres de programmation pour la classe de programmation FSS, exécutez `dispadmin -s`. Les valeurs dans *file* doivent correspondre au format généré par l'option -g. Ces valeurs remplacent les valeurs actuelles dans le noyau. Saisissez ce qui suit :

```
$ dispadmin -c FSS -s file
```

10

♦ ♦ ♦ C H A P I T R E 1 0

A propos du contrôle de la mémoire physique à l'aide du démon de limitation des ressources

Le démon de limitation des ressources `rcapd` permet de réguler l'utilisation de la mémoire physique par les processus exécutés dans le cadre de projets faisant l'objet d'une allocation restrictive de ressources. si vous utilisez des zones sur votre système, vous avez la possibilité d'exécuter la commande `rcapd` à partir de la zone globale dans le but de mieux gérer la mémoire physique réservée aux zones non globales. Reportez-vous à la section “ [Contrôle de la mémoire physique et ressource capped-memory](#) ” du manuel “ [Présentation d'Oracle Solaris Zones](#) ”.

Ce chapitre comprend les sections suivantes :

- “[Introduction au démon de limitation des ressources](#)” à la page 107
- “[Principe de fonctionnement de la limitation des ressources](#)” à la page 108
- “[Attribut permettant de limiter l'utilisation de la mémoire physique pour les projets](#)” à la page 109
- “[Configuration de la commande `rcapd`](#)” à la page 109
- “[Surveillance des ressources à l'aide de `rcapstat`](#)” à la page 114
- “[Commandes utilisées avec `rcapd`](#)” à la page 115

Pour obtenir les instructions nécessaires à l'utilisation de la commande `rcapd`, reportez-vous au [Chapitre 11, Administration du démon de limitation des ressources \(tâches\)](#).

Introduction au démon de limitation des ressources

La *limitation* des ressources est le procédé consistant à fixer un seuil d'utilisation maximum des ressources (telles que la mémoire physique). Il est permis de prévoir des restrictions de mémoire physique par projet.

Le démon d'allocation restrictive des ressources et les utilitaires connexes offrent différents mécanismes pour mettre en oeuvre et gérer les limites s'appliquant à la mémoire physique.

Comme pour le contrôle des ressources, la limitation des ressources peut être configurée en utilisant certains attributs des entrées de projet dans la base de données `project`. Mais à la

différence des contrôles de ressources qui sont synchronisés par le noyau, l'allocation restrictive de ressources est appliquée de façon asynchrone au niveau utilisateur par le démon de limitation des ressources. Le mode asynchrone induit un léger décalage en raison de l'intervalle d'analyse du démon.

Pour plus d'informations sur la commande `rcapd`, reportez-vous à la page de manuel [rcapd\(1M\)](#). Pour plus d'informations sur les projets et la base de données `project`, voir [Chapitre 2, A propos des projets et tâches](#) et la page de manuel [project\(4\)](#). Pour plus d'informations au sujet des contrôles de ressources, reportez-vous au [Chapitre 6, A propos des contrôles de ressources](#).

Principe de fonctionnement de la limitation des ressources

Le démon analyse à maintes reprises l'utilisation des ressources par les projets auxquels s'appliquent des allocations restrictives. C'est l'administrateur qui définit l'intervalle auquel le démon analyse ces informations. Pour plus d'informations, reportez-vous à la section ["Détermination des intervalles d'échantillonnage" à la page 113](#) En cas de dépassement du seuil d'utilisation maximale de la mémoire physique du système et si d'autres conditions sont remplies, le démon prend les mesures nécessaires pour réduire les ressources utilisées par les projets concernés jusqu'au niveau (ou en dessous) des limites prévues.

Le système de mémoire virtuelle divise la mémoire physique en segments appelés pages. Les pages représentent l'unité de base de la mémoire physique dans le sous-système de gestion de la mémoire Oracle Solaris. Pour lire les données à partir d'un fichier en mémoire, le système de mémoire virtuelle *charge* chaque page du fichier à tour de rôle. Pour économiser les ressources, le démon a la possibilité de *décharger* de la mémoire les pages les moins souvent consultées ou de les transférer vers un périphérique de swap (c'est-à-dire vers une zone à l'extérieur de la mémoire physique).

Le démon gère la mémoire physique en adaptant la taille résidente définie de la charge de travail d'un projet à sa taille de travail. Le jeu de pages résidant en mémoire physique. La taille de travail correspond au jeu de pages utilisé de façon active par la charge de travail pendant son cycle de traitement. Elle varie au cours du temps, selon le mode opératoire du processus et le type de données en cours de traitement. Dans l'idéal, chaque charge de travail dispose d'une quantité suffisante de mémoire physique pour que sa taille de travail reste résidente. Cette dernière peut également tirer parti d'un espace de stockage sur disque secondaire prévu pour accueillir les pages qui ne tiennent pas dans la mémoire physique.

Vous ne pouvez exécuter qu'une seule instance à la fois de la commande `rcapd`.

Attribut permettant de limiter l'utilisation de la mémoire physique pour les projets

Pour définir le seuil limite de la mémoire physique susceptible d'être allouée à un projet, il convient d'établir une taille résidente définie en ajoutant cet attribut à l'entrée de la base de données project :

`rcap.max-rss` Quantité totale de mémoire physique, en octets, mise à la disposition des processus dans le projet.

L'instruction suivante dans le fichier `/etc/project` fixe, par exemple, la taille résidente définie à 10 gigaoctets pour le projet nommé `db`.

```
db:100::db,root::rcap.max-rss=10737418240
```

Remarque - Le système peut éventuellement arrondir la valeur indiquée en fonction de la taille de page.

Vous pouvez aussi faire appel à la commande `projmod` pour définir l'attribut `rcap.max-rss` dans le fichier `/etc/project`.

Pour plus d'informations, reportez-vous à la section Définition de la limite de taille résidente définie.

Configuration de la commande `rcapd`

Pour configurer le démon d'allocation restrictive, vous ferez appel à la commande `rcapadm`. Vous pouvez effectuer les actions suivantes :

- Fixer la valeur de seuil d'allocation restrictive de la mémoire
- Définir à quel intervalle analyser le mode d'exploitation des ressources à l'aide de la commande `rcapd`
- Activer ou désactiver la limitation des ressources
- Afficher l'état actuel du démon de limitation des ressources configuré

Pour configurer le démon, vous devez être l'utilisateur `root` ou posséder les droits d'administration requis.

Les modifications apportées à la configuration peuvent être incorporées à `rcapd` en fonction de l'intervalle de configuration défini (voir "[Intervalles des opérations `rcapd`](#)" à la page 112) ou à la demande en envoyant un signal `SIGHUP` (voir la page de manuel `kill(1)`).

En l'absence d'argument, `rcapadm` affiche l'état actuel du démon de limitation des ressources s'il a été configuré.

Les sous-sections suivantes expliquent comment mettre en oeuvre la limitation des ressources, comment définir les valeurs de seuil et les intervalles d'opération rcapd

Utilisation du démon de limitation des ressources sur un système doté de zones

Vous pouvez gérer le mode d'utilisation de la taille résidente définie (RSS) dans une zone en définissant la ressource capped-memory au moment de configurer la zone. Pour plus d'informations, reportez-vous à la section [“ Contrôle de la mémoire physique et ressource capped-memory ” du manuel “ Présentation d’Oracle Solaris Zones ”](#). Pour utiliser la ressource capped-memory, le package resource-cap doit être installé dans la zone globale. Il est possible d'exécuter la commande rcapd dans une zone, y compris dans la zone globale, en vue d'imposer des limites de mémoire aux projets de cette zone.

Vous pouvez définir une limite temporaire sur la quantité maximale de mémoire pouvant être utilisée par une zone spécifiée, jusqu'à la prochaine réinitialisation. Reportez-vous à la section [“Spécification d'une limitation temporaire de ressources pour une zone” à la page 121](#).

Si vous utilisez la commande rcapd dans une zone dans le but de réguler l'utilisation de la mémoire physique par les processus des projets faisant l'objet de restrictions de ressources, il est indispensable de configurer le démon dans cette zone.

Lorsque vous choisissez des limitations de mémoire pour des applications dans plusieurs zones, le fait qu'elles appartiennent à différentes zones n'a généralement pas d'importance sauf pour les services par zone. Les services par zone ont, en effet, besoin de mémoire. Il faut donc en tenir compte lorsque vous calculez la quantité de mémoire physique réservée à un système et déterminez les limitations de mémoire.

Seuil d'allocation restrictive de la mémoire

Le *seuil d'allocation restrictive de la mémoire* correspond au pourcentage d'utilisation de la mémoire physique sur le système qui déclenche la restriction. En cas de dépassement du seuil prévu, les limitations de mémoire sont automatiquement appliquées. Ce pourcentage tient compte de la mémoire physique utilisée par les applications et le noyau. Le pourcentage d'utilisation détermine la façon dont les limitations de mémoire sont mises en oeuvre.

Pour appliquer les limitations de mémoire, des pages de mémoire peuvent être retirées des charges de travail du projet.

- Le déchargement des pages de mémoire permet de réduire la taille de la portion de mémoire qui dépasse le seuil fixé pour une charge de travail donnée.
- Le déchargement des pages de mémoire permet de réduire la proportion de mémoire physique qui dépasse le seuil d'allocation restrictive de la mémoire sur le système.

Une charge de travail est autorisée à utiliser la mémoire physique jusqu'au seuil prévu. Elle peut cependant s'accaparer plus de mémoire tant que l'utilisation des ressources du système ne dépasse pas le seuil d'allocation restrictive de la mémoire.

Pour définir la valeur d'application du seuil, reportez-vous à la section [“Définition du seuil d'allocation restrictive de la mémoire” à la page 119](#).

Détermination des valeurs de seuil

Lorsque le seuil prévu pour un projet est trop bas, il est possible que la mémoire ne soit pas suffisante pour réaliser la charge de travail dans des conditions normales. La pagination liée à une surexploitation de la mémoire par la charge de travail a un effet négatif sur les performances du système.

Les projets dont le seuil est trop élevé peuvent disposer de la mémoire physique jusqu'à leur seuil respectif. Dans ce cas, la mémoire physique est effectivement gérée par le noyau et non pas par la commande rcapd.

Lors de la détermination des valeurs de seuil des projets, il est important de prendre en compte les facteurs suivants.

Impact sur le système d'E/S

Il est possible que le démon essaie de réduire la mémoire physique accaparée par la charge de travail d'un projet dès lors que la quantité de mémoire utilisée dépasse le seuil prévu pour le projet. Lors de l'application du seuil maximum d'utilisation de la mémoire physique, les périphériques de swap et les autres périphériques contenant des fichiers mappés par la charge de travail sont mis à contribution. Les performances des périphériques de swap sont donc un facteur essentiel pour déterminer les performances d'une charge de travail qui dépasse régulièrement le seuil fixé. Le traitement de la charge de travail revient à l'exécuter sur une machine possédant une quantité de mémoire physique équivalente au seuil prévu pour la charge de travail.

Impact sur l'utilisation de la CPU

L'utilisation de la CPU par le démon varie en fonction du nombre de processus mis en jeu dans les charges de travail du projet concerné par les limitations de ressources et des tailles des espaces d'adressage des charges de travail.

Une petite fraction du temps CPU du démon est consacrée à l'échantillonnage de chaque charge de travail. Plus le nombre de processus mis en jeu est important, plus la durée d'échantillonnage augmente.

Une autre partie du temps CPU est consacrée à l'application des seuils en cas de dépassement des limites. La durée est proportionnelle à la quantité de mémoire virtuelle impliquée. Le temps CPU utilisé

augmente ou diminue pour faire face aux variations de la quantité d'espace d'adressage totale d'une charge de travail. Cette information est reportée dans la colonne `vm` de la sortie `rcapstat`. Pour plus d'informations, reportez-vous à [“Surveillance des ressources à l'aide de rcapstat” à la page 114](#) et à la page de manuel `rcapstat(1)`.

Prise en compte
de la mémoire
partagée

Le démon `rcapd` signale au RSS les pages de mémoire partagées avec d'autres processus ou mappées à plusieurs reprises au cours d'un même processus, sous forme d'estimation assez précise. Si des processus de projets différents partagent la même mémoire, celle-ci est prise en compte dans le total RSS pour tous les projets partageant la mémoire. L'estimation peut être utilisée avec les charges de travail telles que celles des bases de données, qui se servent de la mémoire partagée de manière importante. Pour les charges de travail de base de données, il est également possible d'utiliser un échantillon de l'utilisation habituelle d'un projet afin de déterminer une valeur limite initiale appropriée à partir du résultat des options `-J` ou `-Z` de la commande `prstat`. Pour plus d'informations sur `pool`, reportez-vous à la page de manuel `prstat(1M)`.

Intervalles des opérations rcapd

Vous pouvez régler les intervalles de contrôle périodique de la commande `rcapd`.

Tous les intervalles sont exprimés en secondes. Les opérations effectuées par `rcapd` et les valeurs d'intervalle par défaut sont décrites dans le tableau suivant.

Opération	Valeur d'intervalle par défaut en secondes	Description
scan	15	Nombre de secondes entre chaque analyse des processus ayant rejoint ou quitté une charge de travail d'un projet. La valeur minimum est de 1 seconde.
sample	5	Nombre de secondes entre chaque échantillonnage de la taille de mémoire résidente et les applications des limites correspondantes. La valeur minimum est de 1 seconde.
report	5	Nombre de secondes entre les mises à jour des statistiques de pagination. Si la valeur choisie est 0, les statistiques ne sont pas mises à jour et la sortie obtenue avec <code>rcapstat</code> n'est pas actualisée.

Opération	Valeur d'intervalle par défaut en secondes	Description
config	60	Nombre de secondes entre chaque reconfiguration. Dans un événement de reconfiguration, <code>rcapadm</code> vérifie la présence de mises à jour dans le fichier de configuration et analyse la base de données <code>project</code> afin de déterminer si elle contient de nouvelles limitations ou si les limitations ont été révisées. L'envoi d'un signal <code>SIGHUP</code> à la commande <code>rcapd</code> provoque une reconfiguration immédiate.

Pour régler les intervalles, reportez-vous à la section [“Définition des intervalles de fonctionnement”](#) à la page 120.

Détermination des intervalles d'analyse `rcapd`

L'intervalle d'analyse définit la fréquence à laquelle `rcapd` recherche de nouveaux processus. L'analyse prend plus de temps sur les systèmes comptant de nombreux processus en cours d'exécution. Si tel est le cas, il est préférable de rallonger l'intervalle afin de réduire le temps CPU global utilisé. Il ne faut cependant pas oublier que l'intervalle représente également la durée minimale des processus pour qu'ils soient assignés à une charge de travail à ressources limitées. Or, si des charges de travail exécutent de nombreux processus à durée réduite, `rcapd` risque de ne pas allouer les processus à une charge de travail si l'intervalle d'analyse est trop long.

Détermination des intervalles d'échantillonnage

L'intervalle d'échantillonnage configuré avec `rcapadm` correspond au délai d'attente le plus court avant que `rcapd` échantillonne l'utilisation d'une charge de travail et applique la limitation prévue en cas de dépassement. Si vous réduisez cet intervalle, `rcapd` appliquera de façon plus régulière les limitations prévues, ce qui aura pour effet d'augmenter les E/S liées à la pagination. A l'inverse, un intervalle d'échantillonnage plus court peut limiter l'impact d'une brusque augmentation de la mémoire physique utilisée par une charge de travail sur les autres charges de travail. Le délai entre chaque échantillonnage (c'est-à-dire la durée pendant laquelle la charge de travail peut utiliser de la mémoire sans encombre et même exploiter la mémoire allouée à d'autres charges de travail) est limité.

Si l'intervalle d'échantillonnage prévu pour `rcapstat` est inférieur à celui appliqué à `rcapd` avec `rcapadm`, certains intervalles risquent d'être équivalents à zéro. Cette situation est due au fait que `rcapd` ne met pas à jour les statistiques plus fréquemment que l'intervalle spécifié avec

rcapadm. L'intervalle spécifié avec rcapadm est indépendant de l'intervalle d'échantillonnage utilisé par rcapstat.

Surveillance des ressources à l'aide de rcapstat

Utilisez rcapstat pour surveiller l'utilisation des ressources des projets faisant l'objet d'une restriction de ressources. Pour afficher un exemple de rapport rcapstat, reportez-vous à la section [“Etablissement de rapports à l'aide de la commande rcapstat”](#) à la page 121.

Vous pouvez définir l'intervalle d'échantillonnage pour le rapport et spécifier le nombre de répétitions des statistiques.

<i>interval</i>	Spécifie l'intervalle d'échantillonnage en secondes. La valeur par défaut est de 5 secondes.
<i>count</i>	Indique le nombre de fois où les statistiques sont répétées. Par défaut, rcapstat établit des statistiques jusqu'à la réception d'un signal de fin ou jusqu'à l'arrêt du processus rcapd.

Les statistiques de pagination dans le premier rapport émis par rcapstat présentent l'activité depuis le démarrage du démon. Les rapports suivants reflètent l'activité depuis le dernier rapport.

Le tableau suivant décrit les en-têtes de colonne d'un rapport rcapstat.

En-têtes de colonne rcapstat	Description
id	Id du projet auquel s'applique la restriction de ressource.
projet	Nom du projet.
nproc	Nombre de processus dans le projet.
vm	Taille de la mémoire virtuelle totale utilisée par les processus du projet, tous fichiers et périphériques mappés inclus, en kilo-octets (K), mégaoctets (M) ou gigaoctets (G).
rss	Estimation de la quantité totale de taille de mémoire résidente définie des processus du projet, en kilo-octets (K), mégaoctets (M) ou gigaoctets (G), à l'exclusion des pages partagées.
cap	Limite de taille résidente définie prévue pour le projet. Pour plus d'informations sur les limitations de mémoire, voir “Attribut permettant de limiter l'utilisation de la mémoire physique pour les projets” à la page 109 ou la page de manuel rcapd(1M) .
at	Quantité totale de mémoire que la commande rcapd a essayé de décharger depuis le dernier échantillonnage rcapstat.

En-têtes de colonne rcapstat	Description
vgat	La quantité moyenne de mémoire que la commande rcapd a essayé de décharger lors de chaque cycle d'échantillonnage ayant eu lieu depuis le dernier échantillonnage rcapstat. Vitesse à laquelle la commande rcapd échantillonne l'utilisation de la taille résidente définie pour les jeux avec rcapadm. Voir "Intervalles des opérations rcapd" à la page 112.
pg	Quantité totale de mémoire que la commande rcapd a réussi à décharger depuis le dernier échantillonnage rcapstat.
avgpg	Une estimation de la quantité moyenne de mémoire que la commande rcapd a déchargé lors de chaque cycle d'échantillonnage ayant eu lieu depuis le dernier échantillonnage rcapstat. Vitesse à laquelle la commande rcapd échantillonne l'utilisation de la taille résidente définie pour les processus avec rcapadm. Voir "Intervalles des opérations rcapd" à la page 112.

Commandes utilisées avec rcapd

Référence des commandes	Description
rcapstat(1)	Surveille l'utilisation des ressources des projets faisant l'objet d'une restriction de ressources.
rcapadm(1M)	Configure le démon de limitation des ressources, affiche l'état actuel du démon s'il a été configuré et active ou désactive la limitation des ressources. Egalement utilisé pour définir une limitation temporaire de mémoire.
rcapd(1M)	Démon de limitation des ressources.

BETA
DRAFT

♦ ♦ ♦ 11 CHAPITRE 11

Administration du démon de limitation des ressources (tâches)

Ce chapitre présente les procédures à suivre pour configurer et utiliser le démon de limitation des ressources `rcapd`.

Pour plus d'informations sur la commande `rcapd`, reportez-vous à [Chapitre 10, A propos du contrôle de la mémoire physique à l'aide du démon de limitation des ressources](#).

Définition de la limite de taille résidente définie

Définissez la limite de taille résidente définie (RSS) pour les ressources d'une mémoire physique d'un projet en ajoutant un attribut `rcap.max-rss` à l'entrée de la base de données `project`.

▼ Ajout d'un attribut `rcap.max-rss` pour un projet

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.
2. Ajoutez l'attribut suivant au fichier `/etc/project` :

```
rcap.max-rss=value
```

Exemple 11-1 Limite du projet RSS

L'instruction suivante dans le fichier `/etc/project` fixe la taille résidente définie à 10 gigaoctets pour le projet nommé `db`.

```
db:100::db,root::rcap.max-rss=10737418240
```

Notez que le système peut éventuellement arrondir la valeur indiquée en fonction de la taille de page.

▼ Utilisation de la commande projmod pour ajouter un attribut rcap.max-rss pour un projet

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Définissez un attribut rcap.max-rss de 10 gigaoctets dans le fichier /etc/project, dans cet exemple, pour un projet nommé db.

```
# projmod -a -K rcap.max-rss=10GB db
```

Le fichier /etc/project contient alors la ligne suivante :

```
db:100::db,root::rcap.max-rss=10737418240
```

Configuration et utilisation du démon de limitation des ressources (liste des tâches)

Tâche	Description	Pour obtenir des instructions
Définition du seuil d'allocation restrictive de la mémoire	Configurez le niveau d'allocation maximum alloué lorsque la mémoire physique disponible est insuffisante.	“Définition du seuil d'allocation restrictive de la mémoire” à la page 119
Définition de l'intervalle d'opération	Indiquez l'intervalle pendant lequel le démon de limitation des ressources effectue ses opérations périodiques.	“Définition des intervalles de fonctionnement” à la page 120
Activation de la limitation des ressources.	Activez la limitation des ressources sur votre système.	“Activation de la limitation des ressources” à la page 120
Désactivation de la limitation des ressources.	Désactivez la limitation des ressources sur votre système.	“Désactivation de la limitation des ressources” à la page 120
Génération de rapports sur la limitation des ressources et sur le projet	Produisez des rapports à l'aide des exemples de commandes proposés.	“Création d'un rapport sur la limitation des ressources et sur le projet” à la page 122
Surveillance de la taille résidente définie d'un projet	Etablissez un rapport sur la taille résidente définie d'un projet.	“Surveillance de la taille résidente définie d'un projet” à la page 122
Détermination de la taille de charge de travail définie d'un projet	Etablissez un rapport sur la taille de fonctionnement définie d'un projet.	“Détermination de la taille de la charge de travail définie d'un projet” à la page 123
Génération de rapports sur l'utilisation de la mémoire et les limitations définies en la matière	Affichez une ligne relative à l'utilisation de la mémoire et au seuil d'allocation restrictive à la fin du rapport pour chaque intervalle défini.	“Création d'un rapport sur l'utilisation de la mémoire et le seuil d'allocation restrictive” à la page 124

Administration du démon de limitation des ressources avec rcapadm

Cette section présente les procédures à suivre pour configurer le démon de limitation des ressources avec rcapd. Pour plus d'informations, reportez-vous à [“Configuration de la commande rcapd” à la page 109](#) et à la page de manuel [rcapadm\(1M\)](#). Cette section aborde également la spécification de limitations temporaires de ressources pour une zone à l'aide de la commande rcapadm.

En l'absence d'argument, rcapadm affiche l'état actuel du démon de limitation des ressources s'il a été configuré.

▼ Définition du seuil d'allocation restrictive de la mémoire

Il est possible de configurer les seuils d'allocation restrictive de façon à les appliquer uniquement lorsque la mémoire physique mise à la disposition des processus ne suffit plus. Pour plus d'informations, reportez-vous à la section [“Seuil d'allocation restrictive de la mémoire” à la page 110](#).

La valeur minimum (valeur par défaut) est de 0. Elle permet d'appliquer systématiquement les seuils d'allocation restrictive. Pour changer la valeur minimum, procédez de la façon suivante.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Utilisez l'option -c de la commande rcapadm pour redéfinir le niveau d'utilisation de la mémoire physique pour le seuil d'allocation restrictive de la mémoire.**

rcapadm -c percent

percent est compris entre 0 et 100. Les valeurs élevées sont moins restrictives. Autrement dit, une valeur élevée signifie qu'il est possible d'effectuer les charges de travail d'un projet (pour lequel une allocation restrictive de la mémoire a été définie) en s'affranchissant des limitations prévues tant que le seuil d'utilisation de la mémoire du système n'est pas atteint.

Voir aussi Pour afficher le niveau actuel d'utilisation de la mémoire physique et le seuil d'allocation restrictive, reportez-vous à la section [“Création d'un rapport sur l'utilisation de la mémoire et le seuil d'allocation restrictive” à la page 124](#).

▼ Définition des intervalles de fonctionnement

La section “[Intervalles des opérations rcapd](#)” à la page 112 contient des informations sur les intervalles des opérations périodiques réalisées par rcapd. Pour définir ces intervalles à l'aide de la commande rcapadm, procédez de la façon suivante.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Utilisez l'option -i pour définir les valeurs des intervalles.**

```
# rcapadm -i interval=value,...,interval=value
```

Remarque - Les intervalles sont tous exprimés en secondes.

▼ Activation de la limitation des ressources

Vous disposez de trois modes d'activation de la limitation des ressources sur votre système. L'activation de la limitation des ressources permet également de définir les valeurs par défaut du fichier /etc/rcap.conf.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Activez le démon de limitation des ressources de l'une des façons suivantes :**

- Activez la limitation des ressources à l'aide de la commande svcadm.

```
# svcadm enable rcap
```

- Activez le démon de limitation des ressources de façon à le démarrer dès maintenant et à chaque initialisation du système :

```
# rcapadm -E
```

- Pour activer le démon de limitation des ressources uniquement à l'initialisation en spécifiant l'option -n, entrez l'instruction suivante :

```
# rcapadm -n -E
```

▼ Désactivation de la limitation des ressources

Vous disposez de trois modes de désactivation de la limitation des ressources sur votre système.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**

2. **Désactivez le démon de limitation des ressources de l'une des façons suivantes :**

- **Désactivez la limitation des ressources à l'aide de la commande `svcadm`.**

```
# svcadm disable rcap
```

- **Pour désactiver le démon de limitation des ressources de façon à l'arrêter dès maintenant et à chaque initialisation du système, entrez l'instruction suivante :**

```
# rcapadm -D
```

- **Pour désactiver le démon de limitation des ressources sans l'arrêter, spécifiez également l'option `-n` :**

```
# rcapadm -n -D
```

Astuce - Utilisez `rcapadm -D` pour désactiver `rcapd`. En cas d'interruption du démon (voir la page de manuel `kill(1)`), les processus risquent d'être maintenus à l'arrêt et vous devrez alors les redémarrer manuellement. Pour reprendre un processus, servez-vous de la commande `prun`. Pour plus d'informations, reportez-vous à la page de manuel [prun\(1\)](#).

▼ **Spécification d'une limitation temporaire de ressources pour une zone**

Cette procédure est utilisée pour allouer la quantité maximale de mémoire pouvant être utilisée par une zone spécifique. Cette valeur reste valide uniquement jusqu'à la prochaine réinitialisation. Pour définir une limitation persistante, utilisez la commande `zonecfg`.

1. **Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.**
2. **Définissez une valeur de mémoire maximale de 512 Mo pour la zone `my-zone`.**

```
# rcapadm -z testzone -m 512M
```

Etablissement de rapports à l'aide de la commande `rcapstat`

Servez-vous de la commande `rcapstat` pour obtenir des rapports statistiques sur la limitation des ressources. La section [“Surveillance des ressources à l'aide de `rcapstat`”](#) à la page 114 explique comment générer des rapports à l'aide de la commande `rcapstat`. Cette section

décrit également les en-têtes de colonne du rapport. Ces informations sont également détaillées dans la page de manuel [rcapstat\(1\)](#).

Les sous-sections suivantes illustrent par des exemples le mode de création de divers rapports.

Création d'un rapport sur la limitation des ressources et sur le projet

Dans cet exemple, les limitations s'appliquent à deux projets associés à deux utilisateurs. user1 est limité à 50 mégaoctets alors que user2 est limité à 10 mégaoctets.

La commande suivante permet d'obtenir cinq rapports à intervalle de 5 secondes.

```
user1machine% rcapstat 5 5
  id project nproc  vm   rss  cap   at avgat   pg avgpg
112270  user1    24  123M   35M  50M  50M   0K 3312K   0K
78194   user2     1 2368K 1856K 10M   0K   0K   0K   0K
  id project nproc  vm   rss  cap   at avgat   pg avgpg
112270  user1    24  123M   35M  50M   0K   0K   0K   0K
78194   user2     1 2368K 1856K 10M   0K   0K   0K   0K
  id project nproc  vm   rss  cap   at avgat   pg avgpg
112270  user1    24  123M   35M  50M   0K   0K   0K   0K
78194   user2     1 2368K 1928K 10M   0K   0K   0K   0K
  id project nproc  vm   rss  cap   at avgat   pg avgpg
112270  user1    24  123M   35M  50M   0K   0K   0K   0K
78194   user2     1 2368K 1928K 10M   0K   0K   0K   0K
  id project nproc  vm   rss  cap   at avgat   pg avgpg
112270  user1    24  123M   35M  50M   0K   0K   0K   0K
78194   user2     1 2368K 1928K 10M   0K   0K   0K   0K
```

Les trois premières lignes de la sortie correspondent au premier rapport. Celui-ci contient des informations relatives aux limitations et au projet pour les deux projets et des statistiques de pagination depuis le démarrage de rcapd. Les colonnes at et pg columns contiennent un nombre supérieur à zéro pour user1 et zéro pour user2, ce qui signifie qu'à un moment donné dans l'historique du démon, user1 a dépassé la limite fixée, à l'inverse de user2.

Les rapports suivants ne font état d'aucune activité significative.

Surveillance de la taille résidente définie d'un projet

L'exemple suivant contient des informations sur le user1 du projet pour lequel la taille résidente définie dépasse le seuil défini en la matière.

La commande suivante permet d'obtenir cinq rapports à intervalle de 5 secondes.

```
user1machine% rcapstat 5 5
```

	id	project	nproc	vm	rss	cap	at	avgat	pg	avpg
376565	user1	3	6249M	6144M	6144M	690M	220M	5528K	2764K	
376565	user1	3	6249M	6144M	6144M	0M	131M	4912K	1637K	
376565	user1	3	6249M	6171M	6144M	27M	147M	6048K	2016K	
376565	user1	3	6249M	6146M	6144M	4872M	174M	4368K	1456K	
376565	user1	3	6249M	6156M	6144M	12M	161M	3376K	1125K	

Comme vous pouvez le constater, trois processus du projet de user1 font un usage intensif de la mémoire physique. Les valeurs positives de la colonne pg signifient que la commande rcapd renvoie une page de mémoire lorsqu'elle essaie de se conformer au seuil fixé en limitant la mémoire physique utilisée par les processus du projet. rcapd ne parvient pas, cependant, à maintenir la taille résidente définie en dessous de la valeur limite. Comme le prouvent les différentes valeurs rss qui ne montrent pas une réduction correspondante. Dès qu'une page de mémoire est déchargée, la charge de travail l'exploite à nouveau et la taille résidente définie augmente. Cela signifie que l'intégralité de la mémoire résidente du projet est utilisée de façon active et que la taille de travail définie (WSS) est supérieure au seuil fixé. La commande rcapd doit, dans ce cas, décharger une partie de la taille de travail pour éviter de dépasser ce seuil. Le système continuera à connaître un nombre important de défauts de page et d'erreurs E/S associées, jusqu'à ce que l'une des conditions suivantes soit remplie :

- La taille de la charge de travail définie est abaissée.
- Le seuil limite est augmenté.
- L'application change son mode d'accès à la mémoire.

Dans ce type de situation, raccourcir l'intervalle peut limiter l'écart entre la valeur de la taille résidente définie et la valeur limite en permettant à la commande rcapd d'analyser la charge de travail et d'appliquer les limitations plus fréquemment.

Remarque - Un défaut de page se produit lorsqu'il devient nécessaire de créer une nouvelle page ou lorsque le système est dans l'obligation de copier une page à partir d'un périphérique de swap.

Détermination de la taille de la charge de travail définie d'un projet

L'exemple suivant est une suite logique de l'exemple précédent et se base sur le même projet.

L'exemple précédent montre que le projet user1 utilise plus de mémoire physique que le permet en théorie le seuil fixé. L'exemple suivant montre la quantité de mémoire nécessaire pour assurer la charge de travail du projet.

```
user1machine% rcapstat 5 5
```

	id	project	nproc	vm	rss	cap	at	avgat	pg	avpg
376565	user1	3	6249M	6144M	6144M	690M	0K	689M	0K	

376565	user1	3	6249M	6144M	6144M	0K	0K	0K	0K
376565	user1	3	6249M	6171M	6144M	27M	0K	27M	0K
376565	user1	3	6249M	6146M	6144M	4872K	0K	4816K	0K
376565	user1	3	6249M	6156M	6144M	12M	0K	12M	0K
376565	user1	3	6249M	6150M	6144M	5848K	0K	5816K	0K
376565	user1	3	6249M	6155M	6144M	11M	0K	11M	0K
376565	user1	3	6249M	6150M	10G	32K	0K	32K	0K
376565	user1	3	6249M	6214M	10G	0K	0K	0K	0K
376565	user1	3	6249M	6247M	10G	0K	0K	0K	0K
376565	user1	3	6249M	6247M	10G	0K	0K	0K	0K
376565	user1	3	6249M	6247M	10G	0K	0K	0K	0K
376565	user1	3	6249M	6247M	10G	0K	0K	0K	0K
376565	user1	3	6249M	6247M	10G	0K	0K	0K	0K
376565	user1	3	6249M	6247M	10G	0K	0K	0K	0K

Vers la moitié du cycle, la limite prévue pour le projet user1 a été augmentée de 6 à 10 gigaoctets. Cette hausse empêche la mise en oeuvre de l'allocation restrictive et a pour effet d'accroître la taille résidente définie dans la limite fixée par les autres processus et en fonction de la quantité de mémoire installée. Les valeurs de la colonne rss auront tendance à se stabiliser pour refléter la taille de la charge de travail définie du projet (6 247 Mo dans cet exemple). Il s'agit de la valeur minimum d'allocation restrictive nécessaire au fonctionnement des processus du projet sans que cela n'engendre des défauts de page continus.

Tant que cette valeur limite du projet user1 sera équivalente à 6 gigaoctets par intervalle de 5 secondes, la taille résidente définie diminuera et les E/S augmenteront au fur et à mesure que la commande `rcapd` décharge des pages de la mémoire réservée à la charge de travail. Peu de temps après le déchargement d'une page, la charge de travail aura pour effet de les recharger en mémoire pour continuer son exécution. Ce cycle se reproduit jusqu'à ce que le seuil passe à 10 gigaoctets, à mi-chemin de l'exemple. La taille résidente définie se stabilise ensuite à 6,1 gigaoctets. Comme la taille résidente définie de la charge de travail est désormais inférieure au seuil fixé, plus aucun transfert de page n'a lieu. Les E/S associées à la pagination cessent également. Autrement dit, le projet a eu besoin de 6,1 Go pour effectuer le travail en cours de traitement au moment de son observation.

Voir également les pages de manuel [vmstat\(1M\)](#) et [iostat\(1M\)](#).

Création d'un rapport sur l'utilisation de la mémoire et le seuil d'allocation restrictive

Vous pouvez associer l'option `-g` à la commande `rcapstat` pour générer un rapport sur les données suivantes :

- Pourcentage d'utilisation actuelle de la mémoire physique par rapport à la mémoire physique installée sur le système
- Seuil d'allocation restrictive de la mémoire système tel qu'il a été défini par la commande `rcapadm`

L'option -g permet d'afficher une ligne sur l'utilisation de la mémoire et l'allocation restrictive de la mémoire à la fin du rapport pour chaque intervalle.

```
# rcapstat -g
  id project  nproc   vm   rss   cap   at avgat   pg  avgpg
376565  rcap      0    0K    0K   10G    0K    0K    0K    0K
physical memory utilization: 55%  cap enforcement threshold: 0%
  id project  nproc   vm   rss   cap   at avgat   pg  avgpg
376565  rcap      0    0K    0K   10G    0K    0K    0K    0K
physical memory utilization: 55%  cap enforcement threshold: 0%
```

BETA
DRAFT

BETA
DRAFT

12

♦ ♦ ♦ C H A P I T R E 1 2

A propos des pools de ressources

Ce chapitre présente les technologies suivantes :

- Pools de ressources : moyens utilisés pour partitionner les ressources des machines
- Pools de ressources dynamiques (DRP) : moyens permettant de régler de façon dynamique l'allocation des ressources pour chaque pools de ressources afin de remplir les objectifs système qui ont été fixés.

Les pools de ressources et les pools de ressources dynamiques sont des services de l'utilitaire de gestion des services (SMF) d'Oracle Solaris. Ces services sont activés indépendamment les uns des autres.

Ce chapitre comprend les sections suivantes :

- [“Introduction aux pools de ressources” à la page 128](#)
- [“Introduction aux pools de ressources dynamiques” à la page 129](#)
- [“A propos de l'activation et de la désactivation des pools de ressources et des pools de ressources dynamiques” à la page 129](#)
- [“Pools de ressources utilisés dans les zones” à la page 130](#)
- [“Intérêt des pools” à la page 130](#)
- [“Structure des pools de ressources” à la page 131](#)
- [“Implémentation des pools sur un système” à la page 133](#)
- [“Attribut `project.pool`” à la page 134](#)
- [“opérations de reconfiguration dynamique et pools de ressources” à la page 134](#)
- [“Création de configurations de pools” à la page 135](#)
- [“Manipulation directe de la configuration dynamique” à la page 136](#)
- [“Présentation de `pool'd`” à la page 136](#)
- [“Gestion des pools de ressources dynamiques” à la page 137](#)
- [“Contraintes et objectifs de configuration” à la page 137](#)
- [“Fonctionnalités `pool'd` configurables” à la page 142](#)
- [“Mode de fonctionnement de l'allocation dynamique des ressources” à la page 146](#)
- [“Utilisation de `poolstat` pour surveiller l'utilitaire des pools et l'utilisation des ressources” à la page 148](#)

- “Commandes utilisées avec l'utilitaire des pools de ressources” à la page 150

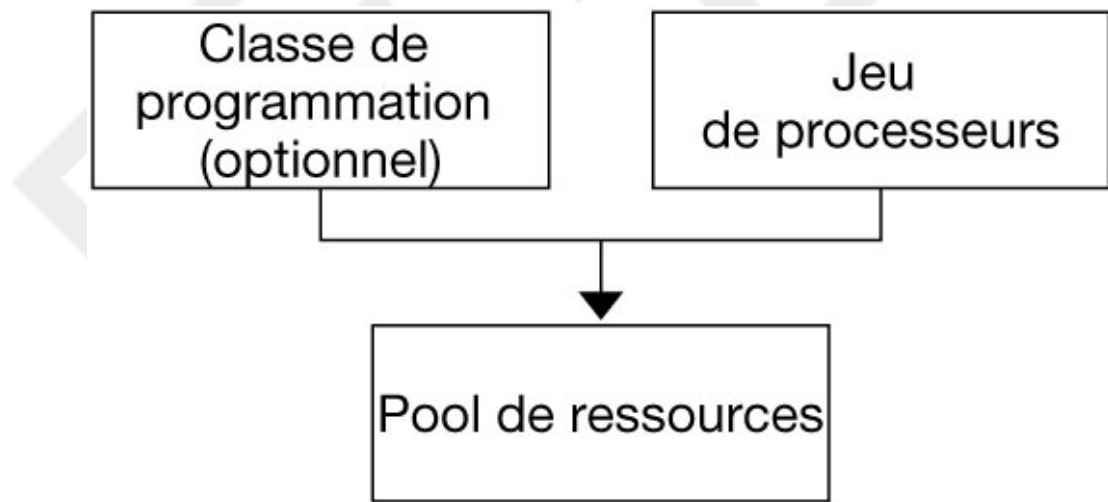
Pour connaître les procédures tirant parti de cette fonctionnalité, reportez-vous au [Chapitre 13, Création et administration des pools de ressources \(tâches\)](#).

Introduction aux pools de ressources

Les pools de ressources vous permettent de séparer des charges de travail de façon à éviter la surconsommation de certaines charges de travail. Cette réservation de ressource contribue à atteindre les performances attendues sur les systèmes combinant des charges de travail hybrides.

Les pools de ressources fournissent un mécanisme de configuration persistant en vue de la configuration du jeu de processeurs (pset) et, éventuellement, de l'assignation d'une classe de programmation.

FIGURE 12-1 Structure de pool de ressources



Il est possible de concevoir un pool comme la liaison spécifique des différents lots de ressources disponibles sur votre système. Vous pouvez créer des pools représentant les différents types de combinaisons de ressources possibles :

```
pool1: pset_default  
pool2: pset1  
pool3: pset1, pool.scheduler="FSS"
```

En regroupant plusieurs partitions, les pools fournissent un descripteur à associer aux charges de travail étiquetées. Chaque entrée de projet figurant dans le fichier `/etc/project` peut posséder un seul pool associé à cette entrée, spécifié grâce à l'attribut `project.pool`.

Lorsque les pools sont activés, un *pool par défaut* et un *jeu de processeurs par défaut* forment la configuration de base. Il est possible de créer d'autres pools et jeux de processeurs définis par l'utilisateur et de les ajouter à la configuration. Une CPU ne peut appartenir qu'à un seul jeu de processeurs. Les pools et les jeux de processeurs définis par l'utilisateur peuvent être détruits à la différence du pool par défaut et du jeu de processeurs par défaut.

La propriété `pool.default` du pool par défaut a la valeur `true`. La propriété `pset.default` du jeu de processeurs par défaut a la valeur `true`. Ainsi, le pool et le jeu de processeurs par défaut restent identifiables même après modification de leurs noms.

Le mécanisme des pools défini par l'utilisateur est initialement prévu pour les ordinateurs équipés d'au moins quatre CPU. Néanmoins, des ordinateurs de plus petites tailles peuvent tirer parti de cette fonctionnalité. Sur ces ordinateurs, la création des pools permet de partager des partitions de ressources non vitales. La séparation en pools se fait exclusivement sur la base de ressources critiques.

Introduction aux pools de ressources dynamiques

Les pools de ressources dynamiques offrent un mécanisme permettant d'ajuster de manière dynamique l'allocation des ressources de chaque pool en réponse aux événements système et aux modifications des charges de travail des applications. Les pools de ressources dynamiques simplifient et réduisent le nombre des décisions exigées d'un administrateur. Les ajustements s'effectuent automatiquement de façon à respecter les objectifs de performances spécifiés par un administrateur. Les modifications apportées à la configuration sont consignées dans un journal. Ces fonctions sont principalement du ressort du contrôleur de ressources `pool`, un démon système qui doit toujours être actif lorsque l'affectation dynamique de ressources est requise. `pool` examine régulièrement la charge du système et détermine si une intervention est nécessaire pour permettre au système de maintenir des performances optimales dans le cadre de la consommation des ressources. La configuration `pool` est conservée au sein de la configuration `libpool`. Pour plus d'informations sur `pool`, reportez-vous à la page de manuel [pool\(1M\)](#).

A propos de l'activation et de la désactivation des pools de ressources et des pools de ressources dynamiques

Pour activer et désactiver les pools de ressources et les pools de ressources dynamiques, reportez-vous à la section [“Activation et désactivation de l'utilitaire Pools”](#) à la page 154.

Pools de ressources utilisés dans les zones

au lieu d'associer une zone à un pool de ressources configuré sur votre système, vous pouvez utiliser la commande `zonecfg` pour créer un pool temporaire qui devient actif lorsque la zone est exécutée. Pour plus d'informations, reportez-vous à [“Création et utilisation des zones Oracle Solaris”](#).

Dans un système dont les zones sont activées, une zone non globale peut être associée à un pool de ressources bien que le pool n'ait pas besoin d'être exclusivement assigné à une zone spécifique. Par ailleurs, vous n'êtes pas en mesure de lier des processus de zones non globales à un pool différent à l'aide de la commande `poolbind` depuis la zone globale. Pour associer une zone non globale à un pool, reportez-vous à [“Création et utilisation des zones Oracle Solaris”](#).

Notez que si vous définissez une classe de programmation pour un pool et que vous associez une zone non globale à ce pool, la zone utilise par défaut cette classe de programmation.

Si vous utilisez des pools de ressources dynamiques, la portée d'une instance d'exécution de `poold` se limite à la zone globale.

L'utilitaire `poolstat` affiche uniquement les informations relatives au pool associé à la zone non globale dans laquelle il s'exécute. La commande `pooladm` exécutée sans argument dans une zone non globale affiche uniquement les informations concernant le pool associé à la zone en question.

Pour plus d'informations sur les commandes de pools de ressources, reportez-vous à la section [“Commandes utilisées avec l'utilitaire des pools de ressources”](#) à la page 150.

Intérêt des pools

Les pools de ressources proposent un mécanisme polyvalent applicable dans de nombreux cas de figure.

Serveur de calcul à traitement par lots	Utilisez la fonctionnalité des pools pour fractionner un serveur en deux pools. Un pool est dédié aux sessions de connexion et aux opérations interactives effectuées par les utilisateurs travaillant en mode de partage de temps. L'autre pool sert aux tâches soumises via le système de traitement par lots.
Serveur d'applications ou de bases de données	Partitionnez les ressources réservées aux applications interactives conformément aux besoins des applications.
Activation des applications par phases	Définissez les attentes des utilisateurs. Vous pouvez, au départ, déployer un ordinateur exécutant une partie des services qu'il est censé fournir. Les utilisateurs peuvent éprouver

des difficultés si les mécanismes de gestion de ressources basés sur la réservation ne sont pas établis à la connexion de l'ordinateur.

Par exemple, l'ordonnanceur FSS optimise l'utilisation de la CPU. Les délais de réponse d'un ordinateur n'exécutant qu'une seule application peuvent vous induire en erreur par leur rapidité. Les utilisateurs ne bénéficieront pas de tels délais de réponse lorsque plusieurs applications sont chargées. L'utilisation de pools distincts pour chaque application permet de fixer un plafond au nombre de CPU accessibles par chaque application avant de déployer toutes les applications.

Serveur de partage
de temps complexe

Partitionnez un serveur prenant en charge des populations importantes d'utilisateurs. Le partitionnement de serveur est un mécanisme d'isolement qui permet de mieux anticiper les réponses par utilisateur.

En répartissant les utilisateurs en plusieurs groupes dépendant de pools distincts et en utilisant la fonctionnalité FSS, vous pouvez ajuster les allocations de CPU pour donner la priorité à des ensembles d'utilisateurs. Vous pouvez vous baser sur le rôle utilisateur, sur des critères d'imputation comptable, etc.

Charges de travail
variant d'une saison
à l'autre

Servez-vous des pools de ressources pour adapter les ressources en fonction de la demande.

Il est possible que les variations des charges de travail soient prévisibles sur votre site sur de longues périodes (cycles mensuels, trimestriels ou annuels, par exemple). Si c'est le cas, vous pouvez passer d'une configuration de pools à une autre en faisant appel à la commande `pooladm` à partir d'une tâche cron. (Voir [“Structure des pools de ressources” à la page 131.](#))

Applications en
temps réel

Créez un pool en temps réel en utilisant l'ordonnanceur RT et les ressources désignées du processeur.

Utilisation du
système

Imposez les objectifs système que vous fixez.

Servez-vous du démon de pools automatisé pour identifier les ressources disponibles, puis surveillez les charges de travail afin de détecter à quel moment les objectifs spécifiés ne sont plus atteints. Le démon procède à des corrections si cela est possible ; sinon, la condition est enregistrée.

Structure des pools de ressources

Le fichier de configuration `/etc/pooladm.conf` décrit la configuration des pools statiques. Une configuration statique représente la manière dont un administrateur aimerait configurer un système en fonction du pool de ressources. Il est possible de spécifier un autre nom de fichier.

Lorsque l'utilitaire de gestion des service (SMF) ou la commande `pooladm -e` est utilisé pour activer la structure des pools de ressources, la configuration contenue dans le fichier `/etc/pooladm.conf` (si celui-ci existe) est appliquée au système.

Le noyau stocke des informations au sujet de la disposition des ressources au sein de la structure des pools de ressources. Ce procédé, appelé configuration dynamique, représente les pools de ressources pour un système particulier à un moment précis dans le temps. Vous pouvez examiner la configuration dynamique à l'aide de la commande `pooladm`. L'ordre d'affichage des propriétés pour les pools et les lots de ressources peut varier. Vous pouvez procéder de l'une ou l'autre des façons suivantes pour modifier la configuration dynamique :

- Indirectement, en appliquant un fichier de configuration statique
- Directement, en exécutant la commande `poolcfg` avec l'option `-d`

Vous pouvez avoir le choix entre plusieurs fichiers de configuration statique de pools et activer ponctuellement celui qui convient le mieux. Il est possible de passer d'une configuration de pools à une autre en faisant appel à la commande `pooladm` à partir d'une tâche `cron`. Pour plus d'informations sur l'utilitaire `cron`, reportez-vous à la page de manuel [cron\(1M\)](#).

Par défaut, la structure des pools de ressources n'est pas active. Or, il est indispensable d'activer les pools de ressources pour créer ou modifier la configuration dynamique. Vous pouvez manipuler des fichiers de configuration statique à l'aide des commandes `poolcfg` ou `libpool` même lorsque la structure des pools de ressources est désactivée. Il est cependant impossible de créer des fichiers de configuration statique si l'utilitaire des pools n'est pas en service. Pour plus d'informations au sujet du fichier de configuration, reportez-vous à la section “[Création de configurations de pools](#)” à la page 135.

Les commandes utilisées avec les pools de ressources et le démon système `poold` sont décrites dans les pages de manuel suivantes :

- [pooladm\(1M\)](#)
- [poolbind\(1M\)](#)
- [poolcfg\(1M\)](#)
- [poold\(1M\)](#)
- [poolstat\(1M\)](#)
- [libpool\(3LIB\)](#)

Contenu du fichier `/etc/pooladm.conf`

Toutes les configurations de pools de ressource, y compris la configuration dynamique, peuvent contenir les éléments suivants.

système	Propriétés ayant une incidence sur le comportement global du système
pool	Définition du pool de ressources
pset	Définition du jeu de processeurs
cpu	Définition du processeur

Tous ces éléments possèdent des propriétés sur lesquelles vous pouvez intervenir pour changer l'état et le comportement de la structure des pools de ressources. La propriété de `pool.importance` indique, par exemple, l'importance relative d'un pool donné. Elle s'avère pratique pour résoudre des conflits d'utilisation des ressources. Pour plus d'informations, reportez-vous à [libpool\(3LIB\)](#).

Propriétés des pools

L'utilitaire des pools prend en charge les propriétés nommées saisies applicables à un pool, une ressource ou un composant. Les administrateurs peuvent stocker des propriétés supplémentaires au sujet des divers éléments du pool. Ils utilisent pour cela un espace de noms de propriétés similaire à l'attribut de projet.

Le commentaire suivant indique, par exemple, qu'un jeu de processeurs donné est associé à une base de données `Datatree` particulière.

```
Datatree,pset.dbname=warehouse
```

Pour plus d'informations au sujet des types de propriété, reportez-vous à la section “[Propriétés pool'd](#)” à la page 142.

Remarque - Un certain nombre de propriétés spéciales est réservé à un usage interne. Il est impossible de les configurer ou de les supprimer. Pour plus d'informations, reportez-vous à la page de manuel [libpool\(3LIB\)](#).

Implémentation des pools sur un système

Voici comment procéder pour mettre en oeuvre sur un système les pools définis par les utilisateurs.

- A l'initialisation du logiciel Oracle Solaris, un script `init` vérifie la présence du fichier `/etc/pooladm.conf`. S'il détecte ce fichier et si les pools sont activés, la commande

`pooladm` est exécutée pour rendre active cette configuration de pools. Le système crée une configuration dynamique afin de refléter l'organisation demandée dans `/etc/pooladm.conf` et les ressources de la machine sont partitionnées en conséquence.

- Lors de l'exécution du système Oracle Solaris, vous avez la possibilité d'activer une configuration de pools si elle n'existe pas déjà, ou de la modifier à l'aide de la commande `pooladm`. Par défaut, la commande `pooladm` s'applique à `/etc/pooladm.conf`. Vous pouvez, cependant, choisir un autre emplacement et un autre nom de fichier et utiliser ce fichier pour mettre à jour la configuration de pools.

Pour savoir comment activer et désactiver des pools de ressources, reportez-vous à la section [“Activation et désactivation de l'utilitaire Pools” à la page 154](#). Il est impossible de désactiver l'utilitaire des pools lorsque des pools ou des ressources définis par l'utilisateur sont en cours d'utilisation.

Pour configurer les pools de ressources, vous devez disposer de privilèges root ou d'un profil doté des droits requis.

Le contrôleur de ressources `poold` est démarré à l'aide de l'utilitaire des pools de ressources dynamiques.

Attribut `project.pool`

L'attribut `project.pool` peut être ajouté à une entrée de projet dans le fichier `/etc/project` afin d'associer un pool à cette entrée. Tout nouveau travail démarré pour un projet est lié au pool approprié. Pour plus d'informations, reportez-vous à [Chapitre 2, A propos des projets et tâches](#).

Vous pouvez, par exemple, vous servir de la commande `projmod` pour définir l'attribut `project.pool` pour le projet `sales` dans le fichier `/etc/project` :

```
# projmod -a -K project.pool=mypool sales
```

SPARC: opérations de reconfiguration dynamique et pools de ressources

La reconfiguration dynamique, appelée aussi opération DR (Dynamic Reconfiguration), permet de reconfigurer le matériel lorsque le système est en cours d'exécution. Ce type d'opération permet d'augmenter ou de réduire un type de ressource, mais peut aussi n'avoir aucun impact. Comme les opérations DR peuvent influencer sur les quantités de ressources disponibles, il est indispensable d'y associer l'utilitaire des pools. Lors du démarrage d'une opération DR, la structure de pools se charge de valider la configuration.

Si l'opération DR peut se poursuivre sans risque de rendre non valide la configuration de pools actuelle, le fichier de configuration privé est mis à jour. Une configuration non valide est une configuration non compatible avec les ressources disponibles.

Si l'opération DR provoque l'invalidité de la configuration de pools, elle échoue et un message est consigné dans le journal correspondant. Dans ce cas, si vous souhaitez mener la configuration à son terme, vous devez utiliser l'option DR force. La configuration de pools est alors modifiée pour se conformer à la nouvelle configuration des ressources. Pour plus d'informations sur l'opération DR et l'option DR force, reportez-vous au guide utilisateur de la reconfiguration dynamique pour votre matériel Sun.

Si vous utilisez des pools de ressources dynamiques, sachez qu'il est possible d'extraire une partition du contrôle pool'd pendant que le démon est actif. Pour plus d'informations, reportez-vous à la section ["Identification d'un manque de ressources" à la page 147](#).

Création de configurations de pools

Le fichier de configuration contient une description des pools à créer sur le système. Il présente les différents éléments qu'il est possible de manipuler.

- système
- pool
- pset
- cpu

Pour plus d'informations au sujet des éléments susceptibles d'être manipulés, reportez-vous à la page de manuel [poolcfg\(1M\)](#).

Lorsque les pools sont activés, vous pouvez créer un fichier `/etc/pooladm.conf` structuré de deux façons.

- Vous pouvez exécuter la commande `pooladm` avec l'option `-s` afin de repérer les ressources sur le système actuel et de placer les résultats dans un fichier de configuration.
Il s'agit de la méthode préférée. L'intégralité des ressources et des composants actifs sur le système et manipulables par l'utilitaire de pools est prise en compte. Les ressources incluent les configurations de jeux de processeurs existantes. Vous êtes libre ensuite de modifier la configuration pour renommer les jeux de processeurs ou créer des pools supplémentaires, si cela est nécessaire.
- Vous pouvez utiliser la commande `poolcfg` avec l'option `-c` et les sous-commandes `discover` ou `create system name` pour créer une nouvelle configuration de pools.
Ces options ont été préservées pour assurer la compatibilité avec les versions précédentes.

Servez-vous de `poolcfg` ou de `libpool` pour modifier le fichier `/etc/pooladm.conf`. N'éditez pas directement ce fichier.

Affectation spécifique de CPU, de noyaux et de sockets

Utilisez les sous-commandes `assign` et `unassign` pour affecter des UC, des noyaux et des sockets spécifiques.

Les sous-commandes `assign` et `unassign` sont applicables aux configurations persistantes et d'exécution. L'utilisation directe de la commande `assign` et la configuration de `pset.min` et `pset.max` s'excluent mutuellement. Chaque méthode remplace la configuration de l'autre. Les UC configurées en `psets` à l'aide des propriétés `pset.min` et `pset.max` sont considérées comme *allouées* plutôt qu'assignées. `assign` et `unassign` ajoutent et suppriment des UC spécifiques à un `pset`. Le premier `assign` effacera toute configuration de `pset` définie par une allocation précédente. Utilisez `unassign` uniquement après un `assign` réussi. `unassign` ne peut pas servir à manipuler des UC sur des `pset` alloués.

Reportez-vous également à “ [Ressource dedicated-cpu](#) ” du manuel “ [Présentation d'Oracle Solaris Zones](#) ”.

Manipulation directe de la configuration dynamique

Il est possible de manipuler directement les types de ressources CPU dans la configuration dynamique en associant la commande `poolcfg` à l'option `-d`. Il existe deux méthodes pour transférer les ressources.

- Vous pouvez effectuer une demande générale afin de déplacer les ressources identifiées disponibles d'un jeu à un autre.
- Transférez, par exemple, des ressources avec des ID spécifiques vers un jeu de destination. Notez que les ID système associés aux ressources peuvent varier en cas de modification de la configuration des ressources ou après une réinitialisation du système.

Vous trouverez un exemple à ce sujet à la section “[Transfert de ressources](#)” à la page 169.

Si `DRP` est en cours d'utilisation, notez que le transfert de ressources peut déclencher une action de `poolld`. Pour plus d'informations, reportez-vous à la section “[Présentation de poolld](#)” à la page 136.

Présentation de poolld

Le contrôleur des ressources des pools, `poolld`, tire parti des cibles système et des statistiques observables pour atteindre les objectifs de performances que vous avez fixés. Ce démon système doit toujours rester actif lorsque l'allocation dynamique des ressources est nécessaire.

Le contrôleur des ressources `pool`d identifie les ressources disponibles et surveille les charges de travail pour savoir précisément à quel moment les objectifs d'utilisation système ne sont plus remplis. `pool`d considère ensuite des configurations alternatives en termes d'objectifs et engage des actions correctives. Les ressources sont reconfigurées de façon à pouvoir atteindre les objectifs. Si cela n'est pas possible, le démon indique dans le journal que les objectifs définis par l'utilisateur ne peuvent plus être remplis. A la suite d'une reconfiguration, le démon reprend la surveillance des objectifs des charges de travail.

`pool`d tient à jour un historique des décisions en vue de l'examiner en cas de besoin. Cet historique sert à éviter les reconfigurations qui n'ont pas permis d'apporter de réelles améliorations dans le passé.

Une reconfiguration peut également être déclenchée de façon asynchrone en cas de modification des objectifs des charges de travail ou des ressources accessibles au système.

Gestion des pools de ressources dynamiques

Le service des pools de ressources dynamiques (DRP, Dynamic Resource Pools) est géré par l'utilitaire de gestion des services (SMF, Service Management Facility) sous l'identificateur de service `svc:/system/pools/dynamic`.

Les actions administratives appliquées à ce service (activation, désactivation ou demande de redémarrage, par exemple), peuvent être réalisées à l'aide de la commande `svcadm`. Servez-vous de la commande `svcs` pour connaître l'état du service. Pour plus d'informations, reportez-vous aux pages de manuel [svcs\(1\)](#) et [svcadm\(1M\)](#).

L'interface SMF est la méthode de prédilection pour contrôler le service DRP, mais pour des raisons de compatibilité ascendante, les méthodes suivantes sont également préconisées.

- Si aucune allocation dynamique des ressources n'est nécessaire, il est possible d'arrêter `pool`d avec le signal `SIGQUIT` ou `SIGTERM`. Ces deux signaux mettent un terme progressif à `pool`d.
- Bien que `pool`d détecte automatiquement les changements apportés à la configuration des ressources ou des pools, vous pouvez imposer une reconfiguration en utilisant le signal `SIGHUP`.

Contraintes et objectifs de configuration

Lorsque vous apportez des modifications à une configuration, `pool`d agit dans le sens que vous indiquez, c'est-à-dire en tenant compte de la série de contraintes et d'objectifs que vous définissez. `pool`d utilise vos spécifications pour juger de la valeur relative des différentes

possibilités de configuration par rapport à la configuration existante. `pool` change ensuite les affectations de ressources de la configuration actuelle pour établir les nouvelles configurations candidates.

Contraintes de configuration

Les contraintes limitent l'étendue des configurations possibles en empêchant certaines modifications potentielles. Les contraintes suivantes, spécifiées dans la configuration `libpool`, sont disponibles.

- Allocations minimum et maximum de la CPU
- Composants rattachés non transférables d'un jeu à un autre
- Importance du facteur de pool

Pour plus d'informations sur les propriétés des pools, reportez-vous à la page de manuel [libpool\(3LIB\)](#) et à [“Propriétés des pools” à la page 133](#).

Pour plus d'instructions, reportez-vous à la section [“Définition des contraintes de configuration” à la page 165](#).

Propriété `pset.min` et contraintes de propriété `pset.max`

Ces deux propriétés fixent le nombre limite de processeurs (valeurs minimum et maximum) qu'il est possible d'allouer à un jeu de processeurs. Pour en savoir plus sur ces propriétés, reportez-vous au [Tableau 12-1, “Noms des propriétés définies”](#).

En vertu de ces contraintes, les ressources d'une partition peuvent être allouées à d'autres partitions dans la même instance Oracle Solaris. Pour accéder à la ressource, il convient d'établir une liaison avec un pool associé au lot de ressources. La liaison est réalisée automatiquement au moment de la connexion ou manuellement par un administrateur doté du privilège `PRIV_SYS_RES_CONFIG`.

Contrainte de propriété `cpu.pinned`

La propriété `cpu-pinned` stipule que le service des pools de ressources dynamiques n'est pas habilité à transférer une UC donnée à partir de son jeu de processeurs d'origine. Vous pouvez définir cette propriété `libpool` pour optimiser l'utilisation du cache d'une application s'exécutant dans un jeu de processeurs.

Pour en savoir plus sur cette propriété, reportez-vous au [Tableau 12-1, “Noms des propriétés définies”](#).

Contrainte de propriété `pool.importance`

La propriété `pool.importance` décrit l'importance relative d'un pool tel qu'elle est définie par l'administrateur.

Objectifs de configuration

Les objectifs sont définis de la même manière que pour les contraintes. L'ensemble complet d'objectifs est documenté dans le [Tableau 12-1, “Noms des propriétés définies”](#).

Il existe deux catégories d'objectifs.

Objectif dépendant de la charge de travail Il s'agit d'un objectif qui varie en fonction de la nature de la charge de travail s'exécutant sur le système, à l'image de l'objectif `utilization`. Le chiffre d'utilisation pour un lot de ressources dépend de la nature de la charge de travail active.

Objectif indépendant de la charge de travail Il s'agit d'un objectif qui ne varie pas en fonction de la nature de la charge de travail s'exécutant sur le système. C'est le cas de l'objectif `CPU locality`. L'estimation du voisinage d'un lot de ressources ne dépend pas de la nature de la charge de travail active.

Vous pouvez définir trois types d'objectif.

Nom	Eléments valides	Opérateurs	Valeurs
<code>wt-load</code>	<code>system</code>	<code>SO</code>	<code>SO</code>
<code>locality</code>	<code>pset</code>	<code>SO</code>	<code>loose tight none</code>
<code>utilization</code>	<code>pset</code>	<code>< > ~</code>	<code>0-100%</code>

Les objectifs sont stockés dans les chaînes de propriétés dans la configuration `libpool`. Les noms des propriétés se présentent de la façon suivante :

- `system.pool.d.objectives`
- `pset.pool.d.objectives`

La syntaxe des objectifs est la suivante :

- `objectives = objective [; objective]*`
- `objective = [n:] keyword [op] [value]`

Tous les objectifs possèdent un préfixe relatif au niveau d'importance (facultatif). Ce niveau d'importance fait office de multiplicateur pour l'objectif et augmente par conséquent sa

contribution à l'évaluation de la fonction de l'objectif. La plage d'évaluation va de 0 à INT64_MAX (9223372036854775807). Si vous omettez de spécifier le niveau d'importance, la valeur par défaut est 1.

Certains types d'élément acceptent plusieurs types d'objectif. C'est le cas, par exemple, de pset. Vous pouvez, en effet, définir plusieurs types d'objectif pour ces éléments. Il est possible également de fixer plusieurs objectifs d'utilisation pour un même élément pset.

La section [“Etablissement des objectifs de configuration” à la page 165](#) offre des exemples d'utilisation.

Objectif wt-load

L'objectif wt-load favorise les configurations faisant correspondre les allocations aux utilisations réelles des ressources. Un lot de ressources nécessitant plus de ressources bénéficiera de plus de ressources lorsque cet objectif est actif. wt-load signifie *charge pondérée*.

Utilisez cet objectif lorsque vous êtes satisfait des contraintes que vous avez établies à l'aide des propriétés minimum et maximum et souhaitez que le démon manipule librement les ressources dans les limites autorisées.

L'objectif locality

L'objectif locality change l'impact du voisinage, tel qu'il est mesuré par les données du groupe de voisinages (lgroupe), sur la configuration sélectionnée. La latence est une autre définition du voisinage. lgroupe décrit les ressources de la CPU et de la mémoire. lgroupe est utilisé par le système Oracle Solaris pour déterminer la distance entre les ressources en fonction du facteur temps.

Cet objectif peut prendre l'une des trois valeurs suivantes :

tight	Si cette valeur est définie, les configurations qui maximisent le voisinage des ressources sont favorisées.
loose	Si cette valeur est définie, les configurations qui minimisent le voisinage des ressources sont favorisées.
none	Si cette valeur est définie, le voisinage des ressources n'a pas d'influence sur les configurations favorisées. Il s'agit de la valeur par défaut de l'objectif locality.

En général, l'objectif locality doit être défini sur tight. Cependant, pour optimiser la bande passante de la mémoire ou minimiser l'impact des opérations de reconfiguration dynamique

sur un lot de ressources, vous pouvez lui donner la valeur `loose` ou conserver le paramètre par défaut (`none`).

Objectif `utilization`

L'objectif `utilization` favorise les configurations allouant des ressources aux partitions ne satisfaisant pas l'objectif d'utilisation spécifié.

Il est défini au moyen d'opérateurs et de valeurs. Les opérateurs qu'il est possible d'utiliser sont les suivants :

- | | |
|---|--|
| < | L'opérateur "inférieur à" indique que la valeur spécifiée représente une valeur cible maximum. |
| > | L'opérateur "supérieur à" indique que la valeur spécifiée représente une valeur cible minimum. |
| ~ | Cet opérateur Indique que la valeur spécifiée est la valeur cible pour laquelle une certaine fluctuation est acceptable. |

Un seul objectif d'utilisation peut être prévu pour chaque type d'opérateur dans le cadre d'un jeu de processeurs (pset).

- Si vous utilisez l'opérateur `~`, les opérateurs `<` et `>` ne sont pas exploitables.
- Si vous utilisez les opérateurs `<` et `>`, l'opérateur `~` n'est pas exploitable. Les paramètres de l'opérateur `<` et de l'opérateur `>` ne peuvent pas être contradictoires.

Vous pouvez combiner les opérateurs `<` et `>` pour définir une plage. Les valeurs seront validées pour éviter tout chevauchement.

Exemple d'utilisation des objectifs dans une configuration

Dans l'exemple suivant, `pool` permet d'évaluer ces objectifs pour le jeu de processeurs (pset) :

- L'objectif `utilization` doit être maintenu entre 30 et 80 %.
- L'objectif `locality` doit être maximisé pour le jeu de processeurs.
- Il convient d'accorder le niveau d'importance par défaut (1) aux objectifs.

EXEMPLE 12-1 Exemple d'objectifs `pool`

```
pset.pool objectives "utilization > 30; utilization < 80; locality tight"
```

La section [“Etablissement des objectifs de configuration” à la page 165](#) offre des exemples d'utilisation supplémentaires.

Propriétés pool'd

Il existe quatre catégories de propriétés :

- Configuration
- Contrainte
- Objectif
- Paramètre d'objectif

TABLEAU 12-1 Noms des propriétés définies

Nom de propriété	Type	Catégorie	Description
system.pool'd.log-level	Chaîne	Configuration	Niveau de consignation
system.pool'd.log-location	Chaîne	Configuration	Emplacement de journalisation
system.pool'd.monitor-interval	UInt64	Configuration	Intervalle de contrôle
system.pool'd.history-file	Chaîne	Configuration	Emplacement de l'historique des décisions
pset.max	UInt64	Contrainte	Nombre maximum de CPU pour ce jeu de processeurs
pset.min	UInt64	Contrainte	Nombre minimum de CPU pour ce jeu de processeurs
cpu.pinned	Booléen	Contrainte	CPU rattachées à ce jeu de processeurs
system.pool'd.objectives	Chaîne	Objectif	Chaîne formatée après la syntaxe d'expression d'objectif pool'd
pset.pool'd.objectives	Chaîne	Objectif	Chaîne formatée après la syntaxe d'expression de pool'd
pool.importance	UInt64	Paramètre d'objectif	Importance définie par l'utilisateur

Fonctionnalités pool'd configurables

Vous pouvez configurer les aspects suivants du comportement du démon.

- Intervalle de surveillance
- Niveau de consignation
- Emplacement de journalisation

Ces options sont spécifiées dans la configuration des pools. Vous pouvez également gérer le niveau de consignation à partir de la ligne de commande en faisant appel à pool'd.

Intervalle de surveillance `pool`d

Servez-vous du nom de propriété `system.pool`d.`monitor-interval` pour spécifier une valeur en millisecondes.

Informations de journalisation `pool`d

La consignation donne accès à trois catégories d'informations. Ces catégories sont identifiées dans les journaux :

- Configuration
- Monitoring (Contrôle)
- Optimisation

Servez-vous du nom de propriété `system.pool`d.`log-level` pour spécifier le paramètre de consignation. En cas d'omission de cette propriété, c'est le niveau de consignation par défaut (NOTICE) qui est appliqué. Les niveaux de paramétrage sont hiérarchiques. Le niveau de consignation `DEBUG` donne à `pool`d l'instruction d'inscrire tous les messages définis dans le journal. Le niveau de consignation `INFO` offre un ensemble équilibré d'informations qui convient à la plupart des administrateurs.

A partir de la ligne de commande, associez la commande `pool`d à une option `-l` et à un paramètre afin de définir le niveau de consignation voulu.

Voici l'ensemble des paramètres disponibles :

- ALERT
- CRIT
- ERR
- WARNING
- NOTICE
- INFO
- DEBUG

Les niveaux de paramètre sont mis directement en correspondance avec leurs équivalents `syslog`. Pour plus d'informations au sujet de l'utilisation de la commande [“Emplacement de journalisation” à la page 145](#), reportez-vous à la section `Logging Location`.

Pour plus d'informations au sujet de la configuration de la consignation `pool`d, reportez-vous à la section [“Définition du niveau de consignation `pool`d” à la page 167](#).

Journalisation des informations de configuration

Voici les types de message susceptibles d'être générés :

ALERT	Problèmes d'accès à la configuration de libpool ou autre défaillance inopinée de l'utilitaire libpool. Provoque l'arrêt du démon et exige une attention immédiate du service administratif.
CRIT	Problèmes liés à des défaillances inopinées. Provoque l'arrêt du démon et exige une attention immédiate du service administratif.
ERR	Problèmes liés aux paramètres définis par l'utilisateur en matière de contrôle des opérations. Il peut s'agir, par exemple, d'objectifs d'utilisation conflictuels et impossible à résoudre pour un lot de ressources. Le service administratif doit intervenir pour corriger les objectifs. pool'd essaie de remédier à cela en ignorant les objectifs conflictuels, mais certaines erreurs provoqueront l'arrêt du démon.
WARNING	Avertissements liés à la définition des paramètres de configuration qui, même s'ils sont techniquement corrects, risquent de ne pas convenir à l'environnement d'exécution. Le fait, par exemple, de marquer toutes les ressources CPU comme étant fixées empêche pool'd de transférer ces ressources d'un jeu de processeurs à un autre.
DEBUG	Messages contenant des informations détaillées nécessaires lors du débogage de la configuration. Ces informations ne sont généralement par utilisées par les administrateurs.

Consignment des informations de contrôle

Voici les types de message susceptibles d'être générés :

CRIT	Problèmes liés à des défaillances de contrôle inopinées. Provoque l'arrêt du démon et exige une attention immédiate du service administratif.
ERR	Problèmes liés à une erreur de contrôle inopinée. Le service administratif devra éventuellement intervenir pour corriger cette erreur.
NOTICE	Messages relatifs aux transitions des régions de contrôles de ressources.
INFO	Messages relatifs aux statistiques d'utilisation des ressources.
DEBUG	Messages contenant des informations détaillées nécessaires lors du débogage du contrôle. Ces informations ne sont généralement par utilisées par les administrateurs.

Consignation des informations d'optimisation

Voici les types de message susceptibles d'être générés :

WARNING	Messages concernant des décisions importantes sur l'optimisation du système. Il est possible, par exemple, que les valeurs minimum et maximum des contraintes s'appliquant aux lots de ressources soient trop proches ou que le nombre de composants rattachés ne soit pas suffisant. Il est possible aussi que la réallocation optimale des ressources ne soit pas permise en raison de limitations inattendues. Le retrait du dernier processeur d'un jeu de processeurs contenant un consommateur de ressources restreint peut, par exemple, poser problème.
NOTICE	Messages relatifs à des configurations utilisables ou à des configurations impossibles à implémenter en raison d'historiques de décisions prioritaires.
INFO	Messages relatifs à d'autres possibilités de configurations.
DEBUG	Messages contenant des informations détaillées nécessaires lors du débogage de l'optimisation. Ces informations ne sont généralement par utilisées par les administrateurs.

Emplacement de journalisation

La propriété `system.pooled.log-location` permet de désigner l'emplacement réservé à la sortie consignée de `poold`. Vous pouvez spécifier un emplacement de type `SYSLLOG` pour la sortie `poold` (voir `syslog(3C)`).

En cas d'omission de cette propriété, l'emplacement par défaut de la sortie consignée de `poold` est `/var/log/pool/poold`.

Lorsque vous faites appel à `poold` à partir de la ligne de commande, cette propriété n'est pas utilisée. Les entrées du journal sont écrites dans `stderr` sur le terminal d'appel.

Gestion du journal avec `logadm`

Si la commande `poold` est active, le fichier `logadm.conf` offre une entrée permettant de gérer le fichier par défaut `/var/log/pool/poold`. Il s'agit de l'entrée suivante :

```
/var/log/pool/poold -N -s 512k
```

Reportez-vous aux pages de manuel `logadm(1M)` et `logadm.conf(4)`.

Mode de fonctionnement de l'allocation dynamique des ressources

Cette section décrit les processus et les facteurs utilisés par `poold` pour allouer les ressources de façon dynamique.

A propos des ressources disponibles

Il s'agit de l'ensemble des ressources destinées à être utilisées dans la portée du processus `poold`. La portée de contrôle équivaut au plus à une seule instance Oracle Solaris.

Sur un système dont les zones sont activées, la portée d'une instance d'exécution de `poold` se limite à la zone globale.

Détermination des ressources disponibles

Les pools de ressources englobent toutes les ressources système réservées aux applications.

Pour une seule instance d'exécution Oracle Solaris, il est indispensable d'allouer une ressource d'un type unique, telle qu'une CPU, à une seule et même partition. Il est possible de prévoir une ou plusieurs partitions par type de ressource. Chaque partition contient un jeu unique de ressources.

Voici une configuration possible pour une machine dotée de quatre CPU et deux jeux de processeurs :

```
pset 0: 0 1
```

```
pset 1 : 2 3
```

où les chiffres 0, 1, 2 et 3 après les deux-points représentent les ID des CPU. Notez que les deux jeux de processeurs comptent pour quatre CPU.

La même machine ne peut pas avoir la configuration suivante :

```
pset 0: 0 1
```

```
pset 1 : 1 2 3
```

En effet, la CPU 1 ne peut figurer que dans un seul jeu de processeurs à la fois.

L'accès aux ressources n'est pas possible à partir d'une partition autre que la partition d'appartenance.

Pour découvrir les ressources disponibles, `pool`d interroge la configuration de pools active afin de détecter les partitions. Toutes les ressources appartenant à l'ensemble des partitions sont additionnées pour déterminer la quantité totale de ressources disponibles pour chaque type de ressource sous contrôle.

Cette quantité correspond au chiffre de base utilisé par `pool`d lors de ses opérations. Il existe cependant des contraintes qui limitent la souplesse dont dispose `pool`d pour allouer les ressources. Pour plus d'informations au sujet des contraintes disponibles, reportez-vous à la section [“Contraintes de configuration” à la page 138](#).

Identification d'un manque de ressources

La portée du contrôle pour `pool`d est l'ensemble des ressources disponibles dont le partitionnement et la gestion sont la principale responsabilité de `pool`d. Cependant, d'autres mécanismes autorisés à manipuler les ressources conformément à cette portée de contrôle peuvent avoir une incidence sur une configuration. Si une partition échappe au contrôle alors que le démon `pool`d est actif, `pool`d essaie de rétablir le contrôle en manipulant les ressources disponibles de façon judicieuse. Si `pool`d ne parvient pas à localiser des ressources supplémentaires à sa portée, le démon indique un manque de ressource dans le journal.

Détermination de l'utilisation des ressources

`pool`d se consacre principalement à étudier l'utilisation des ressources qui sont dans sa portée. Cette surveillance permet de vérifier si les objectifs dépendants des charges de travail sont remplis.

Dans le cas d'un jeu de processeurs, par exemple, toutes les mesures nécessaires sont effectuées pour chacun des processeurs du jeu. L'utilisation des ressources montre le temps d'utilisation de la ressource par rapport à l'intervalle d'échantillonnage. Cette proportion est exprimée sous forme de pourcentage.

Identification des violations de contrôle

Les indications données à la section [“Contraintes et objectifs de configuration” à la page 137](#) permettent de détecter les conditions dans lesquelles un système risque de ne pas remplir ses objectifs. Ces objectifs sont directement liés à la charge de travail.

Une partition qui ne répond pas aux objectifs fixés par l'utilisateur est considérée comme une violation de contrôle. Il existe deux types de violation de contrôle : synchrone et asynchrone.

- Une violation synchrone d'un objectif est détectée par le démon au cours de la surveillance de la charge de travail.
- Une violation asynchrone d'un objectif se produit indépendamment de l'action de surveillance du démon.

Les événements suivants provoquent des violations de contrôle asynchrones :

- Ajout ou retrait des ressources d'une portée de contrôle.
- Reconfiguration de la portée de contrôle.
- Redémarrage du contrôleur de ressources poolld.

On considère que les contributions des objectifs ne dépendant pas de la charge de travail sont constantes entre chaque évaluation de l'objectif. La réévaluation de ce type d'objectif est déclenchée uniquement par le biais de l'une des violations de contrôle asynchrones.

Détermination de l'action corrective appropriée

Lorsque le contrôleur de ressources se rend compte qu'un consommateur de ressources manque de ressources, la réponse initiale consiste à augmenter les ressources en considérant que cela améliore les performances.

Les autres configurations remplissant les objectifs fixés dans le cadre de la configuration prévue pour la portée de contrôle sont ensuite examinés et évalués.

Ce processus est peaufiné à mesure que parviennent les résultats sur la surveillance du transfert des ressources et sur l'évaluation de la réactivité de la partition. L'historique des décisions est consulté pour éliminer les reconfigurations n'ayant pas apporté d'améliorations significatives dans le passé. Pour mieux juger de la pertinence des données d'historique, d'autres informations sont également prises en compte, comme les noms et les quantités de processus.

Si le démon ne parvient pas à prendre de mesure corrective, la condition d'erreur est consignée dans le journal. Pour plus d'informations, reportez-vous à la section [“Informations de journalisation poolld” à la page 143](#).

Utilisation de poolstat pour surveiller l'utilitaire des pools et l'utilisation des ressources

L'utilitaire poolstat sert à surveiller l'utilisation des ressources lorsque des pools sont activés sur votre système. Cet utilitaire examine en boucle tous les pools actifs sur un système et établit des statistiques en fonction du mode de sortie sélectionné. Les statistiques poolstat

vous permettent de savoir si les partitions de ressources sont exploitées de façon intensive. Il peut être intéressant de les analyser pour prendre les décisions qui s'imposent en matière de réallocation des ressources lorsque le système est soumis à une forte pression.

L'utilitaire poolstat inclut des options prévues spécialement pour examiner des pools spécifiques et obtenir des statistiques propres aux lots de ressources.

Si vous exécutez la commande poolstat dans une zone non globale et que des zones sont implémentées sur votre système, les informations relatives aux ressources associées au pool de la zone sont affichées.

Pour plus d'informations sur l'utilitaire poolstat, reportez-vous à la page de manuel [poolstat\(1M\)](#). Pour en savoir plus sur la tâche poolstat et sur son mode d'utilisation, reportez-vous à la section “Création d'un état statistique pour les ressources liées au pool à l'aide de poolstat” à la page 173.

Sortie poolstat

Dans le format de sortie par défaut, poolstat génère une ligne d'en-tête, puis affiche une ligne pour chaque pool. Une ligne de pool commence par l'ID du pool et le nom du pool, suivis d'une colonne de données statistiques pour le jeu de processeurs rattaché au pool. Les lots de ressources s'appliquant à plusieurs pools sont répertoriés autant de fois que cela est nécessaire, à raison d'un par pool.

Les en-têtes de colonne sont les suivants :

id	Code pool.
pool	Nom du pool.
rid	ID du lot de ressources.
rset	Nom du lot de ressources.
type	Type du lot de ressources.
min	Taille de ressource minimum.
max	Taille de ressource maximum.
size	Taille de ressource actuelle.
used	Mesure de la quantité du lot de ressources actuellement utilisée. Il s'agit plus précisément du pourcentage d'utilisation du lot de ressources multiplié par la taille du lot de ressources. Si un lot de ressources a

été reconfiguré au cours du dernier intervalle d'échantillonnage, il est possible que cette valeur ne soit pas signalée. Une valeur non reportée est affichée sous forme d'un tiret (-).

load Représentation absolue de la charge s'appliquant au lot de ressources. Pour plus d'informations sur cette propriété, reportez-vous à la page de manuel [libpool\(3LIB\)](#).

Vous pouvez configurer les éléments suivants dans la sortie `poolstat` :

- L'ordre des colonnes
- Les en-têtes affichés

Réglage des intervalles d'exécution des opérations `poolstat`

Il est possible de personnaliser les opérations réalisées par `poolstat`. Vous pouvez définir l'intervalle d'échantillonnage pour le rapport et spécifier le nombre de répétitions des statistiques :

interval Définissez les intervalles des opérations périodiques exécutées par `poolstat`. Tous les intervalles sont exprimés en secondes.

count Indiquez combien de fois vous souhaitez répéter les statistiques. Par défaut, `poolstat` génère une seule fois les statistiques.

Si vous omettez de spécifier *interval* et *count*, les statistiques sont établies une seule fois. Si vous définissez *interval* mais pas *count*, les statistiques sont reproduites indéfiniment.

Commandes utilisées avec l'utilitaire des pools de ressources

Les commandes présentées dans le tableau suivant représentent l'interface d'administration principale de l'utilitaire des pools. Pour plus d'informations sur l'utilisation de ces commandes dans un système sur lequel des zones sont activées, reportez-vous à la section “[Pools de ressources utilisés dans les zones](#)” à la page 130.

Référence aux pages de manuel	Description
pooladm(1M)	Active ou désactive l'utilitaire des pools sur votre système. Met en place une configuration particulière ou supprime la configuration actuelle et rétablit

Référence aux pages de manuel	Description
	l'état par défaut des ressources associées. Lorsqu'elle est exécutée sans option, <code>pooladm</code> imprime la configuration de pools dynamique actuelle.
poolbind(1M)	Active la liaison manuelle des projets, tâches et processus à un pool de ressources.
poolcfg(1M)	Permet de configurer les pools et les jeux. Les configurations créées à l'aide de cet outil sont instanciées sur un hôte cible au moyen de <code>pooladm</code>. Si vous associez l'argument de sous-commande <code>info</code> à l'option <code>-c</code>, <code>poolcfg</code> affiche des informations sur la configuration statique au niveau de <code>/etc/pooladm.conf</code>. En cas d'ajout d'un argument de nom de fichier, cette commande vous renseigne sur la configuration statique figurant dans le fichier désigné. Par exemple, <code>poolcfg -c info /tmp/newconfig</code> permet d'obtenir des informations au sujet de la configuration statique contenue dans le fichier <code>/tmp/newconfig</code>.
poolld(1M)	Démon système responsable des pools. Le démon tire parti des cibles système et des statistiques observables pour atteindre les objectifs de performances fixés par l'administrateur. S'il ne parvient pas à prendre une mesure corrective en cas de non respect des objectifs, <code>poolld</code> consigne cette information dans le journal.
poolstat(1M)	Affiche les statistiques s'appliquant aux ressources d'un pool. Cela facilite l'analyse des performances et permet aux administrateurs système de disposer d'un ensemble d'informations utiles pour procéder au partitionnement des ressources et à la répartition des tâches. Des options ont été prévues pour examiner les pools spécifiés et établir des statistiques propres aux ressources.

Une interface API de bibliothèque est fournie par `libpool` (reportez-vous à la page de manuel [libpool\(3LIB\)](#)). Cette bibliothèque peut être exploitée par les programmes pour manipuler les configurations de pools.

BETA
DRAFT

13

♦ ♦ ♦ C H A P I T R E 1 3

Création et administration des pools de ressources (tâches)

Ce chapitre décrit comment configurer et gérer des pools de ressources sur votre système.

Pour plus d'informations sur les pools de ressources, reportez-vous au [Chapitre 12, A propos des pools de ressources](#).

Administration des pools de ressources (liste des tâches)

Tâche	Description	Pour obtenir des instructions
Activation ou désactivation de pools de ressources	Activez ou désactivez les pools de ressources sur votre système.	“Activation et désactivation de l'utilitaire Pools” à la page 154
Activation ou désactivation de pools de ressources dynamiques	Activez ou désactivez les pools de ressources dynamiques sur votre système.	“Activation et désactivation de l'utilitaire Pools” à la page 154
Création d'une configuration de pools de ressources statique	Créez un fichier de configuration statique correspondant à la configuration dynamique actuelle. Pour plus d'informations, reportez-vous à la section “Structure des pools de ressources” à la page 131 .	“Création d'une configuration statique” à la page 159
Modification d'une configuration de pools de ressources	Redéfinissez une configuration de pools sur votre système en créant, par exemple, des pools supplémentaires.	“Modification d'une configuration” à la page 161
Association d'un pool de ressources à une classe de programmation	Associez un pool à une classe de programmation pour permettre à tous les processus liés de tirer parti de l'ordonnanceur indiqué.	“Association d'un pool avec une classe de programmation” à la page 163
Définition des contraintes et objectifs de configuration	Fixez des objectifs pour la commande <code>pool'd</code> lors d'une action corrective. Pour plus d'informations sur la configuration des objectifs, reportez-vous à la section “Présentation de <code>pool'd</code>” à la page 136 .	“Définition des contraintes de configuration” à la page 165 et “Etablissement des objectifs de configuration” à la page 165

Tâche	Description	Pour obtenir des instructions
Définition du niveau de consignation	Spécifiez le niveau des informations consignées dans le journal généré par pool.	“Définition du niveau de consignation pool” à la page 167
Utilisation d'un fichier texte avec la commande poolcfg	Exécutez la commande poolcfg de façon à tirer parti des entrées provenant d'un fichier texte.	“Utilisation des fichiers de commandes avec poolcfg” à la page 168
Transfert de ressources dans le noyau	Transfert de ressources dans le noyau telles que des ressources avec des ID spécifiques vers un jeu de destination.	“Transfert de ressources” à la page 169
Activation d'une configuration de pools	Activez la configuration dans le fichier de configuration par défaut.	“Activation d'une configuration de pools” à la page 170
Test d'une configuration de pools avant de valider la configuration	Simulez ce qui se produira après la validation de la configuration.	“Test d'une configuration avant sa validation” à la page 170
Suppression d'une configuration de pools du système	Toutes les ressources associées, telles que les jeux de processeurs, reprennent leur état par défaut.	“Suppression d'une configuration de pools” à la page 170
Liaison de processus à un pool	Associez manuellement un processus en cours d'exécution sur votre système à un pool de ressources.	“Liaison des processus à un pool” à la page 171
Liaison de tâches ou de projets à un pool	Associez des tâches ou des projets à un pool de ressources.	“Liaison de tâches ou de projets à un pool” à la page 172
Liaison de nouveaux processus à un pool de ressources	Pour lier automatiquement de nouveaux processus dans un projet à un pool donné, ajoutez un attribut à chaque entrée de la base de données project.	“Définition de l'attribut project.pool pour un projet” à la page 172
Utilisation d'attributs project pour lier un processus à un autre pool	Modifiez la liaison de pool pour les nouveaux processus démarrés.	“Liaison d'un processus à un autre pool grâce aux attributs project” à la page 172
Exécution de l'utilitaire poolstat pour produire des rapports	Créez plusieurs rapports à intervalles spécifiques.	“Création de plusieurs rapports à intervalles spécifiques” à la page 173
Création d'un état statistique sur un ensemble de ressources	Servez-vous de l'utilitaire poolstat pour obtenir des statistiques sur un ensemble de ressources d'un jeu de processeurs.	“Création d'un état statistique sur l'ensemble de ressources” à la page 174

Activation et désactivation de l'utilitaire Pools

Vous pouvez activer et désactiver les services de pools de ressources et de pools de ressources dynamiques sur le système à l'aide de la commande `svcadm` décrite à la page de manuel [svcadm\(1M\)](#).

Vous pouvez également faire appel à la commande `pooladm` décrite à la page de manuel [pooladm\(1M\)](#) pour réaliser les tâches suivantes :

- Activer l'utilitaire Pools de façon à pouvoir manipuler les pools
- Désactiver l'utilitaire Pools de façon à éviter la manipulation des pools

Remarque - Lors de la mise à niveau d'un système, si la structure de pools de ressources est activée et que le fichier `/etc/pooladm.conf` existe, le service de pools est automatiquement activé et la configuration définie dans le fichier est appliquée au système.

▼ Activation du service de pools de ressources à l'aide de svcadm

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Activez le service de pools de ressources.

```
# svcadm enable system/pools:default
```

▼ Désactivation du service de pools de ressources à l'aide de svcadm

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Désactivez le service de pools de ressources.

```
# svcadm disable system/pools:default
```

▼ Activation du service de pools de ressources dynamiques à l'aide de svcadm

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Activez le service de pools de ressources dynamiques.

```
# svcadm enable system/pools/dynamic:default
```

Exemple 13-1 Dépendance du service de pools de ressources dynamiques sur le service de pools de ressources

Cet exemple montre qu'il faut d'abord activer les pools de ressources pour exécuter DRP.

Il existe une dépendance entre les pools de ressources et les pools de ressources dynamiques. DRP est désormais un service dépendant des pools de ressources. Il peut être activé et désactivé indépendamment des pools de ressources.

L'affichage suivant montre que les pools de ressources et les pools de ressources dynamiques sont actuellement désactivés :

```
# svcs "*pool*"
STATE      STIME    FMRI
disabled   2011     svc:/system/pools:default
disabled   2011     svc:/system/pools/dynamic:default
```

Activer les pools de ressources dynamiques :

```
# svcadm enable svc:/system/pools/dynamic:default
# svcs -a | grep pool
STATE      STIME    FMRI
disabled   2011     svc:/system/pools:default
offline    2011     svc:/system/pools/dynamic:default
```

Notez que le service DRP est encore hors ligne.

Servez-vous de l'option -x de la commande svcs pour déterminer la raison pour laquelle le service DRP est hors ligne :

```
# svcs -x "*pool*"
svc:/system/pools:default (resource pools framework)
State: disabled since Sat Feb 12 02:36:15 2011
Reason: Disabled by an administrator.
See: http://support.oracle.com/msg/SMF-8000-05
See: libpool(3LIB)
See: pooladm(1M)
See: poolbind(1M)
See: poolcfg(1M)
See: poolstat(1M)
Impact: This service is not running.

svc:/system/pools/dynamic:default (dynamic resource pools)
State: disabled since Sat Feb 12 02:36:16 2011
Reason: Disabled by an administrator.
See: http://support.oracle.com/msg/SMF-8000-05
See: poold(1M)
Impact: This service is not running.
```

Activez le service de pools de ressources de façon à pouvoir exécuter le service DRP :

```
# svcadm enable svc:/system/pools:default
```

Lorsque la commande svcs "*pool*" est utilisée, le système s'affiche :

```
# svcs "*pool*"
STATE      STIME    FMRI
online      2011     svc:/system/pools/dynamic:default
online      2011     svc:/system/pools:default
```

Exemple 13-2 Effet des pools de ressources dynamiques lorsque le service de pools de ressources est désactivé

Si les deux services sont en ligne et que vous désactivez les pools de ressources :

```
# svcadm disable svc:/system/pools:default
```

Lorsque la commande `svcs "*pool*"` est utilisée, le système s'affiche :

```
# svcs "*pool*"
STATE      STIME    FMRI
disabled    2011     svc:/system/pools:default
online      2011     svc:/system/pools/dynamic:default
```

En revanche, le service `DRP` finit pas basculer `offline` car le service de pools de ressource a été désactivé :

```
# svcs "*pool*"
STATE      STIME    FMRI
disabled    2011     svc:/system/pools:default
offline     2011     svc:/system/pools/dynamic:default
```

Déterminez la raison pour laquelle le service `DRP` est hors ligne :

```
# svcs -x "*pool*"
svc:/system/pools:default (resource pools framework)
  State: disabled since Sat Feb 12 02:36:15 2011
  Reason: Disabled by an administrator.
    See: http://support.oracle.com/msg/SMF-8000-05
    See: libpool(3LIB)
    See: pooladm(1M)
    See: poolbind(1M)
    See: poolcfg(1M)
    See: poolstat(1M)
  Impact: 1 dependent service is not running. (Use -v for list.)

svc:/system/pools/dynamic:default (dynamic resource pools)
  State: offline since Sat Feb 12 02:36:15 2011
  Reason: Service svc:/system/pools:default is disabled.
    See: http://support.oracle.com/msg/SMF-8000-GE
    See: pool(1M)
    See: /var/svc/log/system-pools-dynamic:default.log
  Impact: This service is not running.
```

Il est indispensable de démarrer les pools de ressources pour tirer parti du service `DRP`. Vous pourriez, par exemple, lancer les pools de ressources à l'aide de la commande `pooladm` et de l'option `-e` :

```
# pooladm -e
```

La commande `svcs *pool*` affiche ensuite l'écran suivant :

```
# svcs "*pool*"
STATE      STIME    FMRI
online     2011    svc:/system/pools:default
online     2011    svc:/system/pools/dynamic:default
```

▼ Désactivation du service de pools de ressources dynamiques à l'aide de svcadm

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Désactivez le service de pools de ressources dynamiques.

```
# svcadm disable system/pools/dynamic:default
```

▼ Activation des pools de ressources à l'aide de pooladm

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Activez l'utilitaire Pools.

```
# pooladm -e
```

▼ Désactivation des pools de ressources à l'aide de pooladm

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Désactivez l'utilitaire Pools.

```
# pooladm -d
```

Affectation de CPU spécifique

Vous pouvez utiliser les sous-commandes `assign` et `unassign` pour affecter des UC, des noyaux et des sockets.

`cpus=` Liste de CPU affectées à la zone.

`cores=` Liste de noyaux affectés à la zone.

`sockets=` Liste de sockets affectés à la zone.

EXEMPLE 13-3 Affecter des noyaux

Affecter des noyaux au pset *new*.

```
# poolcfg -dc 'assign to pset new (core 0 ; core 1)'
```

EXEMPLE 13-4 Mettre à jour le pool en cours d'exécution pour l'adapter à la configuration statique persistante

Utilisez la commande `zonecfg` pour affecter des noyaux. Utilisez la commande `pooladm` avec l'option `-c` pour adapter les pools en cours d'exécution à la configuration statique.

```
# poolcfg -c 'assign to pset new (core 0 ; core 1)'  
# pooladm -c
```

Configuration des pools

▼ Création d'une configuration statique

Appliquez l'option `-s` à `/usr/sbin/pooladm` pour créer un fichier de configuration statique correspondant à la configuration dynamique actuelle et conserver ainsi les modifications au fil des réinitialisations. Sauf en cas de spécification d'un nom de fichier différent, il convient d'utiliser l'emplacement par défaut `/etc/pooladm.conf`.

Validez votre configuration à l'aide de la commande `pooladm` et de l'option `-c`. Exécutez ensuite la commande `pooladm` avec l'option `-s` pour mettre à jour la configuration statique en fonction de l'état de la configuration dynamique.

Remarque - Il est préférable d'utiliser la dernière fonctionnalité `pooladm -s` au lieu de la fonctionnalité précédente `poolcfg -c discover` pour créer une nouvelle configuration adaptée à la configuration dynamique.

Avant de
commencer

Activez les pools sur votre système.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**

2. **Mettez à jour le fichier de configuration statique pour qu'il corresponde à la configuration dynamique actuelle.**

```
# pooladm -s
```

3. **Affichez le contenu du fichier de configuration sous une forme lisible.**

Notez que la configuration contient des éléments par défaut créés par le système.

```
# poolcfg -c info
system tester
  string system.comment
  int system.version 1
  boolean system.bind-default true
  int system.poolid.pid 177916

  pool pool_default
    int pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int pool.importance 1
    string pool.comment
    pset pset_default

  pset pset_default
    int pset.sys_id -1
    boolean pset.default true
    uint pset.min 1
    uint pset.max 65536
    string pset.units population
    uint pset.load 10
    uint pset.size 4
    string pset.comment
    boolean testnullchanged true

  cpu
    int cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

  cpu
    int cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

  cpu
    int cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

  cpu
    int cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line
```

4. Validez la configuration dans `/etc/pooladm.conf`.

```
# pooladm -c
```

5. (Facultatif) Pour copier la configuration dynamique dans le fichier de configuration statique nommé `/tmp/backup`, entrez l'instruction suivante :

```
# pooladm -s /tmp/backup
```

▼ Modification d'une configuration

Pour optimiser votre configuration, créez un jeu de processeurs nommé `pset_batch` et un pool nommé `pool_batch`. Etablissez ensuite une association entre le pool et le jeu de processeurs.

N'oubliez pas d'insérer les arguments de sous-commande contenant un espace vide.

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Créez le jeu de processeurs `pset_batch`.

```
# poolcfg -c 'create pset pset_batch (uint pset.min = 2; uint pset.max = 10)'
```

3. Créez le pool `pool_batch`.

```
# poolcfg -c 'create pool pool_batch'
```

4. Etablissez une association entre le pool et le jeu de processeurs.

```
# poolcfg -c 'associate pool pool_batch (pset pset_batch)'
```

5. Affichez la configuration modifiée.

```
# poolcfg -c info
system tester
  string system.comment kernel state
  int    system.version 1
  boolean system.bind-default true
  int    system.poold.pid 177916

  pool pool_default
    int    pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int    pool.importance 1
    string pool.comment
    pset   pset_default

  pset pset_default
    int    pset.sys_id -1
    boolean pset.default true
```

```
uint    pset.min 1
uint    pset.max 65536
string  pset.units population
uint    pset.load 10
uint    pset.size 4
string  pset.comment
boolean testnullchanged true

cpu
    int    cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line

pool pool_batch
    boolean pool.default false
    boolean pool.active true
    int pool.importance 1
    string pool.comment
    pset pset_batch

pset pset_batch
    int pset.sys_id -2
    string pset.units population
    boolean pset.default true
    uint pset.max 10
    uint pset.min 2
    string pset.comment
    boolean pset.escapable false
    uint pset.load 0
    uint pset.size 0

cpu
    int    cpu.sys_id 5
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 4
    string cpu.comment
```

```
string cpu.status on-line
```

6. Validez la configuration dans `/etc/pooladm.conf`.

```
# pooladm -c
```

7. (Facultatif) Pour copier la configuration dynamique dans un fichier de configuration statique nommé `/tmp/backup`, entrez l'instruction suivante :

```
# pooladm -s /tmp/backup
```

▼ Association d'un pool avec une classe de programmation

Vous pouvez associer un pool à une classe de programmation pour permettre à tous les processus liés de tirer parti de l'ordonnanceur. Pour ce faire, donnez à la propriété `pool.scheduler` le nom de l'ordonnanceur. Cet exemple permet d'associer le pool `pool_batch` à l'ordonnanceur FSS.

1. Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.
2. Modifiez le pool `pool_batch` à associer à l'ordonnanceur FSS.

```
# poolcfg -c 'modify pool pool_batch (string pool.scheduler="FSS")'
```

3. Affichez la configuration modifiée.

```
# poolcfg -c info
system tester
  string system.comment
  int    system.version 1
  boolean system.bind-default true
  int    system.poold.pid 177916

  pool pool_default
    int    pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int    pool.importance 1
    string pool.comment
    pset   pset_default

  pset pset_default
    int    pset.sys_id -1
    boolean pset.default true
    uint   pset.min 1
    uint   pset.max 65536
    string pset.units population
    uint   pset.load 10
```

```
uint    pset.size 4
string  pset.comment
boolean testnullchanged true

cpu
    int    cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line

pool pool_batch
    boolean pool.default false
    boolean pool.active true
    int pool.importance 1
    string pool.comment
    string pool.scheduler FFS
    pset batch

pset pset_batch
    int pset.sys_id -2
    string pset.units population
    boolean pset.default true
    uint pset.max 10
    uint pset.min 2
    string pset.comment
    boolean pset.escapable false
    uint pset.load 0
    uint pset.size 0

cpu
    int    cpu.sys_id 5
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 4
    string cpu.comment
    string cpu.status on-line
```

4. Validez la configuration dans `/etc/pooladm.conf` :

```
# pooladm -c
```

5. **(Facultatif) Pour copier la configuration dynamique dans le fichier de configuration statique nommé /tmp/backup, entrez l'instruction suivante :**

```
# pooladm -s /tmp/backup
```

▼ Définition des contraintes de configuration

Les contraintes limitent l'étendue des configurations possibles en empêchant certaines modifications potentielles. Cette procédure montre comment définir la propriété `cpu.pinned`.

Dans les exemples qui suivent, `cpuid` représente un nombre entier.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Modifiez la propriété `cpu.pinned` dans la configuration statique ou dynamique :**
 - **Changez la configuration d'initialisation (statique) :**

```
# poolcfg -c 'modify cpu <cpuid> (boolean cpu.pinned = true)'
```
 - **Changez la configuration en cours d'exécution (dynamique) sans modifier la configuration d'initialisation :**

```
# poolcfg -dc 'modify cpu <cpuid> (boolean cpu.pinned = true)'
```

▼ Etablissement des objectifs de configuration

Vous pouvez fixer des objectifs pour la commande `poold` lors d'une action corrective.

Dans la procédure suivante, l'objectif `wt-load` est de faire en sorte que `poold` essaie de mettre en adéquation l'allocation des ressources et leur utilisation. L'objectif `locality` est désactivé pour faciliter la réalisation de cet objectif de configuration.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Modifiez le testeur système (tester) pour favoriser l'objectif `wt-load`.**

```
# poolcfg -c 'modify system tester (string system.pool objectives="wt-load")'
```
3. **Désactivez l'objectif `locality` pour le jeu de processeurs par défaut.**

```
# poolcfg -c 'modify pset pset_default (string pset.pool objectives="locality
none")'    one line
```

4. Désactivez l'objectif locality pour le jeu de processeurs pset_batch.

```
# poolcfg -c 'modify pset pset_batch (string pset.pool objectives="locality
none")'    one line
```

5. Affichez la configuration modifiée.

```
# poolcfg -c info
system tester
  string system.comment
  int    system.version 1
  boolean system.bind-default true
  int    system.poolid.pid 177916
  string system.poolid.objectives wt-load

  pool pool_default
    int    pool.sys_id 0
    boolean pool.active true
    boolean pool.default true
    int    pool.importance 1
    string pool.comment
    pset   pset_default

  pset pset_default
    int    pset.sys_id -1
    boolean pset.default true
    uint   pset.min 1
    uint   pset.max 65536
    string pset.units population
    uint   pset.load 10
    uint   pset.size 4
    string pset.comment
    boolean testnullchanged true
    string pset.poolid.objectives locality none

  cpu
    int    cpu.sys_id 3
    string cpu.comment
    string cpu.status on-line

  cpu
    int    cpu.sys_id 2
    string cpu.comment
    string cpu.status on-line

  cpu
    int    cpu.sys_id 1
    string cpu.comment
    string cpu.status on-line
```

```
cpu
    int    cpu.sys_id 0
    string cpu.comment
    string cpu.status on-line

pool pool_batch
    boolean pool.default false
    boolean pool.active true
    int pool.importance 1
    string pool.comment
    string pool.scheduler FSS
    pset batch

pset pset_batch
    int pset.sys_id -2
    string pset.units population
    boolean pset.default true
    uint pset.max 10
    uint pset.min 2
    string pset.comment
    boolean pset.escapable false
    uint pset.load 0
    uint pset.size 0
    string pset.poolid.objectives locality none

cpu
    int    cpu.sys_id 5
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 4
    string cpu.comment
    string cpu.status on-line
```

6. Validez la configuration dans /etc/pooladm.conf.

```
# pooladm -c
```

7. (Facultatif) Pour copier la configuration dynamique dans le fichier de configuration statique nommé /tmp/backup, entrez l'instruction suivante :

```
# pooladm -s /tmp/backup
```

▼ Définition du niveau de consignation poold

Pour spécifier le niveau des informations consignées dans le journal généré par poold, définissez la propriété `system.poolid.log-level` dans la configuration poold. La configuration poold est conservée au sein de la configuration libpool. Pour plus d'informations,

reportez-vous aux pages de manuel [“Informations de journalisation poold”](#) à la page 143, [poolcfg\(1M\)](#) et [libpool\(3LIB\)](#).

Vous pouvez également tirer parti de la commande poold sur la ligne de commande pour indiquer le niveau de consignation voulu dans le journal créé par poold.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Définissez le niveau de consignation en exécutant la commande poold avec l'option -l et un paramètre, par exemple, INFO.**

```
# /usr/lib/pool/poold -l INFO
```

Pour plus d'informations sur les paramètres disponibles, reportez-vous à la section [“Informations de journalisation poold”](#) à la page 143. Le niveau de consignation par défaut est NOTICE.

▼ Utilisation des fichiers de commandes avec poolcfg

La commande poolcfg associée à l'option -f accepte une entrée provenant d'un fichier texte dans lequel figurent les arguments de sous-commande poolcfg correspondant à l'option -c. Cette méthode est pratique pour effectuer une série d'opérations. Lors du traitement des commandes, la configuration n'est mise à jour que si toutes les commandes aboutissent. Dans le cas de configurations plus importantes ou plus complexes, il est préférable d'utiliser cette technique que de procéder par sous-commande.

Dans les fichiers de commandes, le caractère # signale un commentaire dans le reste de la ligne.

1. **Créez le fichier d'entrée poolcmds.txt.**

```
$ cat > poolcmds.txt
create system tester
create pset pset_batch (uint pset.min = 2; uint pset.max = 10)
create pool pool_batch
associate pool pool_batch (pset pset_batch)
```

2. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
3. **Exécutez la commande :**

```
# /usr/sbin/poolcfg -f poolcmds.txt
```

Transfert de ressources

Associez l'argument de sous-commande `transfer` à l'option `-c` (lorsque vous exécutez la commande `poolcfg` avec l'option `-d`) pour transférer les ressources dans le noyau. L'option `-d` permet à la commande de fonctionner directement au niveau du noyau et évite de prendre en compte les entrées provenant d'un fichier.

La procédure suivante déplace deux CPU du jeu de processeurs `pset1` vers le jeu de processeurs `pset2` dans le noyau.

▼ Transfert de CPU entre les jeux de processeurs

1. Connectez-vous en tant qu'utilisateur `root` ou prenez un rôle équivalent.
2. Transférez deux UC de `pset1` à `pset2`.

L'ordre d'utilisation des sous-clauses `from` et `to` n'a pas d'importance. Une seule sous-clause `to` et `from` est acceptée par commande.

```
# poolcfg -dc 'transfer 2 from pset pset1 to pset2'
```

Exemple 13-5 Autre méthode pour déplacer des CPU entre jeux de processeurs

S'il est nécessaire de transférer des ID connus d'un type de ressource, une autre syntaxe a été prévue. La commande suivante permet, par exemple, d'allouer deux CPU identifiées par les ID 0 et 2 au jeu de processeurs `pset_large` :

```
# poolcfg -dc 'transfer to pset pset_large (cpu 0; cpu 2)'
```

Dépannage

Si les ressources sont insuffisantes pour satisfaire la demande ou si les ID spécifiés ne peuvent pas être localisés, le transfert échoue et le système affiche un message d'erreur.

Activation et suppression des configurations de pools

Utilisez la commande `pooladm` pour rendre une configuration de pools active ou pour supprimer la configuration active. Pour plus d'informations sur cette commande, reportez-vous à la page de manuel [pooladm\(1M\)](#).

▼ Activation d'une configuration de pools

Pour activer la configuration dans le fichier de configuration par défaut, `/etc/pooladm.conf`, invoquez `pooladm` avec l'option `-c`, "valider la configuration."

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Validez la configuration dans `/etc/pooladm.conf`.**

```
# pooladm -c
```

3. **(Facultatif) Copiez la configuration dynamique dans un fichier de configuration statique, `/tmp/backup` par exemple.**

```
# pooladm -s /tmp/backup
```

▼ Test d'une configuration avant sa validation

En combinant l'option `-n` et l'option `-c`, vous avez la possibilité de simuler le résultat que vous obtiendrez après validation. Cela n'a pas pour effet de valider la configuration.

La commande suivante essaie de tester la configuration figurant dans `/home/admin/newconfig`. Les conditions d'erreur rencontrées sont affichées, mais la configuration n'est pas modifiée.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Testez la configuration avant de la valider.**

```
# pooladm -n -c /home/admin/newconfig
```

▼ Suppression d'une configuration de pools

Pour supprimer la configuration active et rétablir le statut par défaut des ressources associées, telles que les jeux de processeurs, utilisez l'option `-x` pour "suppression de la configuration."

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Supprimez la configuration active.**

```
# pooladm -x
```

L'option `-x` associée à `pooladm` a pour effet d'effacer tous les éléments définis par l'utilisateur de la configuration dynamique. Les ressources reprennent toutes leur état par défaut et l'ensemble des liaisons de pools sont remplacées par une liaison vers le pool par défaut.

Mélange des classes de programmation au sein d'un jeu de processeurs

Vous pouvez combiner, en toute sécurité, des processus des classes TS et AI au sein du même jeu de processeurs. Le mélange d'autres classes de programmation au sein d'un même jeu de processeurs peut produire des résultats imprévisibles. Si l'instruction `pooladm -x a` pour effet de combiner des classes de processeurs au sein d'un même jeu de processeurs, utilisez la commande `priocntl` pour transférer les processus en cours vers une autre classe de programmation. Pour ce faire, reportez-vous à la section [“Transfert manuel de processus de la classe TS vers la classe FSS” à la page 104](#). Reportez-vous également à la page de manuel [priocntl\(1\)](#).

Définition des attributs des pools et liaison à un pool

Vous pouvez définir un attribut `project.pool` afin d'associer un pool de ressources à un projet.

Vous disposez de deux modes de liaison d'un processus en cours à un pool :

- Vous pouvez utiliser la commande `poolbind` décrite à la page [poolbind\(1M\)](#) pour lier un processus spécifique à un pool de ressources nommé.
- Servez-vous de l'attribut `project.pool` de la base de données `project` pour identifier la liaison de pools pour une nouvelle session de connexion ou une tâche lancée au moyen de la commande `newtask`. Reportez-vous aux pages de manuel [newtask\(1\)](#), [projmod\(1M\)](#), et [project\(4\)](#).

▼ Liaison des processus à un pool

La procédure suivante utilise la commande `poolbind` avec l'option `-p` pour établir une liaison manuelle entre un processus (le shell actuel, dans le cas présent) et un pool nommé `ohare`.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Liez un processus à un pool de façon manuelle :**

```
# poolbind -p ohare $$
```

3. **Vérifiez la liaison du pool pour le processus en exécutant la commande `poolbind` avec l'option `-q`.**

```
$ poolbind -q $$  
155509 ohare
```

Le système affiche l'ID du processus et la liaison du pool.

▼ Liaison de tâches ou de projets à un pool

Pour lier des tâches ou des projets à un pool, exécutez la commande `poolbind` avec l'option `-i`. L'exemple suivant permet de lier tous les processus du projet `airmiles` au pool `laguardia`.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Liez tous les processus du projet `airmiles` au pool `laguardia`.**

```
# poolbind -i project -p laguardia airmiles
```

▼ Définition de l'attribut `project.pool` pour un projet

Vous pouvez définir l'attribut `project.pool` afin de lier les processus d'un projet à un pool de ressources.

1. **Connectez-vous en tant qu'utilisateur root ou prenez un rôle équivalent.**
2. **Ajoutez un attribut `project.pool` à chaque entrée de la base de données `project`.**

```
# projmod -a -K project.pool=poolname project
```

▼ Liaison d'un processus à un autre pool grâce aux attributs `project`

Supposons que vous disposez d'une configuration constituée de deux pools nommés `studio` et `backstage`. Le fichier `/etc/project` contient les données suivantes :

```
user.paul:1024:::project.pool=studio
user.george:1024:::project.pool=studio
user.ringo:1024:::project.pool=backstage
passes:1027::paul::project.pool=backstage
```

Sous cette configuration, les processus lancés par l'utilisateur `paul` sont liés par défaut au pool `studio`.

L'utilisateur `paul` a la possibilité de modifier la liaison du pool pour les processus qu'il se charge de démarrer. `paul` peut utiliser la commande `newtask` pour lier également le travail au pool `backstage` en lançant le projet `passes`.

1. **Démarrez un processus dans le projet `passes`.**

```
$ newtask -l -p passes
```

2. Exécutez la commande `poolbind` avec l'option `-q` pour vérifier la liaison du pool pour le processus. Servez-vous également du symbole double dollar (`$$`) pour transmettre le numéro de processus du shell parent à la commande.

```
$ poolbind -q $$
6384 pool backstage
```

Le système affiche l'ID du processus et la liaison du pool.

Création d'un état statistique pour les ressources liées au pool à l'aide de poolstat

La commande `poolstat` permet d'obtenir des statistiques sur les ressources ayant trait au pool. Pour plus d'informations, reportez-vous à [“Utilisation de poolstat pour surveiller l'utilitaire des pools et l'utilisation des ressources” à la page 148](#) et à la page de manuel `poolstat(1M)`.

Les sous-sections suivantes illustrent par des exemples le mode de création de divers rapports.

Affichage de la sortie poolstat par défaut

Le fait d'exécuter la commande `poolstat` sans argument génère une ligne d'en-tête et une ligne d'informations pour chaque pool. La ligne d'informations contient l'ID du pool, le nom du pool et les statistiques sur les ressources pour le jeu de processeurs associé au pool.

```
machine% poolstat
```

		pset		
id	pool	size	used	load
0	pool_default	4	3.6	6.2
1	pool_sales	4	3.3	8.4

Création de plusieurs rapports à intervalles spécifiques

La commande suivante permet d'obtenir trois rapports à intervalle de 5 secondes.

```
machine% poolstat 5 3
```

		pset		
id	pool	size	used	load
46	pool_sales	2	1.2	8.3

```

0 pool_default      2  0.4  5.2
                    pset
id pool            size used load
46 pool_sales      2  1.4  8.4
0 pool_default      2  1.9  2.0
                    pset
id pool            size used load
46 pool_sales      2  1.1  8.0
0 pool_default      2  0.3  5.0

```

Création d'un état statistique sur l'ensemble de ressources

L'exemple suivant a recours à la commande `poolstat` et à l'option `-r` pour produire un état statistique sur les ressources du jeu de processeurs. Comme l'ensemble de ressources `pset_default` est lié à plusieurs pools, le jeu de processeurs n'apparaît qu'une seule fois dans la liste pour chaque appartenance au pool.

```

machine% poolstat -r pset
  id pool      type rid rset      min  max size used load
  0 pool_default pset -1 pset_default  1 65K  2  1.2  8.3
  6 pool_sales  pset  1 pset_sales    1 65K  2  1.2  8.3
  2 pool_other  pset -1 pset_default  1 10K  2  0.4  5.2

```

14

♦ ♦ ♦ C H A P I T R E 1 4

Exemple de configuration de la gestion des ressources

Ce chapitre vous propose de découvrir la structure de gestion des ressources à travers un projet de consolidation de serveur.

Ce chapitre comprend les sections suivantes :

- “Configuration à consolider” à la page 175
- “Configuration de la consolidation” à la page 176
- “Création de la configuration” à la page 176
- “Visualisation de la configuration” à la page 177

Configuration à consolider

Dans cet exemple, cinq applications sont consolidées dans un seul système. Les besoins en ressources des applications en question sont variables. Leurs utilisateurs et leurs architectures sont également différents. Chaque application fait actuellement partie d'un serveur dédié conçu pour répondre à leurs exigences. Le tableau suivant donne un aperçu de ces applications et de leurs caractéristiques.

Description de l'application	Caractéristiques
Serveur d'applications	Evolutivité négative au-delà de 2 CPU
Instance de base de données pour le serveur d'applications	Traitement intensif de transactions
Serveur d'applications dans un environnement de test et de développement	Exécution de code non testé depuis une interface graphique
Serveur de traitement de transactions	Temps de réponse primordial
Instance de base de données autonome	Traitement d'un grand nombre de transactions et gestion de plusieurs fuseaux horaires

Configuration de la consolidation

La configuration suivante a pour but de consolider les applications au sein d'un seul système dont les pools de ressources et les pools de ressources dynamiques sont activés.

- Le serveur d'applications possède un jeu de processeurs à double CPU.
- L'instance de base de données pour le serveur d'applications et l'instance de base de données autonome sont consolidées sur un seul jeu de processeurs doté d'au moins quatre CPU. L'instance de base de données autonome a la garantie d'obtenir 75 % des ressources.
- Le serveur d'applications de test et de développement a besoin de la classe de programmation AI pour garantir le taux de réponse de l'interface utilisateur. L'utilisation de la mémoire est soumise à des restrictions pour atténuer les effets de générations de code incorrectes.
- Un jeu de processeurs dédié d'au moins deux CPU est alloué au serveur de traitement des transactions pour minimiser le temps de latence des réponses.

Cette configuration couvre les applications connues exécutant et consommant des cycles de processeur dans chaque lot de ressources. Il est donc possible d'établir des contraintes afin de transférer les ressources du processeur vers les jeux qui en font la demande.

- Pour accorder des allocations plus importantes aux lots de ressources utilisés de façon intensive, l'objectif `wt-load` a été défini.
- L'objectif `locality` est réglé sur `tight`, qui est utilisé pour maximiser le voisinage du processeur.

Il a fallu également prévoir une contrainte supplémentaire pour que l'utilisation du lot de ressources ne dépasse par le seuil de 80 %. Cela permet de s'assurer que les applications ont accès aux ressources dont elles ont besoin. En ce qui concerne le jeu de processeurs de transaction, l'objectif de maintenir l'utilisation en dessous des 80 % est deux fois plus importante que tout autre objectif fixé. Cela sera pris en compte dans la configuration.

Création de la configuration

Modifiez le fichier de base de données `/etc/project`. Ajoutez des entrées afin d'implémenter les contrôles de ressources nécessaires et de mapper les utilisateurs aux pools de ressources, puis visualisez le fichier.

```
# cat /etc/project
.
.
.
user.app_server:2001:Production Application Server::project.pool=appserver_pool
user.app_db:2002:App Server DB::project.pool=db_pool;project.cpu-shares=(privileged,1,deny)
development:2003:Test and development::staff:project.pool=dev_pool;
process.max-address-space=(privileged,536870912,deny)    keep with previous line
user.tp_engine:2004:Transaction Engine::project.pool=tp_pool
```

```
user.geo_db:2005:EDI DB:::project.pool=db_pool;project.cpu-shares=(privileged,3,deny)
.
.
.
```

Remarque - L'équipe de développement est tenue d'exécuter les tâches prévues dans le projet de développement, car l'accès au projet est basé sur un ID de groupe d'utilisateurs (GID).

Créez un fichier d'entrée nommé `pool.host` qui servira à configurer les pools de ressources nécessaires. Visualisez le fichier.

```
# cat pool.host
create system host
create pset dev_pset (uint pset.min = 0; uint pset.max = 2)
create pset tp_pset (uint pset.min = 2; uint pset.max=8)
create pset db_pset (uint pset.min = 4; uint pset.max = 6)
create pset app_pset (uint pset.min = 1; uint pset.max = 2)
create pool dev_pool (string pool.scheduler="IA")
create pool appserver_pool (string pool.scheduler="TS")
create pool db_pool (string pool.scheduler="FSS")
create pool tp_pool (string pool.scheduler="TS")
associate pool dev_pool (pset dev_pset)
associate pool appserver_pool (pset app_pset)
associate pool db_pool (pset db_pset)
associate pool tp_pool (pset tp_pset)
modify system tester (string system.poolid.objectives="wt-load")
modify pset dev_pset (string pset.poolid.objectives="locality tight; utilization < 80")
modify pset tp_pset (string pset.poolid.objectives="locality tight; 2: utilization < 80")
modify pset db_pset (string pset.poolid.objectives="locality tight;utilization < 80")
modify pset app_pset (string pset.poolid.objectives="locality tight; utilization < 80")
```

Mettez à jour la configuration à l'aide du fichier d'entrée `pool.host`.

```
# poolcfg -f pool.host
```

Rendez la configuration active.

```
# pooladm -c
```

La structure est maintenant opérationnelle sur le système.

Activez DRP.

```
# svcadm enable pools/dynamic:default
```

Visualisation de la configuration

Pour afficher la configuration de la structure, laquelle contient également des éléments par défaut créés par le système, entrez l'instruction suivante :

```
# pooladm
```

```
system host
  string system.comment
  int    system.version 1
  boolean system.bind-default true
  int    system.poold.pid 177916
  string system.poold.objectives wt-load

pool dev_pool
  int    pool.sys_id 125
  boolean pool.default false
  boolean pool.active true
  int    pool.importance 1
  string pool.comment
  string pool.scheduler IA
  pset   dev_pset

pool appserver_pool
  int    pool.sys_id 124
  boolean pool.default false
  boolean pool.active true
  int    pool.importance 1
  string pool.comment
  string pool.scheduler TS
  pset   app_pset

pool db_pool
  int    pool.sys_id 123
  boolean pool.default false
  boolean pool.active true
  int    pool.importance 1
  string pool.comment
  string pool.scheduler FSS
  pset   db_pset

pool tp_pool
  int    pool.sys_id 122
  boolean pool.default false
  boolean pool.active true
  int    pool.importance 1
  string pool.comment
  string pool.scheduler TS
  pset   tp_pset

pool pool_default
  int    pool.sys_id 0
  boolean pool.default true
  boolean pool.active true
  int    pool.importance 1
  string pool.comment
  string pool.scheduler TS
  pset   pset_default

pset dev_pset
  int    pset.sys_id 4
```

```
string pset.units population
boolean pset.default false
uint   pset.min 0
uint   pset.max 2
string pset.comment
boolean pset.escapable false
uint   pset.load 0
uint   pset.size 0
string pset.pool objectives locality tight; utilization < 80

pset tp_pset
int     pset.sys_id 3
string  pset.units population
boolean pset.default false
uint    pset.min 2
uint    pset.max 8
string  pset.comment
boolean pset.escapable false
uint    pset.load 0
uint    pset.size 0
string  pset.pool objectives locality tight; 2: utilization < 80

cpu
int     cpu.sys_id 1
string  cpu.comment
string  cpu.status on-line

cpu
int     cpu.sys_id 2
string  cpu.comment
string  cpu.status on-line

pset db_pset
int     pset.sys_id 2
string  pset.units population
boolean pset.default false
uint    pset.min 4
uint    pset.max 6
string  pset.comment
boolean pset.escapable false
uint    pset.load 0
uint    pset.size 0
string  pset.pool objectives locality tight; utilization < 80

cpu
int     cpu.sys_id 3
string  cpu.comment
string  cpu.status on-line

cpu
int     cpu.sys_id 4
string  cpu.comment
string  cpu.status on-line
```

```
cpu
    int    cpu.sys_id 5
    string cpu.comment
    string cpu.status on-line

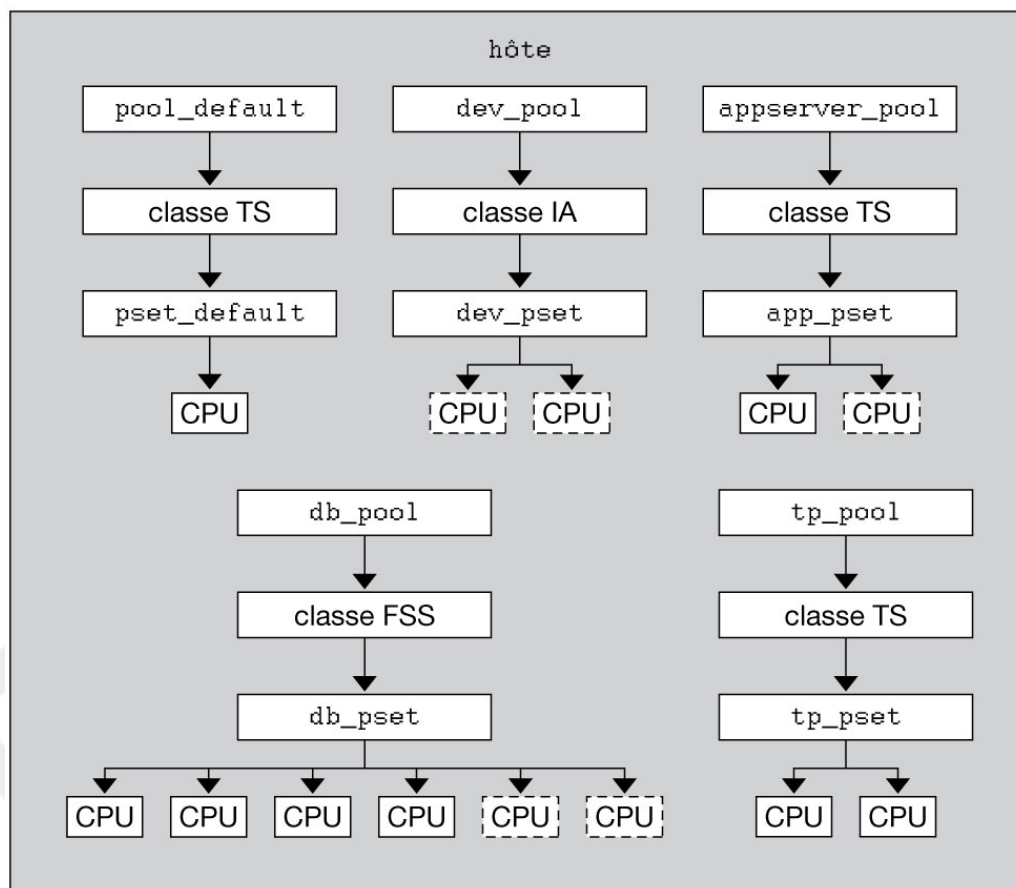
cpu
    int    cpu.sys_id 6
    string cpu.comment
    string cpu.status on-line

pset app_pset
    int    pset.sys_id 1
    string pset.units population
    boolean pset.default false
    uint   pset.min 1
    uint   pset.max 2
    string pset.comment
    boolean pset.escapable false
    uint   pset.load 0
    uint   pset.size 0
    string pset.pool.objectives locality tight; utilization < 80
    cpu
        int    cpu.sys_id 7
        string cpu.comment
        string cpu.status on-line

pset pset_default
    int    pset.sys_id -1
    string pset.units population
    boolean pset.default true
    uint   pset.min 1
    uint   pset.max 4294967295
    string pset.comment
    boolean pset.escapable false
    uint   pset.load 0
    uint   pset.size 0
    cpu
        int    cpu.sys_id 0
        string cpu.comment
        string cpu.status on-line
```

Voici une représentation graphique de la structure.

FIGURE 14-1 Configuration de la consolidation du serveur



Remarque - Dans le pool `db_pool`, l'instance de base de données autonome a la garantie d'obtenir 75 % des ressources de la CPU.

BETA
DRAFT

Index

Nombres et symboles

/etc/project
 fichier, 23
 format d'entrée, 25
 /etc/user_attr fichier, 22
 /var/adm/exacct annuaire, 47

A

acctadm, commande, 53
 activation de la comptabilisation étendue, 52
 activation de la limitation des ressources, 120
 activation des pools de ressources, 154
 activation des pools de ressources dynamiques, 154
 administration des pools de ressources, 150
 Affichage de l'état de la comptabilisation étendue, 53
 attribut
 project.pool, 134

C

Classes de programmation, 98
 commandes
 comptabilisation étendue, 48
 contrôles de ressources, 74
 projets et tâches, 29
 Commandes
 Ordonnanceur FSS, 99
 Communication interprocessus (IPC) Voir Contrôles de ressources
 Comptabilisation étendue
 Présentation, 43
 comptabilisation étendue
 activation, 52
 commandes, 48

 état, affichage, 53
 format de fichier, 44
 imputation, 44
 SMF, 47
 configuration
 rcapd, 109
 configuration des contrôles de ressources, 61
 configuration des parts de CPU, 94
 configuration des pools dynamiques, 131
 Consolidation serveur, 17
 Contrôles de ressources
 Communication interprocessus (IPC), 60
 Valeurs de seuil, 68
 contrôles de ressources
 actions globales, 68
 actions locales, 61, 69
 configuration, 61
 inf, valeur, 72
 mise à jour temporaire, 73
 modification temporaire, 74
 valeurs de seuil, 61, 69
 contrôles des ressources
 définition, 59
 liste de, 62
 présentation, 59
 création de pools de ressources, 135

D

Définition des attributs de pools de ressources, 171
 démon d'allocation restrictive des ressources, 107
 désactivation de la limitation des ressources, 121
 désactivation des pools de ressources, 154
 désactivation des pools de ressources dynamiques, 154
 DRP, 129

E

exacct, fichier, 44

F

format d'entrée

/etc/project, fichier, 25

FSS *Voir* Ordonnanceur FSS

G

gestion des ressources

contraintes, 15

définition, 13

partitionnement, 16

programmation, 16

tâches, 28

I

implémentation des pools de ressources, 133

Interface Perl, 48

J

jeu de processeurs par défaut, 129

L

liaison à un pool de ressources, 171

libexacct, bibliothèque, 44

limitation des ressources, 107

activation, 120

désactivation, 121

limites d'utilisation des ressources, 60

M

mise à jour temporaire des contrôles de ressources, 73

modification temporaire des contrôles de ressources,

74

module d'authentification enfichable *Voir* PAM

N

niveaux de privilège, valeurs de seuil, 67

O

Ordonnanceur FSS, 88

Définition d'une part, 88

Jeux de processeurs, 95

project.cpu-shares, 88

ordonnanceur FSS

configuration, 103

P

PAM (module d'authentification enfichable)

gestion des identités, 24

pool de ressources par défaut, 129

poold

allocation dynamique des ressources, 129

composants configurables, 142

contraintes, 138

cpu-pinned propriété, 138

description, 136

Informations de consignation, 143

objectifs, 139

portée du contrôle, 147

violation de contrôle asynchrone, 147

violation de contrôle synchrone, 147

pools, 128

pools de ressources, 128

/etc/pooladm.conf, 131

activation, 154

activation de la configuration, 170

administration, 150

configuration des pools statiques, 131

création, 135

désactivation, 154

éléments de configuration, 132

implémentation, 133

liaison à, 171

propriétés, 133

Reconfiguration dynamique, 134
 suppression , 170
 suppression d'une configuration, 170
pools de ressources dynamiques
 activation, 154
 désactivation, 154
poolstat
 description, 148
 Exemples d'utilisation, 173
 format de sortie, 149
project, base de données, 23
project.cpu-shares, 94
project.pool, attribut, 134
projet
 définition, 22
 état actif, 89
 état inactif, 89
 sans parts, 88
projet 0, 95
Projet par défaut, 22
projet system Voir projet 0
putacct appel système, 46

R

rcap.max-rss, attribut, 109
rcapadm, commande, 109
rcapd
 configuration, 109
 intervalles d'analyse, 113
 Intervalles d'échantillonnage, 113
rcapd, démon, 107
rcapstat, commande, 114
rctls, 59 Voir contrôles des ressources
rlimits Voir limites d'utilisation des ressources

S

seuil d'allocation restrictive de la mémoire, 110
suppression des pools de ressources, 170

T

tâches

gestion des ressources, 28

V

valeurs de seuil, contrôles de ressources, 67

BETA
DRAFT