



---

Guide d'administration du système  
Solaris Resource Manager 1.0 pour  
Solaris 2.6 (Édition plateforme  
SPARC)

---

Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303  
U.S.A.

Référence : 806-0052-10  
Mars 1999

---

Copyright 1998 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, Californie 94303-4900 Etats-Unis. Tous droits réservés.

ShareII Copyright 1989-1998 Softway Pty. Limited. Tous droits réservés.

Ce produit ou document est protégé par des droits d'auteur et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et, le cas échéant, de ses bailleurs de licence. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par des droits d'auteur et licencié par des fournisseurs de Sun.

Certaines parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunDocs, Sun Enterprise, Sun Enterprise SyMON, Solaris, Solaris Resource Manager, AnswerBook, NFS, et docs.sun.com sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. ShareII est une marque de fabrique ou une marque déposée de Softway Pty. Limited aux États-Unis et dans d'autres pays.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE EN L'ÉTAT ET AUCUNE GARANTIE, EXPRESSE OU TACITE, N'EST ACCORDÉE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION À RÉPONDRE À UNE UTILISATION PARTICULIÈRE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DÉNI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OÙ IL SERAIT TENU JURIDIQUEMENT POUR NUL ET NON AVENU.



# Sommaire

---

**Préface ix**

**1. Aperçu 1**

Introduction à Solaris Resource Manager 1

Atteinte des objectifs de l'organisation 1

Quand utiliser Solaris Resource Manager 2

Caractéristiques principales 3

Lien avec les autres fonctions de contrôle des ressources de Solaris 4

Différences entre Solaris Resource Manager et les produits similaires 7

**2. Utilisations normales 9**

Aperçu des noeuds limites 9

Structure hiérarchique 9

Limites hiérarchiques 10

Processus 10

Contrôle des ressources 11

Gestion des ressources de l'UC 13

Mémoire virtuelle (limites par utilisateur et par processus) 14

Nombre de processus 14

Terminaux et temps de connexion 15

Administration des utilisateurs 15

	Aperçu des données d'usage	15
	Exemples	16
	Exemple de regroupement de serveurs	16
	Ajout d'un utilisateur d'application de calcul par lots	19
	Mise en place d'un processus Web frontal	21
	Ajout d'utilisateurs ayant des besoins de mémoire spéciaux	22
	Partage d'une machine entre plusieurs services	24
	Serveur d'applications type	27
<b>3.</b>	<b>Configuration</b>	<b>33</b>
	Paramètres d'initialisation du noyau	33
	Configuration de démarrage multi-utilisateur	35
	Paramètres globaux de Solaris Resource Manager - commande <code>srmadm</code>	36
	Désactivation de Solaris Resource Manager	38
	Options de la commande <code>limdaemon</code>	39
	Sous-système PAM	40
	Gestion des comptes	41
	Scripts	41
	Interaction du module PAM avec des groupes de périphériques	43
	Gestion des sessions	43
<b>4.</b>	<b>Procédure d'initialisation</b>	<b>45</b>
	Initialisation sans Solaris Resource Manager	45
	Événements de la séquence d'initialisation	46
	Processus démons du système	47
	Activation de Solaris Resource Manager à l'aide de la commande <code>srmadm</code>	48
	Lancement du démon Solaris Resource Manager	49
<b>5.</b>	<b>Gestion des noeuds limites</b>	<b>51</b>
	Administration déléguée	51

Sécurité	53
Indicateurs <i>uselimadm</i> et <i>admin</i>	53
Structure suggérée de noeud limite sous-administrateur	54
Base de données des limites	55
Création de la base de données des limites	56
Enregistrement et restauration de la base de données des limites	56
Création et suppression de noeuds limites	57
Programmes de maintenance des noeuds limites	59
Unités	60
Conversions	61
Commande <i>limadm</i>	61
Commande <i>liminfo</i>	62
Commande <i>limreport</i>	62
Manipulation de noeuds limites	62
Commandes <i>limreport</i> et <i>limadm</i>	62
Copie et déplacement de noeuds limites	63
<b>6. Ordonnanceur SHR</b>	<b>65</b>
Description technique	65
Parts	65
Part attribuée	66
Usage et décroissance	67
Usage cumulatif	67
Part effective	67
Priorité de partage par processus ( <i>sharepri</i> )	67
Exemple d'attribution de parts	68
Structure de l'arbre d'ordonnement	68
Description de l'arbre	68
Calcul de la part attribuée	69

Relation entre Solaris Resource Manager et la fonction nice de Solaris	70
Reconfiguration dynamique	70
Noeud limite inactif	71
Noeud limite autre	71
Noeud limite perdu	72
<b>7. Limites de mémoire, de mémoire de processus et de comptage de processus</b>	<b>73</b>
Attributs pour le contrôle de la mémoire virtuelle (total)	73
Attribut pour le contrôle de la mémoire de processus (par processus)	74
Description technique des limites de mémoire	74
Reconfiguration dynamique et limites de mémoire virtuelle	75
Attributs pour le contrôle du comptage de processus	76
Description technique du comptage de processus	76
<b>8. Données d'usage</b>	<b>77</b>
Attributs cumulatifs	77
Facturation	77
Commande liminfo	78
Commande limreport	78
Commande limadm	79
<b>9. Dépannage</b>	<b>81</b>
Un utilisateur ne peut pas ouvrir une session	81
Un utilisateur n'est pas informé lorsqu'il atteint ses limites	82
Impossible de changer le groupe d'un utilisateur	82
Base de données des limites endommagée	83
Temps de connexion d'un terminal non actualisé	83
Des utilisateurs dépassent souvent des limites	84
Le système est lent	84
Processus du noeud limite racine	84

Ressources d'unité centrale non contrôlées par Solaris Resource Manager	85
Messages inhabituels	86
Noeuds limites orphelins	86
Boucles de groupe	87
Causes	88
Effets	88
Correction	88
Reprise après une panne	89
<b>10. Messages d'erreur</b>	<b>93</b>
<b>A. Exemples de script Solaris Resource Manager</b>	<b>95</b>
Script de démarrage	95
Script "pas de noeud limite" par défaut	99
<b>Glossaire</b>	<b>101</b>





# Préface

---

Ce guide est destiné aux administrateurs de système qui sont responsables de la configuration et de l'administration du logiciel Solaris Resource Manager™ 1.0 dans l'environnement d'exploitation Solaris™.

---

## Avant de lire ce manuel

Avant d'utiliser ce manuel, vous devriez vous familiariser avec la collection de manuels AnswerBook™ ci-dessous, disponibles au site [docs.sun.com](http://docs.sun.com).

- *Collection pour Administrateur de système Solaris 2.6, Vol 1*
- *Manuel AnswerBook pour Administrateur de système Solaris 2.6, Vol 2*

---

## Structure du guide

Ce manuel comporte des chapitres, une annexe et un glossaire.

Chapitre 1 Présente une introduction à Solaris Resource Manager et décrit comment ce produit peut être utilisé pour attribuer et commander les principales ressources système.

Chapitre 2 Décrit le fonctionnement du logiciel Solaris Resource Manager et donne des exemples de son utilisation.

Chapitre 3 Explique comment configurer le logiciel Solaris Resource Manager dans votre système Solaris.

Chapitre 4 Décrit les effets de la procédure d'initialisation d'UNIX™ sur le produit Solaris Resource Manager.

Chapitre 5 Décrit la structure "par utilisateur" introduite dans Solaris Resource Manager.

Chapitre 6 Décrit l'ordonnanceur, qui permet de contrôler l'attribution des ressources de l'UC.

Chapitre 7 Explique comment contrôler la quantité de mémoire virtuelle détenue par les utilisateurs et les processus individuels.

Chapitre 8 Décrit le mécanisme de collecte de données d'usage cumulatif des ressources de système et d'utilisateur.

Chapitre 9 Aide à diagnostiquer les problèmes de fonctionnement de Solaris Resource Manager.

Chapitre 10 Décrit les messages d'erreur possibles et leur signification.

Annexe A Donne des exemples de script.

*Glossaire* Présente une liste de mots et de phrases utilisés dans ce manuel, ainsi que leur définition.

---

## Publications connexes

Les documents ci-après fournissent des renseignements sur l'installation, la configuration et l'utilisation de Solaris Resource Manager ainsi que sur la version du produit :

- Le document *Solaris Resource Manager 1.0 Release Notes for Solaris 2.6 (SPARC Platform Edition)* est inclus dans la boîte du produit. Il présente une brève introduction au logiciel Solaris Resource Manager, identifie les fichiers correctifs requis et donne des informations sur les défauts et les problèmes connus.
- Le *Guide d'Installation de Solaris Resource Manager 1.0 pour Solaris 2.6 (Édition plateforme SPARC)* décrit l'installation de Solaris Resource Manager sur votre système d'exploitation. Il est inclus dans la boîte du produit.
- Le document *Solaris Resource Manager 1.0 Reference Manual for Solaris 2.6 (SPARC Platform Edition)* est la version AnswerBook des pages du manuel de Solaris Resource Manager. Il complète les pages du manuel de base Solaris 2.6 installées dans votre système. Le guide d'administration fait référence à ces pages. Des versions en ligne des pages du manuel, accessibles au moyen de la commande `man`, sont également fournies avec l'ensemble Solaris Resource Manager SUNWsrn.

---

## Pour commander des documents Sun

Le programme SunDocs<sup>SM</sup> comprend plus de 250 manuels de Sun Microsystems, Inc. Si vous résidez aux États-Unis, au Canada, en Europe ou au Japon, vous pouvez vous procurer des ensembles de documentation ou des manuels individuels par l'intermédiaire de ce programme.

Pour connaître la liste des documents disponibles et savoir comment les commander, consultez la section catalogue du site Web SunExpress<sup>TM</sup> à l'adresse <http://www.sun.com/sunexpress>.

---

## Accès en ligne à la documentation de Sun

Le site [Web docs.sun.com](http://docs.sun.com)<sup>TM</sup> vous permet d'accéder en ligne à la documentation technique de Sun. Vous pouvez consulter les archives [docs.sun.com](http://docs.sun.com) et rechercher un titre ou un sujet particulier. L'adresse du site est <http://docs.sun.com>.

La collection de documents Solaris Resource Manager 1.0, dont le présent manuel fait partie, est disponible à l'adresse [docs.sun.com](http://docs.sun.com); vous pouvez également accéder à ce site afin de savoir si une version mise à jour de ce guide est disponible.

TABLEAU P-1 Règles typographiques

Police ou symbole	Signification	Exemple
AaBbCc123	Noms de commandes, de fichiers et de répertoires dans le texte ; affichage à l'écran	Modifiez le fichier <code>.login</code> . Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers.  <code>machine_name% you have mail.</code>
<b>AaBbCc123</b>	Ce que vous devez entrer, par rapport à ce qui est affiché à l'écran	<code>machine_name% <b>su</b></code> <code>Password :</code>

TABLEAU P-1 Règles typographiques (suite)

Police ou symbole	Signification	Exemple
<i>AaBbCc123</i>	Paramètre substituable de ligne de commande : remplacez-le par un nom ou une valeur	Pour supprimer un fichier, tapez <code>rm nomfichier</code> .
<i>AaBbCc123</i>	Titres de publications, nouveaux mots ou termes mis en évidence.	Consultez le chapitre 6 du <i>Guide de l'utilisateur</i> . Ces options sont de type <i>classe</i> . Vous <i>devez</i> d'abord ouvrir une session.

---

## Invites du shell dans des exemples de commandes

Le tableau suivant indique l'invite système par défaut et l'invite de superutilisateur pour les shells C, Bourne et Korn.

TABLEAU P-2 Invites du shell

Shell	Invite
Invite du shell C	<code>machine_name%</code>
Invite de superutilisateur du shell C	<code>machine_name#</code>
Invite des shells Bourne et Korn	<code>\$</code>
Invite de superutilisateur des shells Bourne et Korn	<code>#</code>

## Aperçu

---

Le logiciel Solaris Resource Manager assure la disponibilité des ressources pour les utilisateurs, les groupes et les applications. Il permet d'attribuer et de contrôler les principales ressources système telles que l'UC, la mémoire virtuelle et le nombre de processus. Il met également en oeuvre les politiques administratives régissant les ressources auxquelles les différents utilisateurs ont accès, et en particulier, le niveau de consommation de ces ressources permis à chaque utilisateur. Le logiciel Solaris Resource Manager est un outil clé pour le regroupement de serveurs et l'usage accru des ressources.

Le logiciel Solaris Resource Manager est basé sur la technologie ShareII™ développée par Softway Pty. Limited (Australie).

---

## Introduction à Solaris Resource Manager

### Atteinte des objectifs de l'organisation

Les entreprises exigent souvent que les organisations de technologie de l'information contrôlent les coûts et garantissent les niveaux de service des applications de l'entreprise. La gestion des ressources permet d'accroître la disponibilité des procédures, ce qui diminue le coût de propriété, et de contrôler avec plus de précision les utilisateurs et l'usage du système.

En utilisant le logiciel Solaris Resource Manager pour catégoriser et définir les priorités d'usage, les administrateurs peuvent employer efficacement la capacité de réserve durant les périodes hors-pointe, éliminant souvent le besoin d'augmenter la puissance de traitement.

En répartissant les charges de travail dans le système, Solaris Resource Manager permet à l'administrateur d'exécuter et de gérer différentes applications dans un

même système, plutôt que d'allouer un système entier—et toute sa capacité de pointe—à chaque application. L'approche la plus couramment employée pour assurer un service et un temps de réponse prévisibles consiste à utiliser un système pour chaque fonction. Cette méthode fonctionne, mais la prolifération de systèmes dans le centre informatique est coûteuse et difficile à gérer.

Les gestionnaires de centre informatique veulent pouvoir regrouper de multiples applications sur un seul serveur UNIX afin d'utiliser au maximum toutes les ressources. En même temps, tous les utilisateurs doivent avoir accès à des ressources proportionnelles à leur niveau de service et à l'importance relative de leur travail.

## Quand utiliser Solaris Resource Manager

Solaris Resource Manager peut fournir un contrôle efficace des ressources dans diverses situations, y compris le regroupement de serveurs, l'accueil de sites Web pour les fournisseurs de services Internet, l'administration de sites à population élevée ou variée, et l'établissement de politiques visant à assurer un temps de réponse approprié pour les applications cruciales.

Solaris Resource Manager est idéal pour les environnements regroupant de multiples applications sur un même serveur. Le coût et la complexité de la gestion de nombreux ordinateurs incitent les gestionnaires de système à regrouper les applications sur des systèmes plus puissants et évolutifs. Avec Solaris Resource Manager, il est facile de réaliser de telles économies d'échelle.

Par exemple, un seul serveur Sun™ pourrait fournir les services d'applications, de fichiers et d'impression pour des clients hétérogènes, des services de messagerie/courrier, des services Web et des applications de base de données cruciales. Comme les serveurs Sun Enterprise™ peuvent comporter de 1 à 64 processeurs, un serveur peut être partagé entre plusieurs services ou utilisé par toute l'entreprise. Dans d'autres systèmes regroupés, les environnements de développement, de prototypage et de production sont combinés sur un seul serveur puissant, par exemple un modèle Sun Enterprise 10000 ou Sun Enterprise 6500, plutôt que d'être répartis sur trois serveurs distincts. On peut également combiner des serveurs de base de données et d'applications sur une seule machine, ou encore des banques de données multiples. Quel que soit le type d'application ou la configuration, Solaris Resource Manager vous aide à répartir les ressources système entre tous les utilisateurs, toutes les applications et tous les groupes, en fonction des politiques de votre entreprise. Les applications cruciales sont protégées, car elles reçoivent leur part appropriée des ressources système.

De même, avec Solaris Resource Manager, un fournisseur de services Internet peut loger un grand nombre (voire des milliers) de serveurs Web sur une seule machine. Solaris Resource Manager permet aux administrateurs de contrôler la consommation des ressources associée à chaque site Web, en protégeant chacun d'eux contre les excès potentiels des autres. Solaris Resource Manager empêche également les scripts CGI erronés d'épuiser les ressources de l'UC, et les applications des utilisateurs de

monopoliser toute la mémoire virtuelle disponible. Autrefois, les fournisseurs de services Internet devaient assigner des ordinateurs spécialisés à chaque client, ce qui augmentait considérablement les coûts et la complexité.

Solaris Resource Manager peut vous aider à gérer les ressources dans les systèmes à utilisateurs nombreux. Les établissements d'enseignement sont des exemples représentatifs de sites desservant une base d'utilisateurs importante et diversifiée. (D'ailleurs, Solaris Resource Manager a été initialement conçu en tant qu'ordonnanceur de ressources d'UC dans les universités de Sydney et de Nouvelle-Galles du Sud.) Si les charges de travail sont diversifiées, Solaris Resource Manager peut être configuré de manière à favoriser certains utilisateurs. 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. Si les négociants doivent bénéficier d'une puissance de traitement proportionnellement plus élevée, Solaris Resource Manager fait en sorte qu'ils l'obtiennent.

## Caractéristiques principales

Solaris Resource Manager permet de gérer la consommation de diverses ressources système importantes telles que le temps de processeur, la mémoire virtuelle, le nombre de processus, le nombre de connexions et le temps de connexion. Le modèle administratif de Solaris Resource Manager augmente la flexibilité en permettant la délégation de privilèges administratifs dans le cadre d'une hiérarchie, ce qui libère le personnel du centre informatique des transactions administratives intra-groupe. De plus, Solaris Resource Manager comporte des mécanismes de collecte de données sur l'usage des ressources pouvant servir à la planification de la capacité ou à des fins de contre-passation.

L'une des tâches fondamentales du système d'exploitation consiste à décider à quels processus attribuer l'accès aux ressources système. L'ordonnanceur par défaut de Solaris tente d'accorder à chaque processus environ le même temps d'accès aux ressources système. Une limitation de l'accès est appliquée aux processus sans ressources de mémoire physique dont l'exécution n'est pas autorisée, ainsi qu'aux processus ayant des requêtes d'E-S en suspens qui sont bloqués.

Cette méthode est à la base de la plupart des systèmes d'exploitation modernes ; elle donne de bons résultats à partir du moment où la politique "d'accès égal" convient à l'entreprise. Cependant, des mécanismes plus perfectionnés sont requis pour mettre en oeuvre des politiques différentes. Par exemple, un service de fabrication peut posséder un système puissant, mais généralement peu utilisé en raison de la fluctuation de la demande saisonnière, tandis que le service de génie est presque toujours à cours de ressources informatiques. Bien que la sous-utilisation des capacités d'une machine constitue un gaspillage, il a toujours été problématique de partager le système de fabrication avec le service de génie. Avec une politique d'ordonnement élémentaire, il est impossible de signifier au système d'exploitation que les utilisateurs du service de fabrication ont priorité sur ceux du

service de génie dans le même système. Si le service de fabrication exécute un travail consommant 75 % des ressources système, ce travail ne pourra progresser normalement que si les autres travaux demandent 25 ou moins de 25 % des ressources. Toutefois, si un travail de génie demande 50 % des ressources système, le travail de fabrication crucial ne disposera pas des ressources dont il a besoin, car le système tentera de satisfaire les besoins des deux travaux, à parts égales.

Supposons maintenant que les administrateurs déterminent que les besoins de traitement normaux de la fabrication peuvent être satisfaits avec 80 % de la capacité du système. Grâce à Solaris Resource Manager, l'administrateur du système peut spécifier que les utilisateurs du service de fabrication peuvent recevoir jusqu'à 85 % de la puissance de traitement du système s'ils le demandent, et l'ordonnanceur répartira le reste parmi les autres utilisateurs éventuels. Une autre possibilité de configuration consisterait à spécifier que les utilisateurs du service de fabrication peuvent recevoir jusqu'à 100 % des ressources système au besoin, ce qui empêcherait les autres services d'utiliser le système lorsque la fabrication en a besoin.

Solaris Resource Manager comporte une nouvelle classe d'ordonnement de l'UC qui remplace l'ordonnanceur de partage de temps standard. Cette classe, appelée classe d'ordonnement SHR, permet d'effectuer un partage dit "équitable". Ce terme est toutefois imprécis, car c'est l'administrateur du système qui décide ce qui est "équitable" et ce qui ne l'est pas. Ainsi, dans l'exemple ci-dessus il est équitable que le service de fabrication reçoive 100 % des ressources système. L'ordonnanceur SHR est responsable de l'attribution des ressources en fonction du plan défini dans le profil administratif.

Solaris Resource Manager met à jour une base de données de la consommation des ressources et des limites associées.

L'ordonnanceur SHR tient compte des spécifications administratives relativement aux ressources garanties. Il peut gérer les ressources renouvelables (le temps machine, par exemple) ou fixes (telles que le nombre de connexions).

D'autres modules de Solaris Resource Manager mettent en oeuvre des restrictions de la consommation de diverses ressources. Par exemple, le temps de connexion et le nombre de sessions utilisateur sont gérés par un module PAM (module d'authentification enfichable). Le module PAM consulte la base de données de Solaris Resource Manager chaque fois qu'un utilisateur tente d'ouvrir une session. Lorsque le système authentifie l'utilisateur (généralement par mot de passe), le temps de connexion de l'utilisateur et le nombre de connexions actuel sont comparés aux limites. Si les limites sont dépassées, la session est refusée.

## Lien avec les autres fonctions de contrôle des ressources de Solaris

L'environnement d'exploitation Solaris comporte plusieurs autres fonctions permettant de contrôler certains types de ressources. Certaines fonctions telles que



l'ordonnancement en temps réel, `nice(1)`, les quotas et les groupes de processeurs, font partie du système d'exploitation Solaris de base.

Bandwidth Allocator est un progiciel offert en option, les domaines de système dynamiques sont des fonctions de la plateforme du système Sun Enterprise 10000, et la reconfiguration dynamique est une fonction de la plateforme du système Sun Enterprise.

Toutes ces composantes permettent une certaine gestion des ressources, mais elles diffèrent des fonctions de Solaris Resource Manager.

Le système d'exploitation Solaris standard utilise la classe d'ordonnancement de partage de temps (TS) pour la plupart des tâches ordinaires. Cependant, il offre également l'ordonnancement en temps réel (RT) aux utilisateurs détenant des privilèges suffisants. La classe d'ordonnancement RT met en oeuvre une politique d'ordonnancement très différente (et intentionnellement très pondérée) afin d'assurer que des charges de travail particulières aient immédiatement accès au processeur. Solaris Resource Manager peut coexister sur le même système que la classe d'ordonnancement RT, mais n'a alors aucun contrôle sur les processus exécutés dans la classe RT. L'ordonnanceur à partage équitable de Solaris Resource Manager peut gérer les ressources de l'UC uniquement pour les processus exécutés dans la classe RT. Par exemple, dans le cas d'un système à 4 processeurs, un processus à une unité d'exécution peut utiliser toutes les ressources d'un processeur; c'est exactement ce qui se produit si le processus demandeur est lié à l'UC. Si ce système exécute aussi Solaris Resource Manager, les processus utilisateur ordinaires vont se disputer les 3 UC non accaparés par le processus en temps réel. (Il faut noter que les processus RT n'utilisent pas nécessairement l'UC continuellement. Lorsque l'UC se libère, Solaris Resource Manager commande les 4 processeurs.)

La commande `nice(1)` permet aux utilisateurs de définir leur priorité d'exécution. Sans privilèges de superutilisateur, cette commande ne permet à l'utilisateur que de diminuer sa priorité. Dans certains cas, cette commande peut s'avérer utile (par exemple lorsqu'un utilisateur lance un travail par lots à basse priorité dans sa session de connexion interactive), mais elle exige la coopération de l'utilisateur. Solaris Resource Manager met en oeuvre les politiques administratives même sans la coopération de l'utilisateur.

Les systèmes de fichiers Solaris sont dotés de mécanismes de quota permettant à l'administrateur de limiter l'usage du disque par des utilisateurs individuels. Cette fonctionnalité est indépendante de Solaris Resource Manager.

Les groupes de processeurs ont été instaurés dans Solaris 2.6. Cette caractéristique permet à l'administrateur de diviser les systèmes multiprocesseurs en groupes logiques, et aux utilisateurs de lancer des processus dans ces groupes. L'avantage de cette méthode est que les charges de travail exécutées dans un groupe de processeurs sont protégées contre les actions de l'UC effectuées dans tout autre groupe de processeurs. D'une certaine manière, cette méthode est comparable à celle utilisée par Solaris Resource Manager, mais les deux fonctions ont des bases complètement différentes. Les groupes de processeurs ne contrôlent que l'activité de l'UC. Le contrôle s'effectue à un niveau matériel bas, car les processeurs ne peuvent

appartenir qu'à un groupe de processeurs à la fois. Particulièrement dans le cas des petits systèmes, la granularité peut être très élevée. Dans un système à 4 processeurs, la ressource minimale pouvant être attribuée équivaut à 25 % du système.

Solaris Resource Manager exerce toutefois un contrôle beaucoup plus subtil ; en effet, chaque utilisateur dispose d'une part du système. Les parts peuvent être distribuées arbitrairement avec une fine granularité, et l'ordonnanceur attribue les ressources en conséquence. Par exemple, si 50 parts sont accordées et qu'un utilisateur en détient 40, cet utilisateur disposera de  $40 / 50 = 80\%$  de la ressource. De même, si un total de 67 parts est accordé, un utilisateur en possédant 57 disposera de 85 % de la ressource. En outre, Solaris Resource Manager peut contrôler d'autres ressources que l'UC.

Sun Enterprise 10000 est doté d'une caractéristique appelée domaines de systèmes dynamiques, qui permet à l'administrateur de diviser logiquement une baie système en un ou plusieurs systèmes indépendants exécutant chacun sa propre copie de Solaris. Par exemple, un système comportant 32 UC réparties sur 8 cartes système pourrait être exploité comme un système à 16 UC et deux autres systèmes à 8 UC chacun. Avec une telle configuration, trois copies de Solaris seraient exécutées. Les domaines de systèmes dynamiques permettent également l'entrée et la sortie contrôlées de ressources dans chaque image de Solaris, créant ainsi un système à granularité assez élevée de gestion des ressources physiques. (L'unité minimale d'attribution inter-domaine est une carte système entière.) Solaris Resource Manager est semblable aux domaines de systèmes dynamiques en ce sens qu'il fournit à l'administrateur des mécanismes d'attribution des ressources, mais de manière tout à fait différente. Solaris Resource Manager fonctionne avec une seule instance de Solaris, et permet de gérer les ressources du système avec une haute granularité. Les domaines de systèmes dynamiques divisent une même unité matérielle en instances multiples de Solaris ; ils fournissent des outils de gestion du transfert des ressources entre instances de Solaris exécutées dans le même bâti Sun Enterprise 10000. Solaris Resource Manager est perpendiculaire aux domaines de systèmes dynamiques et peut être utilisé conjointement avec eux. Solaris Resource Manager peut tourner sous chaque instance de Solaris dans un système Sun Enterprise 10000.

La caractéristique de reconfiguration dynamique des serveurs Sun Enterprise permet aux utilisateurs d'ajouter et de supprimer dynamiquement des cartes système contenant des ressources matérielles telles que des processeurs, de la mémoire et des périphériques d'E-S. La reconfiguration dynamique de la mémoire n'a aucun effet sur la vérification de limite de mémoire effectuée par Solaris Resource Manager.

Le progiciel optionnel Bandwidth Allocator fonctionne conjointement avec le noyau Solaris afin de mettre en oeuvre les limites de consommation de largeur de bande réseau. Bandwidth Allocator est un logiciel effectuant la gestion d'une classe de ressources différente. Solaris Resource Manager et Bandwidth Allocator ont des domaines de gestion différents : Solaris Resource Manager fonctionne sur une base "par utilisateur" ou "par application", tandis que Bandwidth Allocator gère sur une base "par port", "par service" ou "par protocole".

## Différences entre Solaris Resource Manager et les produits similaires

Solaris Resource Manager est relié à plusieurs autres composants logiciels pouvant être présents dans le système, mais sans toutefois les remplacer. Comme nous l'avons vu, Bandwidth Allocator gère un type de ressources différent. Bien que Solaris Resource Manager possède certaines fonctions de gestion et de surveillance de système, il n'est pas un logiciel de surveillance de système tel que Sun Enterprise SyMON™ 2.0. De même, Solaris Resource Manager n'est pas un outil de planification de capacité. Il aide l'administrateur à gérer la capacité, et ses fonctions de comptabilité génèrent des registres d'usage pouvant être utilisés par le personnel de gestion pour effectuer des analyses de tendances, mais il n'effectue pas la planification de capacité en tant que telle. Solaris Resource Manager n'est pas non plus un ordonnanceur de travaux ; il contrôle l'exécution d'un processus sur son système hôte, mais non l'endroit et le moment où celui-ci est exécuté. Enfin, étant donné que Solaris Resource Manager fonctionne sur un seul système, il ne s'agit pas d'un mécanisme d'équilibrage des charges parmi les membres d'un groupe. Solaris Resource Manager peut toutefois être utilisé efficacement pour gérer les charges de travail individuelles de chaque membre d'un groupe. Par exemple, vous pouvez donner priorité au travail d'un membre défaillant d'un groupe à haute disponibilité par rapport à une charge de travail d'arrière plan d'un membre de relève.



## Utilisations normales

---

Ce chapitre décrit les principes de fonctionnement de Solaris Resource Manager et ses concepts clés. Il comporte des exemples éclaircissant les descriptions et illustrant les utilisations courantes de Solaris Resource Manager.

---

### Aperçu des noeuds limites

Solaris Resource Manager est basé sur une nouvelle composante du noyau Solaris appelée "noeud limite". Les noeuds limites correspondent aux UID (ID utilisateur) de UNIX et peuvent représenter des utilisateurs individuels, des groupes d'utilisateurs et des exigences spéciales. Les noeuds limites sont indexés par UID et servent à enregistrer les politiques d'attribution des ressources et les données d'usage cumulatives par processus au niveau de l'utilisateur, du groupe d'utilisateurs ou de l'application.

---

### Structure hiérarchique

Le modèle de gestion de Solaris Resource Manager organise les noeuds limites en une structure hiérarchique appelée arbre d'ordonnement. L'arbre d'ordonnement est organisé par UID : chaque noeud limite fait référence à l'UID de son père dans l'arbre. Chaque sous-arbre de l'arbre d'ordonnement est appelé groupe d'ordonnement, et l'utilisateur situé à la racine d'un groupe d'ordonnement est le chef de groupe. L'utilisateur racine est le chef de groupe de l'arbre d'ordonnement entier. Un chef de groupe peut être autorisé à gérer les politiques de ressources du groupe. Les noeuds limites sont initialement créés par

analyse du fichier de l'UID. Une commande d'administration de noeuds limites (`limadm(1MSRM)`) permet de créer des noeuds limites supplémentaires après l'installation de Solaris Resource Manager et d'assigner des noeuds limites aux pères. Les données de l'arbre d'ordonnancement sont enregistrées dans une base de données bidimensionnelle pouvant, au besoin, être modifiée.

Bien que les UID utilisées par les noeuds limites ne doivent pas obligatoirement correspondre à un compte système, avec une entrée dans la table des mots de passe du système, il est fortement recommandé de créer un compte système pour l'UID de chaque noeud limite. Dans le cas des noeuds limites qui ne sont pas des "feuilles" (ayant des noeuds limites subordonnés dans la hiérarchie), il se peut que le compte associé à ce noeud soit purement administratif et que personne n'y ouvre une session. Toutefois, il est également possible qu'il s'agisse du noeud limite d'un utilisateur réel qui y ouvre des sessions et exécute les processus reliés à ce noeud qui n'est pas une feuille.

Il est à remarquer que les groupes d'ordonnancement et les chefs de groupe de Solaris Resource Manager n'ont rien à voir avec les groupes de systèmes définis dans la base de données `/etc/group`. Chaque noeud de l'arbre d'ordonnancement, y compris les chefs de groupe, correspond à un utilisateur de système réel ayant une UID unique.

---

## Limites hiérarchiques

Si une limite hiérarchique est assignée à un chef de groupe dans un arbre de noeuds limites (groupe d'ordonnancement), elle s'applique aussi au niveau d'usage de cet utilisateur et au niveau d'usage total de tous les membres du groupe d'ordonnancement. Cela permet d'appliquer des limites à des groupes entiers ainsi qu'à des membres individuels. Les ressources sont attribuées au chef de groupe, lequel peut les attribuer aux utilisateurs ou aux groupes d'utilisateurs appartenant au même groupe.

---

## Processus

Chaque processus correspond à un noeud limite. Le processus `init` est toujours attaché au noeud limite racine. Lorsque des processus sont créés au moyen de l'appel système `fork(2)`, ils sont attachés au même noeud limite que leur père. Les processus peuvent être reliés de nouveau à tout noeud limite à l'aide d'un appel système de Solaris Resource Manager, à condition que les privilèges soient suffisants. Les privilèges sont définis par racine ou par utilisateur avec les autorisations administratives pertinentes activées.

---

# Contrôle des ressources

Solaris Resource Manager effectue le contrôle des ressources système suivantes : usage de l'UC (taux d'utilisation du processeur), mémoire virtuelle, nombre de processus, nombre de connexions et temps de connexion du terminal.

Solaris Resource Manager fait le suivi de l'usage de chaque ressource par utilisateur. Pour toutes les ressources à l'exception de l'usage de l'UC, vous pouvez assigner aux utilisateurs des limites strictes d'usage des ressources. Avec une limite stricte, les tentatives de consommation de ressources échouent si l'utilisateur atteint cette limite. Les limites strictes sont appliquées directement soit par le noyau, soit par le logiciel responsable de la gestion de la ressource en question. Si la valeur de la limite est zéro, il n'y a pas de limite. Tous les attributs relatifs aux limites du noeud limite racine doivent être laissés à zéro.

Solaris Resource Manager élimine graduellement l'usage antérieur, de sorte que seul l'usage la plus récente soit significative. L'administrateur du système définit un paramètre de demi-vie qui contrôle le taux de décroissance. Une longue demi-vie favorise un usage uniforme, convenant aux longs traitements par lots, tandis qu'une demi-vie courte favorise les utilisateurs interactifs.

Généralement, toutes les ressources système peuvent être divisées en deux classes : les ressources fixes (non renouvelables) et les ressources renouvelables. Solaris Resource Manager gère différemment ces deux types de ressources.

## **Ressources fixes**

Les ressources fixes, ou non renouvelables, sont disponibles en quantité limitée. Par exemple : la mémoire virtuelle, le nombre de processus, le nombre de connexions et le temps de connexion. Les ressources fixes peuvent être consommées (attribuées) et libérées (désattribuées), mais aucune autre entité ne peut utiliser la ressource avant que son propriétaire ne la libère. Solaris Resource Manager emploie un modèle d'usage et de limite pour contrôler la consommation des ressources fixes. L'usage est défini comme la ressource en cours d'utilisation, et la limite est le niveau maximum d'usage permis par Solaris Resource Manager.

## **Ressources renouvelables**

Les ressources renouvelables sont celles qui sont continuellement disponibles, par exemple le temps de processeur. Les ressources renouvelables peuvent uniquement être consommées, et une fois consommées, ne peuvent plus être récupérées. À tout instant, la

disponibilité d'une ressource renouvelable est limitée. Si elle n'est pas consommée immédiatement, elle ne sera plus disponible à l'avenir. (On peut faire une analogie avec la lumière du soleil. Une quantité fixe de lumière nous parvient du soleil à un instant donné, mais nous continuerons à en recevoir durant des millions d'années.) Pour cette raison, les ressources renouvelables peuvent être réassignées à d'autres utilisateurs sans réattribution explicite afin d'éviter le gaspillage. Solaris Resource Manager emploie un modèle d'usage, de limite et de décroissance pour contrôler le taux de consommation d'une ressource renouvelable par un utilisateur. L'usage est défini comme la quantité totale de ressources utilisées ; une limite fixe le taux maximum d'usage par rapport aux autres utilisateurs. La décroissance désigne la période de réduction de l'usage historique. Le quantum de ressource suivant, par exemple un top d'horloge, sera attribué au noeud limite actif ayant la valeur d'usage décréue la plus faible par rapport à sa part attribuée. La valeur d'usage décréue mesure l'utilisation totale dans le temps, moins une partie de l'usage historique déterminée selon un modèle de décroissance à demi-vie.

Les ressources de l'UC sont contrôlées par l'ordonnanceur SHR de Solaris Resource Manager. Le temps de processeur est attribué dynamiquement aux utilisateurs proportionnellement au nombre de parts qu'ils détiennent (par analogie avec les parts d'une compagnie), et en proportion inverse de leur usage récent. L'ordonnanceur joue un autre rôle important : en plus de gérer l'ordonnement d'unités d'exécution individuelles (en termes techniques, dans Solaris, l'entité ordonnée est appelée processus léger), il répartit les ressources de l'UC entre les utilisateurs.

Chaque utilisateur possède également un ensemble d'indicateurs, qui sont en fait des variables booléennes permettant d'activer ou de désactiver certains privilèges système, par exemple la connexion. Les indicateurs peuvent être réglés individuellement par utilisateur, ou être hérités d'un noeud limite père.

Les taux d'usage, les limites et les indicateurs peuvent être consultés par tout utilisateur, mais ne peuvent être modifiés que par les utilisateurs détenant des privilèges administratifs.



---

## Gestion des ressources de l'UC

L'attribution du service d'UC renouvelable est contrôlé au moyen d'un ordonnanceur à partage équitable. Un certain nombre de parts d'UC, comparables aux parts d'une compagnie, est attribué à chaque noeud limite. Des ressources d'UC sont attribuées aux processus associés à chaque noeud limite proportionnellement au nombre total de parts actives disponibles (une part est active si le noeud limite est relié à des processus en cours). Seuls les noeuds limites actifs sont pris en compte pour l'attribution d'une ressource, car ils comportent des processus en cours d'exécution et exigent du temps de processeur. À mesure qu'un processus consomme des tops d'horloge d'UC, l'attribut d'usage de l'UC de ce noeud limite augmente.

L'ordonnanceur règle périodiquement les priorités de tous les processus afin de forcer les taux relatifs d'usage de l'UC à converger vers les taux relatifs de parts d'UC pour tous les noeuds limites actifs à leurs niveaux respectifs. Ainsi, les utilisateurs recevront à long terme au moins leur part équitable de ressources de l'UC, quelle que soit l'activité des autres utilisateurs. L'ordonnanceur est hiérarchique, car il fait également en sorte que les groupes reçoivent leur part équitable indépendamment de l'activité des membres. Solaris Resource Manager est un ordonnanceur à long terme ; il fait en sorte que tous les utilisateurs et toutes les applications reçoivent leur juste part au cours de la "période d'ordonnancement". Cela signifie que lorsqu'un utilisateur demande d'accéder à l'UC, cet utilisateur reçoit proportionnellement plus de ressources que les gros utilisateurs, jusqu'à ce que leurs usages comparatifs correspondent à leur part "équitable". Plus vous dépassez votre part équitable actuellement, moins vous recevrez ultérieurement. En outre, Solaris Resource Manager emploie une période de décroissance, réglée par l'administrateur du système, qui lui fait "oublier" l'usage antérieur. Le modèle de décroissance est à demi-vie : au cours d'une demi-vie, 50 % de la ressource est éliminée. Grâce à cette méthode, les utilisateurs stables ne sont pas pénalisés par les utilisateurs à court terme consommant beaucoup de ressources. La période de décroissance à demi-vie fixe la réponse ou le "terme" de l'ordonnanceur ; la valeur par défaut est 120 secondes. Les valeurs plus faibles augmentent la rapidité de réaction dans le système, au détriment de la précision de calcul et de la mise à jour des attributions dans l'ensemble du système. Quels que soient les réglages administratifs, l'ordonnanceur tente d'empêcher l'abandon (la privation de ressources) et d'assurer un comportement raisonnable, même dans les situations extrêmes.

Le principal avantage de l'ordonnanceur de Solaris Resource Manager par rapport à l'ordonnanceur standard de Solaris est qu'il programme des utilisateurs ou des applications plutôt que des processus individuels. Chaque processus associé à un noeud limite est soumis à un ensemble de limites. Dans le cas simple d'un utilisateur exécutant un seul processus actif, cela revient à soumettre chaque processus aux limites spécifiées dans le noeud limite correspondant. Lorsque plusieurs processus sont reliés à un noeud limite, par exemple lorsque les membres d'un groupe exécutent chacun de multiples processus, tous les processus sont collectivement soumis aux limites imposées. Cela signifie que les utilisateurs ou les applications ne

peuvent pas consommer les ressources de l'UC à un taux dépassant leur part attribuée, peu importe le nombre de processus concurrents qu'ils exécutent. Le mode d'attribution du nombre de parts est simple et compréhensible, et l'effet de la modification du nombre de parts d'un utilisateur est prévisible.

Solaris Resource Manager ne gaspille jamais la disponibilité de l'UC. Même si la part attribuée à un utilisateur est très faible, cet utilisateur recevra toutes les ressources de l'UC s'il n'y a aucun utilisateur concurrent. Par conséquent, il est possible que les utilisateurs remarquent une baisse des performances. Si un utilisateur ayant une faible part exécute un processus interactif en l'absence de concurrence, il semblera être exécuté rapidement. Toutefois, dès qu'un autre utilisateur ayant une part supérieure demandera d'accéder à l'UC, il aura priorité sur le premier utilisateur, lequel remarquera un ralentissement de l'exécution de sa tâche. Néanmoins, Solaris Resource Manager fait le nécessaire pour que les utilisateurs légitimes ne restent pas sans ressources. Tous les processus programmés par Solaris Resource Manager (sauf ceux ayant une valeur `nice` maximale) ont périodiquement accès à l'UC. De plus, il empêche les nouveaux utilisateurs qui viennent de s'ouvrir une session d'obtenir une proportion "équitable" de l'UC du point de vue arithmétique, mais excessive au détriment des utilisateurs existants.

---

## Mémoire virtuelle (limites par utilisateur et par processus)

La mémoire virtuelle est gérée au moyen d'un modèle à ressources fixes. La limite de mémoire virtuelle s'applique à la mémoire totale employée par tous les processus reliés au noeud limite. Il existe également une limite de mémoire virtuelle par processus qui restreint l'espace total d'adresses virtuelles du processus, y compris toutes les bibliothèques de code, de données, de pile, de mappage de fichiers et les bibliothèques partagées. Les deux limites sont hiérarchiques. La limitation de la mémoire virtuelle est utile afin d'éviter le manque de mémoire virtuelle. Par exemple, Solaris Resource Manager arrête les applications qui consomment une quantité injustifiée de mémoire virtuelle au détriment de tous les utilisateurs. En effet, de tels processus se privent eux-mêmes de ressources, ou même les autres processus appartenant au même groupe.

---

## Nombre de processus

Le nombre de processus que les utilisateurs peuvent exécuter simultanément est contrôlé au moyen d'un modèle à ressources fixes avec limites hiérarchiques.

---

## Terminaux et temps de connexion

L'administrateur de système et le chef de groupe peuvent définir des privilèges de connexion de terminal et limiter le nombre de connexions et le temps de connexion ; ces limites sont appliquées hiérarchiquement par Solaris Resource Manager. Lorsqu'un utilisateur s'approche de sa limite de temps de connexion, des messages d'avertissement sont envoyés au terminal de l'utilisateur. Lorsque la limite est atteinte, l'utilisateur en est avisé, puis sa session fermée après un court délai de grâce.

---

## Administration des utilisateurs

L'administrateur du système peut définir des privilèges administratifs pour tout noeud limite, y compris l'attribution sélective de privilèges administratifs aux utilisateurs. Un utilisateur détenant des privilèges administratifs hiérarchiques est appelé sous-administrateur. Un sous-administrateur peut créer, supprimer et modifier les noeuds limites des utilisateurs dans le sous-arbre dont il est le chef.

Toutefois, le sous-administrateur ne peut normalement pas modifier ses propres limites ou indicateurs, et ne peut pas non plus contourner ses propres indicateurs et limites en modifiant les indicateurs ou les taux d'usage de son groupe.

L'administrateur central (ou superutilisateur) peut modifier les limites, les taux d'usage et les indicateurs de chaque utilisateur, y compris lui-même. Les utilisateurs ordinaires peuvent aussi obtenir ce privilège si l'indicateur correspondant est activé.

---

## Aperçu des données d'usage

Solaris Resource Manager conserve des informations (principalement sur l'usage courant et cumulatif des ressources) dont les administrateurs peuvent se servir pour comptabiliser les ressources système. Aucun programme de comptabilité n'est fourni avec Solaris Resource Manager, mais il comporte des utilitaires de base permettant le développement d'un système personnalisé de comptabilisation des ressources.

Pour de plus amples renseignements sur les procédures de comptabilité, voir Chapitre 8.

---

# Exemples

Les exemples de cette section illustrent les fonctions de Solaris Resource Manager utilisées pour contrôler et attribuer des ressources système, et pour afficher des informations.

## Exemple de regroupement de serveurs

Le premier exemple illustre les commandes suivantes :

<b>liminfo</b>	Affiche dans une fenêtre de terminal les attributs de mot de passe et les informations sur les limites pour un ou plusieurs utilisateurs.
<b>limadm</b>	Permet de modifier les attributs de limite ou de supprimer des entrées dans la base de données des limites pour une liste d'utilisateurs.
<b>srmuser</b>	Permet d'afficher et de régler les modes de fonctionnement et certains paramètres système de Solaris Resource Manager.
<b>srmstat</b>	Affiche des informations sur l'activité des noeuds limites.

Considérons le cas d'un regroupement de deux serveurs exécutant chacun une application de base de données, en une seule machine. Le simple fait d'exécuter les deux applications sur une seule machine résulte en un système utilisable ; mais sans Solaris Resource Manager, le système d'exploitation Solaris attribue les ressources aux applications sur une base égale, et ne protège pas chaque application contre les exigences concurrentielles de l'autre. Toutefois, Solaris Resource Manager comporte des mécanismes empêchant les applications de manquer de ressources. Solaris Resource Manager accomplit cela en liant chaque base de données à des noeuds limites correspondants, *bd1* et *bd2*. Pour ce faire, trois nouveaux paramètres substituables d'utilisateur administratif doivent être créés. Dans cet exemple, nous utiliserons les *bases de données bd1* et *bd2*. Ces paramètres sont ajoutés à la base de données des noeuds limites. Étant donné que les noeuds limites correspondent à des UID UNIX, celles-ci doivent aussi être ajoutées au fichier `passwd` (ou à la table des mots de passe, si le système utilise un service de nom tel que NIS ou NIS+). En supposant que les UID sont ajoutées au fichier `passwd` ou à la table des mots de passe, les utilisateurs des paramètres substituables *bd1* et *bd2* sont attribués au groupe de noeuds limites *bases de données* au moyen de la commande suivante :

```
% limadm set sgroup=0 databases
% limadm set sgroup=databases db1 db2
```

Cela suppose que le répertoire `/usr/srm/bin` se situe dans la route de l'utilisateur.

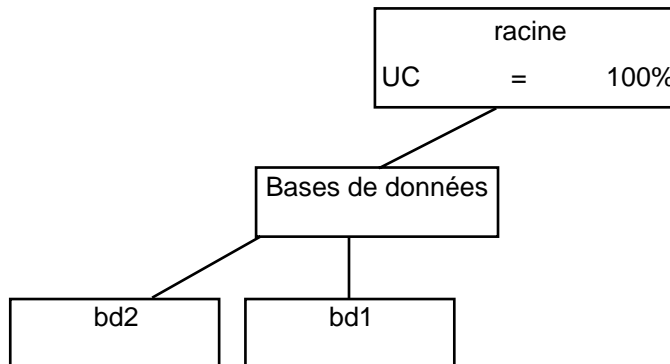


Figure 2-1 Regroupement des serveurs

Comme il n'existe pas d'autres groupes définis, le groupe *bases de données* a actuellement un accès complet à la machine. Deux noeuds limites associés aux bases de données sont en cours, et les processus exécutant les applications de base de données sont reliés aux noeuds limites appropriés au moyen de la commande `srmuser` dans le script de démarrage pour les instances de base de données, par exemple :

```
% srmuser bd1 /usr/bin/database1/init.db1
% srmuser bd2 /usr/bin/database2/init.db2
```

Lorsque l'une des base de données, *bd1* ou *bd2*, est lancée, utilisez la commande `srmuser` pour vous assurer que la base de données est reliée au noeud limite approprié et correctement chargée (`srmuser` n'influence pas la propriété de processus lors de cette opération). Pour exécuter la commande ci-dessus, l'utilisateur doit détenir les autorisations UNIX requises pour exécuter `init.db1` et l'autorisation administrative de lier des processus au noeud limite *bd1*. Lorsque les utilisateurs ouvrent une session et utilisent les bases de données, les activités effectuées par les bases de données sont cumulées dans les noeuds limites *bd1* et *bd2*.

En utilisant l'attribution par défaut d'une part par noeud limite, le taux d'usage du groupe *bases de données* à long terme fera en sorte que les bases de données *bd1* et *bd2* reçoivent une part égale des ressources de la machine. Plus particulièrement, il y a une part disponible—dans le groupe *bases de données*—, et *bases de données* la détient. Chacun des noeuds limites *bd1* et *bd2* reçoit également l'attribution par défaut d'une part. Dans le groupe *bases de données*, deux parts sont disponibles, donc *bd1* et *bd2* reçoivent une attribution égale des ressources de *bases de données* (dans ce simple

exemple, il n'y a pas d'attributions concurrentielles, et *bases de données* a ainsi accès à la totalité du système).

S'il se trouve que l'activité de la Base de données 1 exige 60 % de la capacité de l'UC de la machine et que la Base de données 2 exige 20 % de la capacité, l'administrateur peut spécifier que le système doit fournir au moins cette proportion (en supposant que l'application l'exige) en augmentant le nombre de parts de l'UC ( `cpu.shares`) attribué à *bd1* :

```
% limadm set cpu.shares=3 db1
```

Quatre parts sont maintenant disponibles dans le groupe *bases de données* ; *bd1* en détient trois, et *bd2* une. Ce changement est effectué dès l'exécution de la commande ci-dessus. Durant la période d'uniformisation qui suit, le noeud limite *bd1* (Base de données 1) reçoit plus que sa juste part de 60 % des ressources de la machine, étant donné que Solaris Resource Manager étale l'usage dans le temps. Toutefois, selon le paramètre global de décroissance, cette période est assez courte.

Pour surveiller cette activité en tout temps, utilisez les commandes `liminfo` et `srmsstat` dans des fenêtres distinctes :

```
% liminfo -c db1
# limit information shows all the data and
# settings for the lnode db1.
```

Voir "Serveur d'applications type" , page 27.

Par contre, `srmsstat` fournit un affichage régulièrement mis à jour :

```
% srmsstat -ac # srmsstat shows the server activity and the
# flag -ac sets a screen default update period
# of 4 seconds to display the results.
```

La machine exécute maintenant deux applications de base de données, dont l'une reçoit 75 % des ressources, et l'autre 25 %. N'oubliez pas que le superutilisateur (racine) est le chef de groupe au niveau le plus élevé. Les processus exécutés en tant que racine ont donc accès à la totalité du système, s'ils le demandent. De même, des noeuds limites supplémentaires devraient être créés pour l'exécution de sauvegardes, de démons et d'autres scripts afin d'empêcher les processus racine de monopoliser toute la machine, comme ils pourraient le faire dans le mode traditionnel.

# Ajout d'un utilisateur d'application de calcul par lots

Cet exemple illustre la commande suivante :

**srmkill** Détruit tous les processus actifs reliés à un noeud limite.

Le service des finances est propriétaire du système de base de données, mais un utilisateur (Joe) du service de génie doit exécuter un travail de calcul et voudrait utiliser la machine des finances durant les heures où le système est généralement inactif. Le service des finances considère le travail de Joe comme moins important que les bases de données, et accepte d'exécuter son travail uniquement s'il ne perturbe pas le rôle principal du système. Pour mettre cette politique en oeuvre, ajoutez un nouveau groupe ( *lot* ) au noeud limite *base de données*, et ajoutez Joe au nouveau groupe *lot* dans la hiérarchie de noeuds limites du serveur.

```
% limadm set cpu.shares=20 databases
% limadm set cpu.shares=1 batch
% limadm set cpu.shares=1 joe
% limadm set sgroup=lot joe
```

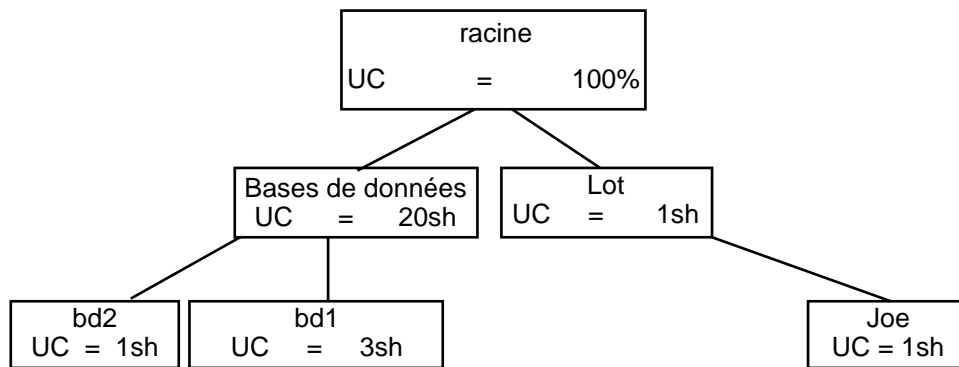


Figure 2-2 Ajout d'une application de calcul par lots

Cette série de commandes permet de modifier l'attribution des parts afin que le groupe *bases de données* possède 20 parts, et le groupe *lot* une seule. Ainsi, les membres du groupe *lot* (Joe seulement) utiliseront au plus 1/21 des ressources de la machine si le groupe *bases de données* est actif. Le groupe *bases de données* reçoit 20/21, ou 95,2 %, ce qui dépasse la proportion de 60 % + 20 % = 80 % antérieurement déterminée comme étant suffisante pour exécuter les applications de base de données. Si les *bases de données* ne demandent pas leur attribution entière, Joe recevra plus que sa part de 4,8 %. Si les *bases de données* sont inactives, l'attribution de Joe

pourrait atteindre 100 %. Lorsque le nombre de parts disponibles attribué aux *bases de données* est augmenté de 1 à 20, il n'est pas nécessaire de modifier l'attribution des parts de *bd1* et *bd2*. Dans le groupe *bases de données*, quatre parts sont toujours disponibles, attribuées à raison de 3/1. Les différents niveaux de l'arbre d'ordonnancement sont complètement indépendants ; seul le rapport du nombre de parts entre les groupes homologues importe.

Même avec ces assurances, le service des finances souhaite s'assurer que Joe ne pourra pas ouvrir de session durant les heures de bureau. Cela peut être accompli en insérant certains contrôles de connexion dans le groupe *lot*. Comme les contrôles tiennent compte de l'heure, l'objectif peut être réalisé en exécutant un script modifiant le nombre de connexions accordées au groupe *lot* au début et à la fin de la journée. Par exemple, des entrées *crontab* pourraient être employées :

```
0 6 * * * /usr/srm/bin/limadm set logins=0 batch
0 18 * * * /usr/srm/bin/limadm set logins=100 batch
```

À 6h00, la limite de connexions de *lot* passe à 0, et à 18h00, elle est augmentée de manière à permettre 100 connexions.

Une politique encore plus stricte peut être mise en oeuvre en ajoutant une autre ligne à l'entrée *crontab* :

```
01 6 * * * /usr/srm/bin/srmkill joe
```

Cette ligne utilise la commande *srmkill* pour détruire tout processus relié au noeud limite Joe à 6h01. Cela ne sera pas nécessaire si les seules ressources requises par le travail sont contrôlées par Solaris Resource Manager. Cette action pourrait toutefois être utile si le travail de Joe était susceptible d'accaparer des ressources qui nuiraient aux opérations normales, par exemple un travail pouvant bloquer une base de données ou dominant un canal d'E-S.

À présent, Joe peut ouvrir une session et exécuter son travail uniquement durant la nuit. Vu que Joe (et le groupe *lot*) détient beaucoup moins de parts que les autres applications, l'exécution de son application utilisera moins de 5 % des ressources de la machine. De même, la commande *nice(1)* peut être utilisée pour réduire la priorité des processus reliés à ce travail, afin que sa priorité d'exécution soit inférieure à celle des autres travaux ayant le même nombre de parts dans Solaris Resource Manager.

Maintenant, le service des finances a fait le nécessaire pour que ses applications de base de données aient un accès suffisant au système et ne se nuisent pas mutuellement. En outre, les besoins de traitement par lot nocturnes de Joe sont satisfaits, tout en s'assurant qu'ils ne perturberont pas les applications critiques.



## Mise en place d'un processus Web frontal

Supposons qu'il a été décidé de mettre en place un processus frontal sur la Base de données 1, mais que l'accès à cette application doit être limité à 10 utilisateurs simultanés. Pour ce faire, utilisez la fonction de limites de processus.

D'abord, créez un noeud limite appelé *sw1*. En lançant l'application de serveur Web sous le noeud limite *sw1*, vous pouvez contrôler le nombre de processus auquel il a accès, et donc le nombre de sessions `http` actives.

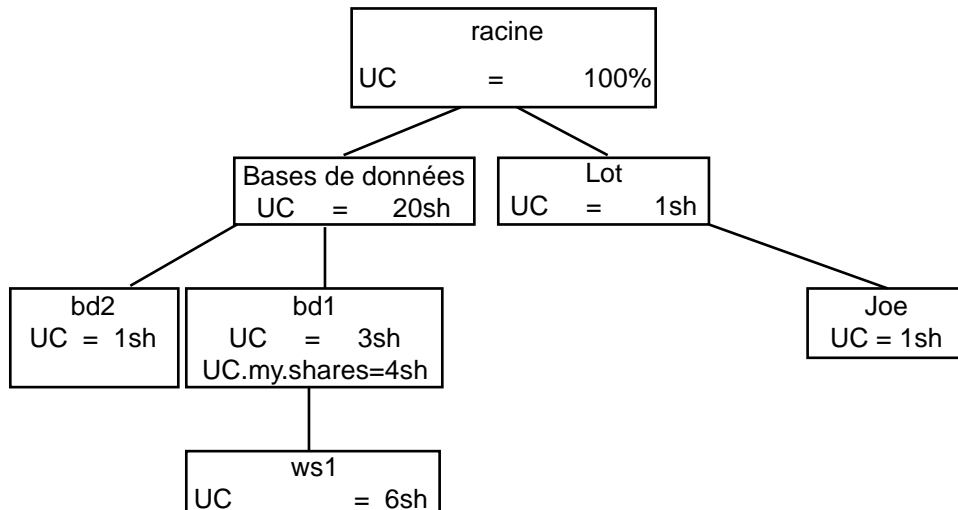


Figure 2-3 Ajout d'un processus Web frontal

Étant donné que le serveur Web fait partie de l'application de la Base de données 1, vous pourriez lui attribuer une part du noeud limite *bd1* et lui permettre de concurrencer la Base de données 1 quant à l'accès aux ressources. Attribuez 60 % des ressources informatiques au serveur Web, et 40 % à l'application de la Base de données 1 :

```
# limadm set cpu.shares=6 sw1
# limadm set sgroup=bd1 sw1
# limadm set cpu.myshares=4 bd1
# srmuser sw1 /etc/bin/Webserver1/init.webserver
```

La dernière ligne démarre le serveur Web et charge l'application dans le noeud limite *sw1*. Remarquez que pour la Base de données 1, la variable `cpu.myshares` est réglée à 4. Cela définit la proportion de parts pour lesquelles *bd1* va concurrencer son processus enfant, c'est-à-dire le serveur Web, à raison de 4/6.

---

**Remarque** - `cpu.shares` indique le rapport d'attribution des ressources au niveau homologue dans une hiérarchie, tandis que `cpu.myshares` indique le rapport d'attribution des ressources au niveau père/enfants lorsque le père exécute des applications. Solaris Resource Manager attribue les ressources en fonction de la proportion des parts disponibles de tous les noeuds limites actifs à leur niveau correspondant, ce dernier incluant la valeur `my.shares` du père du groupe et de tous ses enfants.

---

Pour spécifier le nombre de processus pouvant être exécutés par le serveur Web, imposez une limite de processus au noeud limite `sw1`. L'exemple emploie la valeur 20, car une interrogation de serveur Web engendre habituellement 2 processus, ce qui limite le nombre d'interrogations actives du serveur Web à 10 :

```
# limadm set process.limit=20 sw1
```

Une autre application a maintenant été ajoutée à l'arbre d'ordonnancement en tant que noeud feuille sous un noeud limite actif. Pour distribuer les ressources de l'UC entre le père actif et l'enfant, utilisez la variable `cpu.myshares` de manière à attribuer une partie de la ressource disponible au père, et une partie à l'enfant. Les limites de processus servent à restreindre le nombre de sessions actives dans un noeud limite.

## Ajout d'utilisateurs ayant des besoins de mémoire spéciaux

Cet exemple illustre les mécanismes de contrôle des ressources, le partage de l'UC, les limites de processus et les contrôles de connexion, ainsi que les outils d'affichage permettant l'impression de noeuds limites et la visualisation des noeuds limites actifs.

<b>srmadm</b>	Administrer Solaris Resource Manager.
<b>limreport</b>	Générer des informations sur des utilisateurs.
<b>limdaemon</b>	Indiquer au démon d'envoyer des messages lorsqu'une limite est atteinte.

Un autre utilisateur, Sally, veut également utiliser la machine durant la nuit afin d'exécuter son application. Vu que son application fait un usage intensif de l'UC, pour empêcher que l'application de Joe n'en souffre, imposons une limite à l'usage de la mémoire virtuelle par Sally, en termes d'usage total et par processus.

```
% limadm set memory.limit=50M sally
% limadm set memory.plimit=25M sally
```

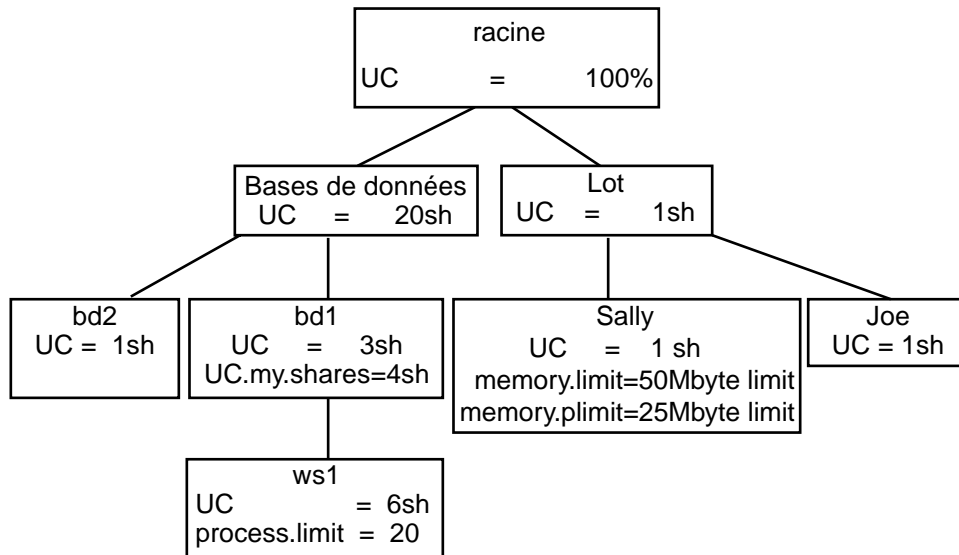


Figure 2-4 Ajout d'utilisateurs

Si l'application de Sally tente de dépasser sa limite totale de mémoire virtuelle ou de processus, la commande `limdaemon` lui signalera, ainsi qu'à l'administrateur de système, par l'intermédiaire de la console, que sa limite a été dépassée.

Utilisez la commande `limreport(1MSRM)` pour générer un rapport sur les utilisateurs actuels du système et sur leur usage cumulatif. On utilise généralement la commande `limreport` pour savoir qui utilise la machine à un moment donné et où ces personnes se situent dans la hiérarchie des utilisateurs.

```
% limreport 'flag.real' - uid sgroup lname cpu.shares cpu.usage |sort +1n +0n
```

**Remarque** - `limreport` comporte plusieurs paramètres. Dans cet exemple, la vérification porte sur "flag.real" (seuls les "vrais" noeuds limites/UID sont recherchés), puis un trait d'union (-) est employé pour indiquer d'utiliser la meilleure estimation du format de sortie, et la liste "uid sgroup lname cpu.shares cpu.usage" signifie que la commande `limreport` doit fournir ces cinq paramètres pour chaque noeud limite dont l'indicateur "flag.real" est fixé à TRUE. La sortie fait l'objet d'un tri primaire UNIX de la deuxième colonne et d'un tri secondaire de la première colonne afin de générer un rapport simple sur les utilisateurs du serveur.

Tout utilisateur ayant la route et les autorisations appropriés peut vérifier l'état de Solaris Resource Manager en tout temps à l'aide de la commande `srmadm show`. Cette commande produit un rapport formaté indiquant l'état actuel des opérations de Solaris Resource Manager, ainsi que ses principaux paramètres de configuration. Ce rapport constitue un moyen pratique de vérifier que Solaris Resource Manager est actif et que tous les paramètres de contrôle sont correctement réglés. Il indique également la valeur des paramètres globaux tels que le taux de décroissance et l'emplacement de la banque de données de Solaris Resource Manager.

Il est possible d'exécuter Solaris Resource Manager sans limites actives et sans ordonnancement de l'UC, ce qui peut être utile lors du démarrage, à des fins de dépannage et de configuration initiale de Solaris Resource Manager :

```
# srmadm set share=n:limits=n, -
```

## Partage d'une machine entre plusieurs services

Un autre groupe de développement voudrait acheter une mise à niveau pour cette machine (plus de processeurs et de mémoire) en échange d'un accès au système lorsqu'il est libre. Les deux groupes bénéficieront de la transaction. Pour configurer cela, créez un nouveau groupe appelé *développement* au même niveau que *bases de données* et *lot*. Attribuez au groupe *développement* 33 % de la machine, car il a augmenté de 50 % la puissance de l'UC et la mémoire du système.

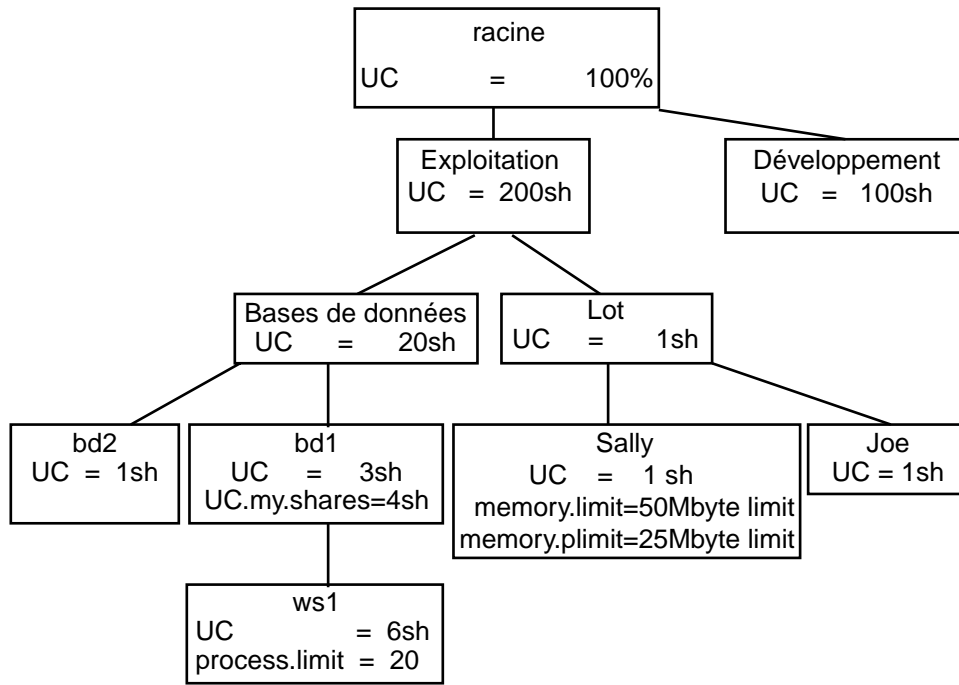


Figure 2-5 Partage d'une machine, étape 1

Le groupe Développement comporte des centaines d'utilisateurs. Pour éviter de définir la distribution de ses ressources, utilisez l'indicateur d'administration de Solaris Resource Manager afin de permettre à l'administrateur de système du groupe Développement d'attribuer ses propres ressources. Vous définissez alors les limites au niveau des opérations et du développement selon l'entente conclue avec l'autre groupe, puis chaque groupe contrôle sa propre part de la machine.

Pour ajouter le nouveau niveau à la hiérarchie, ajoutez le groupe *opérations* en tant que nouveau noeud limite et remplacez les groupes père de *lot* et *bases de données* par *opérations* :

```
% limadm set sgroup=operations batch databases
```

Pour régler l'indicateur d'administration :

```
% limadm set flag.admin=set operations development
```

Puisque, dans des circonstances normales, tous les serveurs ont des démons et des processus de sauvegarde à exécuter, ceux-ci devraient être ajoutés dans un noeud limite distinct de haut niveau.

---

**Remarque** - N'employez pas le niveau racine, car il n'a pas de limites.

---

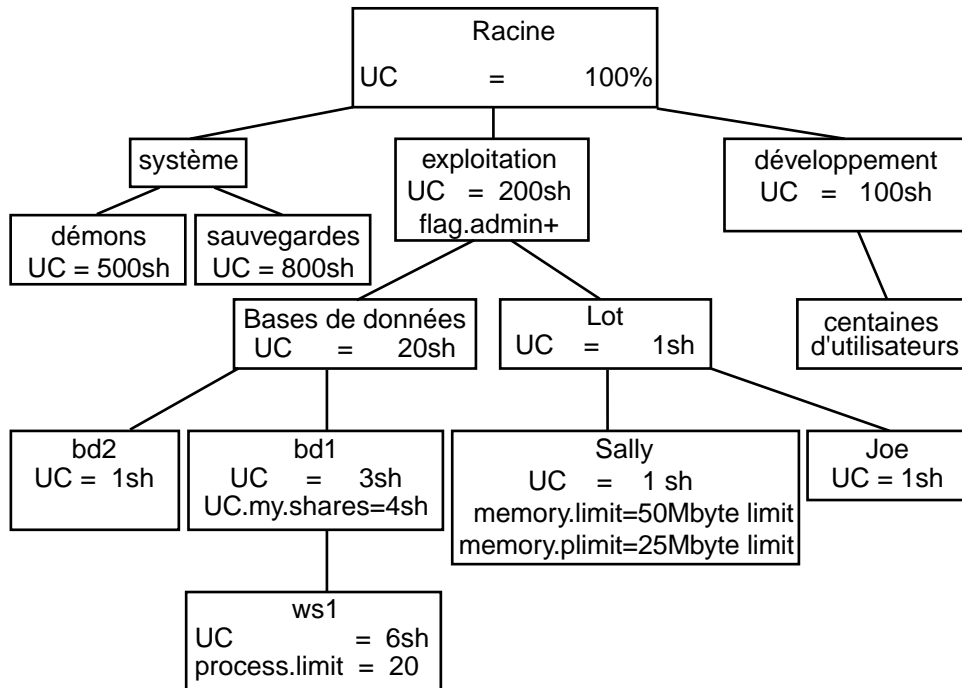


Figure 2-6 Partage d'une machine, étape 2

Comme nous l'avons vu dans les exemples, Solaris Resource Manager vous permet de regrouper divers types d'utilisateurs et d'applications dans la même machine. En utilisant judicieusement les contrôles de partage de l'UC, les limites de mémoire virtuelle, les limites de processus et les contrôles de connexion, vous pouvez vous assurer que les diverses applications ne recevront que les ressources dont elles ont besoin. Grâce aux limites, aucune application et aucun utilisateur ne peut perturber les applications des autres utilisateurs ou groupes d'utilisateurs. Solaris Resource Manager est compatible avec des utilitaires de production de rapports simples permettant aux utilisateurs et aux administrateurs de système de savoir exactement ce qui se passe à un instant donné et durant une certaine période. La production de rapports peut servir à visualiser la répartition de l'usage des ressources entre les applications et les groupes, à des fins de planification de la capacité et de facturation.

# Serveur d'applications type

Le texte suivant serait affiché suite à un listage `liminfo` de `bd1` à la fin de l'exemple de la section précédente. Texte tapé :

```
# liminfo -c db1
```

Résultat :

```
# ./liminfo db1
Login name:          db1      Uid(Real,Eff):      223 (223,223)
Sgroup (uid) :      other (98) Gid(Real,Eff):      50 (50,50)

Shares:              3        Myshares:           6
Share:               60.0%    E-share:            35.4%
Usage:               76000    Accrued usage:      6.4e+08

Mem usage:           11.06 B   Term usage:         0s
Mem limit:           0 B      Term accrue:        0s
Proc mem limit:     0 B       Term limit:         0s
Mem accrue:          13.67 TB.s

Processes:           8        Current logins:     1
Process limit:      0

Last used: Tue Jul 4 15:04:20 1998
Directory: /usr/people/db1
Name: Database1
Shell: /usr/sh/

Flags: userlimadm+
```

*Figure 2-7* `liminfo` Listage

Pour de plus amples renseignements sur les champs décrits ci-après, reportez-vous à `liminfo(1SRM)`.

Les deux premières lignes de la sortie de la commande `liminfo(1SRM)` concernent les aspects de l'UID du noeud limite et sa position dans l'arbre des noeuds limites.

## Nom de connexion

Le nom de connexion et la GID initiale de la table des mots de passe correspondant à l'UID du noeud limite associé. Une UID système est associée à chaque noeud limite. Il est fortement recommandé de créer un compte système pour l'UID de chaque noeud limite. Dans cet exemple, une UID substituable est utilisée pour la Base de données 1 de `bd1`.

Il faut noter que la configuration par défaut du module PAM sous Solaris Resource Manager crée un noeud limite pour tout utilisateur qui ouvre une session sans noeud limite. Par défaut, les noeuds limites créés par un superutilisateur

(racine) ou par un utilisateur dont l'indicateur `uselimadm` est activé sont créés avec le noeud limite de l'utilisateur comme père, ou, si ce dernier n'existe pas, avec le noeud limite racine comme père. Les noeuds limites créés par un utilisateur dont l'indicateur d'administration est activé sont créés avec cet utilisateur comme père. Le père d'un noeud limite peut être changé au moyen de la commande générale de modification des attributs des noeuds limites, `limadm`.

**UID**

L'UID du noeud limite relié au processus en cours. Normalement, cette UID est identique à l'UID réelle du processus (l'utilisateur en session), mais dans certains cas (décrits plus loin), elle peut être différente.

**R,Euid et R,Egid**

L'UID et la GID réelles et effectifs du processus en cours. Il s'agit des mêmes informations que celles fournies par la commande système standard `id(1M)`. Ces champs ne sont pas strictement reliés à Solaris Resource Manager, mais sont affichés à des fins de commodité. Ces champs ne sont toutefois pas affichés si la commande `liminfo(1SRM)` affiche des informations sur un utilisateur autre que celui par défaut (c.-à-d. si vous avez spécifié un nom de connexion ou une UID comme argument).

**Sgroup (uid) [sgroup]**

Le nom et l'UID du noeud limite père dans la hiérarchie des noeuds limites. Ce champ est vide dans le cas du noeud limite racine. Plusieurs fonctions de Solaris Resource Manager dépendent de la position d'un noeud limite dans la hiérarchie de l'arbre. Il est donc utile pour un utilisateur de retracer les noeuds limites père successifs jusqu'à la racine de l'arbre.

Après la ligne en blanc, les deux lignes suivantes de l'affichage de `liminfo(1SRM)` comportent des champs relatifs à l'ordonnement de l'UC.

**Shares [cpu.shares]**

Il s'agit du nombre de parts d'UC attribué à cet utilisateur. Ce nombre ne peut être comparé directement qu'à celui des autres utilisateurs ayant le même noeud limite, ainsi qu'à la valeur "Myshares" du noeud limite père lui-même. Généralement, l'administrateur répartit



également les parts entre tous les utilisateurs appartenant à un groupe d'ordonnancement particulier (afin de donner les mêmes privilèges à tous les utilisateurs). Cette valeur étant normalement supérieure à 1, l'administrateur dispose d'une certaine latitude pour diminuer, au besoin, les parts de certains utilisateurs.

**Myshares [cpu.myshares]**

Cette valeur n'est utilisée que si l'utilisateur possède des noeuds limites enfants (c.-à-d. s'il existe d'autres noeuds limites ayant la valeur `s_group` de cet utilisateur) qui sont actifs (reliés à des processus). Le cas échéant, cette valeur représente la part relative de l'UC attribuée aux processus associés à ce noeud limite, par opposition à ceux qui sont associés à ses noeuds limites enfants.

**Share**

Pourcentage calculé des ressources d'UC attribué à l'utilisateur actuel. À mesure que d'autres utilisateurs ouvrent et ferment une session (ou que des noeuds limites deviennent actifs ou inactifs), cette valeur change, car seuls les utilisateurs actifs sont inclus dans le calcul. L'usage récent de l'utilisateur actuel n'est pas compris dans ce calcul.

**E-Share**

Il s'agit de la part effective de cet utilisateur (c.-à-d. le pourcentage des ressources de l'UC qui serait attribué à cet utilisateur à court terme s'il en avait besoin et si tous les autres utilisateurs actifs exigeaient également leur part). En d'autres termes, il s'agit de la disposition de Solaris Resource Manager à allouer des ressources d'UC à ce noeud limite. Cette valeur varie dans le temps à mesure que l'utilisateur consomme (ou ne consomme pas) des ressources de l'UC. Les noeuds limites actifs mais non actuellement en usage (dont les processus connexes sont au repos) ont une part effective élevée. Réciproquement, la part effective des utilisateurs associés à des processus utilisant actuellement l'UC peut être très faible.

**Usage [cpu.usage]**

Il s'agit de l'usage cumulatif des ressources système servant à déterminer la priorité d'ordonnancement. Généralement, cette valeur

indique l'usage récent de l'UC, bien que d'autres paramètres puissent être pris en compte. La commande `srmadm(1MSRM)` vous permet de connaître les paramètres employés. Chaque accroissement de cette valeur diminue exponentiellement dans le temps, de sorte que Solaris Resource Manager "oublie" l'usage des ressources après un certain temps. Le taux de cette décroissance est représenté par sa demi-vie, laquelle peut être consultée au moyen de la commande `srmadm(1MSRM)`.

#### **Accrued usage [cpu.accrue]**

Cette mesure cumulative est identique à l'"usage", mais n'est jamais réduite. Cette valeur n'est pas directement employée par Solaris Resource Manager, mais peut être utile au personnel de gestion à des fins de comptabilité. Contrairement à l'usage, cette valeur indique l'usage cumulatif de tous les noeuds limites appartenant au groupe, ainsi que celle du noeud limite actuel.

Après la deuxième ligne vierge, les quatre lignes suivantes de l'affichage de la commande `liminfo(1SRM)` présentent quatre champs relatifs à la mémoire virtuelle :

#### **Mem usage [memory.usage][memory.myusage]**

Il s'agit de l'usage de mémoire combiné de tous les processus reliés à ce noeud limite.

Si deux valeurs séparées par une barre oblique (/) sont affichées, ce noeud limite est un chef de groupe, et la première valeur indique l'usage de tout le groupe d'ordonnancement, tandis que la deuxième valeur indique seulement celle de l'utilisateur actuel.

#### **Mem limit [memory.limit]**

Usage maximal de mémoire autorisé pour tous les processus reliés à ce noeud limite et ses membres (s'il y en a). Il s'agit de la somme maximale permise d'usage de mémoire par tous les processus appartenant au groupe et reliés au chef de groupe. Il faut noter que la valeur "0" signifie qu'il n'y a pas de limite.

#### **Proc mem limit [memory.plimit]**

Cette valeur indique l'usage maximal de mémoire autorisée pour tout processus individuel relié à ce noeud limite et à ses membres.

### **Mem accrue [memory.accrue]**

Cette valeur cumulative est exprimée en octets/seconde et indique l'usage global des ressources de mémoire par unité de temps.

Après la troisième ligne vierge, les quatre lignes suivantes de l'affichage de la commande `liminfo(1SRM)` présentent des champs relatifs à l'utilisateur et aux processus.

### **Processes [process.usage][process.myusage]**

Il s'agit du nombre de processus reliés à ce noeud limite. Il faut noter que cette valeur ne tient pas compte du nombre d'unités d'exécution comprises dans un processus.

Si deux valeurs séparées par une barre oblique (/) sont affichées, ce noeud limite est un chef de groupe, et la première valeur indique l'usage de tout le groupe d'ordonnancement, tandis que la deuxième valeur indique seulement celle de l'utilisateur actuel.

### **Process limit [process.limit]**

Nombre total maximum autorisé de processus reliés à ce noeud limite et à ses membres.

### **Current logins [logins]**

Nombre actuel de sessions simultanées de Solaris Resource Manager pour cet utilisateur. Lorsqu'un utilisateur ouvre une session à l'aide d'un des mécanismes standard (y compris `login(1)`, `rlogin(1)`, etc., ou toute méthode utilisant un module PAM pour l'authentification et créant une entrée `utmp(4)`), ce compteur est incrémenté. À la fin de la session, le compte est décrémenté.

Si l'indicateur `flag.onelogin` d'un utilisateur est activé, cet utilisateur ne peut ouvrir qu'une seule session de Solaris Resource Manager.

### **Last used [lastused]**

Ce champ indique la dernière fois où le noeud limite a été actif. Il s'agit normalement de la dernière fermeture de session effectuée par l'utilisateur.

### **Directory**

Répertoire d'accueil de l'utilisateur (les articles relatifs à la table des mots de passe plutôt qu'à Solaris Resource Manager sont indiqués à des fins de commodité).

**Name**

Informations relatives à *bd1* (servant au repérage). Il s'agit généralement du nom de l'utilisateur (les articles relatifs à la table des mots de passe plutôt qu'à Solaris Resource Manager sont indiqués à des fins de commodité).

**Shell**

Shell de connexion initial de l'utilisateur (les articles relatifs à la table des mots de passe plutôt qu'à Solaris Resource Manager sont indiqués à des fins de commodité).

**Flags**

Les indicateurs activés ou ayant la valeur d'un groupe dans le noeud limite sont affichés ici. Chaque indicateur affiché est suivi de caractères indiquant la valeur et la manière dont l'indicateur a été activé (par exemple, s'il a été activé explicitement depuis ce noeud limite [+] ou s'il a été hérité [^]).

## Configuration

---

La configuration de Solaris Resource Manager est extrêmement flexible pour l'administrateur central (par exemple, l'utilisateur racine). Ce chapitre décrit les éléments de configuration suivants :

- Paramètres d'initialisation du noyau, utilisés lors du démarrage du noyau (voir "Paramètres d'initialisation du noyau" , page 33).
- Paramètres globaux de Solaris Resource Manager spécifiés au moyen de la commande `srmadm(1MSRM)` (voir "Paramètres globaux de Solaris Resource Manager - commande `srmadm`" , page 36).
- Paramètres communiqués au programme `limdaemon(1MSRM)` (voir "Options de la commande `limdaemon`" , page 39).
- Sous-système PAM et gestion des comptes (voir "Sous-système PAM" , page 40).

---

## Paramètres d'initialisation du noyau

Le noyau comporte certains paramètres de Solaris Resource Manager pouvant être définis par l'administrateur central lors de l'amorçage du noyau. Lors de l'amorçage, Solaris lit le fichier `/etc/system` et l'utilise pour configurer les modules du noyau (voir `system(4)` pour plus de détails). Voici les paramètres pouvant être définis dans le module SHR (qui sont tous des nombres entiers à 32-bits) pour remplacer les réglages par défaut de Solaris Resource Manager :

**SRMLnodes**

Nombre de noeuds limites à mettre en mémoire cache dans le noyau. Dans les systèmes Solaris, chaque noeud limite du noyau exige environ 3

ko. La valeur zéro (la valeur par défaut) signifie que le noyau déterminera la valeur. En pareil cas, voici l'approche employée :

```
(nproc / SRMProcsPerUid ) + SRMLnodesExtra
```

où *nproc* est le nombre maximum de processus simultanés autorisés dans le système. La valeur minimum 6 a priorité sur ce calcul. Le maximum spécifié par SRMMemoryMax a également priorité sur ce calcul.

**SRMProcsPerUid**

Nombre moyen prévu de processus par utilisateur. La valeur par défaut est 4.

**SRMLnodesExtra**

Biais utilisé dans l'approche heuristique visant à déterminer la taille du tableau de noeuds limites en mémoire. La valeur par défaut est 20.

**SRMNhash**

Nombre d'entrées dans la table de hachage utilisé pour le mappage des valeurs UID avec les noeuds limites dans le noyau. Sous Solaris, chaque entrée a une longueur de 4 octets. La valeur par défaut est zéro, ce qui indique d'utiliser la même valeur que pour le nombre de noeuds limites.

**SRMMemoryMax**

L'inverse de cette valeur est une fraction indiquant le pourcentage maximum de mémoire réelle à utiliser pour le noeud limite et les tables de hachage de Solaris Resource Manager. La valeur par défaut est 20, ce qui signifie qu'un maximum de 5 % de la mémoire réelle sera attribué aux structures de données de Solaris Resource Manager.

**SRMMemWarnFreq**

Intervalle minimum, en secondes, entre les avertissements de dépassement de mémoire pour un noeud limite. La valeur par défaut est 4.

Par exemple, dans le fichier `/etc/system`, la ligne

```
set srmlim:SRMMemWarnFreq=10
```

indique que les messages de dépassement de mémoire doivent être envoyés au maximum toutes les 10 secondes par utilisateur.

Il existe également certains paramètres ne faisant pas partie de Solaris Resource Manager, mais qui en influencent le comportement, par exemple :

**initclass**

Nom de la classe d'ordonnement où le processus `init(1M)` est lancé. Sous Solaris Resource Manager, ce paramètre devrait avoir la valeur "SHR" (y compris les guillemets). La valeur par défaut sous Solaris est "TS". Si vous voulez que Solaris Resource Manager contrôle les ressources de l'UC, incluez la ligne suivante dans le fichier `/etc/system` :

```
set initclass="SHR"
```

afin de remplacer la valeur par défaut.

**extraclass**

Nom d'un module de classe d'ordonnement à charger, sans nécessairement l'utiliser en tant que classe d'ordonnement par défaut. Pour que Solaris Resource Manager ne contrôle que les ressources autres que l'UC, incluez la ligne suivante dans le fichier `/etc/system` :

```
set extraclass="SHR"
```

Pour initialiser un système sans charger Solaris Resource Manager, il ne faut pas utiliser `/etc/system`, mais plutôt `/etc/system.noshrlod`. Pour en savoir davantage, consultez "Initialisation sans Solaris Resource Manager" , page 45.

## Configuration de démarrage multi-utilisateur

Lors d'un démarrage normal du système, lorsque le système passe du mode mono-utilisateur au mode multi-utilisateur, un script de démarrage de Solaris Resource Manager est exécuté afin de régler divers paramètres de Solaris Resource Manager. Pour obtenir plus de détails sur ce script, consultez Chapitre 4.

Si vous modifiez le script d'initialisation (`/etc/init.d/init.srm`), conservez des copies distinctes des versions originale et modifiée. Les mises à jour de Solaris Resource Manager ne conservent pas nécessairement les scripts d'initialisation existants.

## Paramètres globaux de Solaris Resource Manager - commande `srmadm`

La commande `srmadm(1MSRM)` permet à l'administrateur de définir, de modifier et d'afficher les paramètres globaux de Solaris Resource Manager. Consultez la page de manuel pour obtenir des détails complets sur tous les paramètres.

Vous pouvez exécuter la commande `srmadm(1MSRM)` autant de fois que nécessaire pour régler divers paramètres. Il n'est pas nécessaire de tous les inclure dans un même appel. La commande `srmadm(1MSRM)` vous permet également de modifier les paramètres d'exploitation durant l'exécution du système Solaris Resource Manager, bien que certaines précautions doivent être prises.

Les options `srmadm(1MSRM)` d'activation et de désactivation des fonctions principales de Solaris Resource Manager ont une importance particulière pour les administrateurs. Voici ces options :

`fileopen[={y|n}]`

La base de données par défaut, `/var/srm/srmDB`, peut être remplacée au moyen de l'option `-f`. Il faut noter que la fermeture du fichier de base de données de Solaris Resource Manager en cours de fonctionnement est une action d'urgence, car elle a des conséquences indésirables : tous les processus continueront de fonctionner sur le noeud limite racine délégué, ce qui pourrait leur accorder des privilèges trop élevés ; l'ordonnanceur de l'UC est désactivé ; Solaris Resource Manager cesse d'appliquer les limites. Lorsque cette option est activée, Solaris Resource Manager n'ouvre pas de base de données des limites, et sa mémoire cache ne contient que le noeud limite racine délégué auquel tous les processus sont reliés.

`share[={y|n}]`

Lorsque cette option est activée, l'ordonnanceur d'UC de Solaris Resource Manager est utilisé, et l'ordonnement de l'UC est effectué en fonction de l'algorithme d'usage dynamique et de décroissance de Solaris Resource Manager. Ce mode peut être choisi uniquement lorsque le mode `fileopen` est activé. Si cette option est désactivée, les calculs d'usage de l'ordonnanceur d'UC de Solaris Resource Manager cessent, et les processus sont ordonnancés avec priorités égales fixes, selon un mode de recherche circulaire.

`limits[={y|n}]`

Lorsque cette option est activée, Solaris Resource Manager applique les limites de mémoire virtuelle et de processus. Ce mode peut être choisi uniquement lorsque le mode `fileopen` est activé. Si cette option est désactivée, Solaris Resource Manager continue de mettre à jour les attributs d'usage, mais n'applique pas les limites.

`adjgroups[={y|n}]`

Lorsque cette option est activée, l'ajustement global des parts effectives des groupes de l'ordonnanceur d'UC de Solaris Resource Manager est employé. Dans la plupart des cas, il est recommandé d'activer cette option. Durant chaque intervalle d'exécution, les usages normalisés de toutes les entrées de limite sont recalculés. Si le



mode d'ordonnement `adjgroups` est activé, un traitement supplémentaire des usages normalisés est effectué comme suit. L'ordonneur effectue une passe sur l'arbre d'ordonnement, en comparant la part réelle récente de chaque groupe à sa part attribuée. Les groupes qui ont reçu moins que leur juste part sont biaisés de manière à recevoir une part réelle supérieure lors de l'intervalle d'exécution suivant. Ainsi, les groupes recevront leur part équitable des ressources de l'UC lorsque possible, quelles que soient les actions de leurs membres.

`limshare[={y|n}]`

Lorsque cette option est activée, l'ordonneur d'UC de Solaris Resource Manager applique un plafond de priorité afin de limiter les parts effectives de tous les utilisateurs pour empêcher les utilisateurs ayant une très faible attribution d'acquiescer brièvement presque 100 % de l'UC. L'état activé est recommandé.

Le taux de service de l'UC d'un utilisateur est à peu près inversement proportionnel à son usage. Si un utilisateur n'a pas été actif depuis longtemps, son usage diminue à près de zéro. Lorsque cet utilisateur ouvre une session (ou que le noeud limite devient actif d'une manière quelconque), durant l'intervalle d'exécution suivant, la priorité des processus de l'utilisateur pourrait être suffisamment élevée pour monopoliser l'UC.

Si l'indicateur d'ordonnement `limshare` est activé, l'ordonneur évalue la part réelle qu'un noeud limite recevra avant l'intervalle d'exécution suivant. Si le résultat dépasse la part attribuée à l'utilisateur par un certain facteur (voir `maxushare`), l'usage normalisé de l'utilisateur est modifié de manière à empêcher cela.

Il existe aussi deux paramètres facultatifs de la commande `srmadm(1MSRM)` utiles à l'administrateur. Voici ces paramètres :

- Le paramètre `-v` permet d'imprimer un rapport formaté de tous les réglages de paramètres dans la sortie standard. Si deux ou trois options `-v` sont spécifiées, le rapport comporte de plus en plus d'informations. Si la commande `srmadm(1MSRM)` est exécutée sans arguments, cela équivaut à spécifier une seule option `-v`.
- Le paramètre `-d` permet d'initialiser la structure du système Solaris Resource Manager avec les valeurs par défaut, plutôt que de lire les réglages actuels du noyau. Les valeurs par défaut, permettant principalement de contrôler le comportement d'ordonnement, sont intégrées à la commande `srmadm(1MSRM)` et constituent un bon point de départ pour la personnalisation de Solaris Resource Manager. Le noyau démarre avec les mêmes valeurs pré-réglées.

Voici des exemples représentatifs de commandes `srmadm(1MSRM)`.

Pour lancer Solaris Resource Manager en activant l'ordonneur d'UC et les limites de ressources :

```
# srmadm set -f /var/srm/srmDB fileopen=y:share=y:limits=y
```

Pour régler le taux de décroissance d'usage de l'UC avec une demi-vie de 5 minutes :

```
# srmadm set usagedecay=300s
```

Pour afficher les réglages actuels des indicateurs et les charges :

```
% srmadm
```

Pour afficher tous les réglages par défaut :

```
% srmadm show -dv
```

## Désactivation de Solaris Resource Manager

La commande `srmadm(1MSRM)` permet de désactiver Solaris Resource Manager en désactivant l'indicateur "fileopen" ; tous les processus sont transférés vers le noeud limite racine délégué, les autres noeuds limites modifiés dans la mémoire cache sont transférés sur le disque, et le fichier du noeud limite est fermé. Cela désactive automatiquement les indicateurs de part et de limites, ainsi que, respectivement, l'ordonnanceur d'UC de Solaris Resource Manager et l'application des limites. Au besoin, vous pouvez désactiver séparément les indicateurs de part et de limites en laissant le fichier du noeud limite ouvert. Cela est préférable à la fermeture du fichier, car les processus peuvent demeurer reliés à leur noeud limite correspondant.

Il faut noter que si seul l'ordonnanceur de Solaris Resource Manager est désactivé en cours d'exécution, cela n'aura pour effet que d'interrompre l'algorithme d'usage et de décroissance. L'ordonnanceur continue de traiter les processus dans la classe d'ordonnement SHR, mais à mesure que chacun d'eux reçoit une nouvelle priorité, la même valeur est employée, ce qui produit un ordonnancement simple à "recherche circulaire".

Si vous réactivez Solaris Resource Manager en ouvrant le fichier et en réglant les indicateurs de part ou de limite après la fermeture du fichier, les processus existants ne sortiront pas du noeud limite racine. Il est déconseillé de fermer la base de données de Solaris Resource Manager au cours de son fonctionnement normal. Autrement, vous devrez réinitialiser le système pour vous assurer que les processus sont correctement reliés aux noeuds limites.

---

# Options de la commande `limdaemon`

`limdaemon(1MSRM)` comporte plusieurs options pouvant être définies lors de son exécution :

- L'option `-l` indique à la commande `limdaemon(1MSRM)` de consigner les messages au moyen de `syslog(3)` —.
- L'étiquette `-m` et l'option `-p` sont utilisées conjointement avec l'option `-l` pour étiqueter les messages et de contrôler leur acheminement selon la configuration de la commande `syslogd(1M)`.
- L'option `-s` indique à la commande `limdaemon(1MSRM)` d'horodater les messages (sauf ceux produits par la commande `syslog(3)`, qui sont déjà horodatés).
- L'option `-c` indique à la commande `limdaemon(1MSRM)` de supprimer la mise à jour des usages de temps de connexion de terminal.
- L'option `-d` indique à la commande `limdaemon(1MSRM)` de décroître les usages de temps de connexion pour tous les terminaux des utilisateurs en session, l'intervalle entre les décroissances étant l'argument de l'option `-t` (1 minute par défaut).
- L'option `-Dn` indique à la commande `limdaemon(1MSRM)` de décroître les usages de temps de connexion pour tous les terminaux et tous les utilisateurs, toutes les  $n$  minutes.
- L'option `-k` met fin à l'exécution de la commande `limdaemon(1MSRM)`.
- L'option `-t` peut être employée pour définir la période (en minutes) entre les mises à jour de l'attribut d'usage de temps de connexion dans la catégorie du périphérique terminal. La valeur par défaut est 1 minute.
- L'option `-e` peut être utilisée pour supprimer la déconnexion des utilisateurs ayant atteint leur limite de temps de connexion. Cette option est implicitement activée par l'option `-c`.
- L'option `-w` permet de régler le nombre de minutes avant l'expiration du temps de connexion où le message d'avertissement est affiché. Le délai d'avertissement par défaut est de 5 minutes.
- L'option `-g` vous permet de régler le temps de grâce (en secondes). Le temps de grâce par défaut est de 30 secondes.

L'administrateur devrait déterminer l'équilibre nécessaire entre les ressources supplémentaires requises pour la mise à jour rapide des attributs d'usage du temps de connexion, et l'augmentation de granularité causée par des mises à jour moins fréquentes. Consultez la page de manuel `limdaemon(1MSRM)` pour de plus amples renseignements sur ces options et d'autres.

Par exemple, la commande :

```
% limdaemon -g300
```

lance le démon et règle le temps de grâce à 5 minutes. Il faut noter qu'il n'est pas nécessaire d'ajouter le caractère de shell "&" à la commande. Lorsque `limdaemon` est lancé, il devient un démon. En d'autres termes, un processus enfant est créé et se détache du terminal de contrôle pour se placer dans son propre groupe de traitement.

---

## Sous-système PAM

Les systèmes Solaris 2.6 prennent en charge les modules PAM (Pluggable Authentication Modules). Lorsqu'un utilisateur veut effectuer une opération dans laquelle l'identité de l'utilisateur est modifiée ou définie (par exemple la connexion au système, l'exécution d'une commande "r" telle que `rcp` ou `rsh`, l'emploi de `ftp` ou de `su`), un ensemble de modules configurables est utilisé pour effectuer l'authentification, la gestion des comptes, la gestion des justificatifs d'identité et la gestion des sessions. Solaris Resource Manager comporte un module permettant de comptabiliser les connexions et de modifier le comportement de la commande `su`.

Le programme servant à demander l'opération est appelé le service.

Le système PAM global est décrit dans les pages de manuel `pam.conf(4)`, `pam(3)`, `pam_unix(5)` et `pam_srm(5SRM)`.

Le module PAM de Solaris Resource Manager effectue des fonctions de gestion des comptes et des sessions. Vous pouvez configurer le comportement du module PAM en modifiant le fichier `/etc/pam.conf`. Pour un comportement normal de Solaris Resource Manager, le module PAM devrait être configuré comme exigence pour tous les services de connexion, la gestion des sessions et la gestion des comptes pour tous les services PAM. Généralement, le module Solaris Resource Manager doit être placé après tous les autres modules requis ou exigés, et avant tout autre module suffisant ou optionnel.

Lors de l'installation, Solaris Resource Manager modifie le fichier `/etc/pam.conf` de manière à assurer un comportement approprié. Il insère des lignes comme celles-ci pour chaque service (y compris `other`) dont la gestion des sessions ou des comptes est déjà configurée :

```
login account requisite pam_srm.so.1 nolnode=/etc/srm/nolnode
login session requisite pam_srm.so.1
other account requisite pam_srm.so.1 nolnode=/etc/srm/nolnode
```

La première ligne indique que, pour la connexion des services, le module `pam_share.so.1` doit être utilisé pour la gestion des comptes, qu'il doit permettre la connexion pour que celle-ci réussisse, et qu'il doit recevoir l'argument `nolnode=/etc/srm/nolnode`. Reportez-vous à la section `pam.conf(4)` pour une explication complète des divers indicateurs de contrôle (`required`, `requisite`, `optional` et `sufficient`).

La deuxième ligne indique que le service de connexion utilisera le module `pam_share.so.1` pour la gestion des sessions.

La liste complète des arguments des modules de gestion des comptes et des sessions de Solaris Resource Manager est présentée à la section `pam_srm(5SRM)`.

---

## Gestion des comptes

Lorsque le module PAM de gestion des comptes de Solaris Resource Manager obtient le contrôle :

1. Il détermine si Solaris Resource Manager est installé et activé ; sinon, il indique au système PAM de ne pas tenir compte de ce module.
2. Il détermine si l'utilisateur possède un noeud limite ; sinon, il lance un script de "noeud limite absent" configurable par l'administrateur.
3. Il détermine si l'utilisateur a la permission d'utiliser le service et le périphérique demandés.
4. Il détermine si l'utilisateur a dépassé la limite d'avertissement ; le cas échéant, il lui interdit d'ouvrir une session.
5. Il lance un script "toutes sessions" configurable par l'administrateur.

Si l'une de ces étapes échoue, les autres ne sont pas effectuées, et le module PAM de gestion des comptes de Solaris Resource Manager interdit l'usage du service. Si possible, un message explicatif est envoyé à l'utilisateur par le service.

## Scripts

Le script "noeud limite absent" par défaut crée un noeud limite pour l'utilisateur et en avise l'administrateur de système par courrier. Le script par défaut est `/etc/srm/nolnode`, mais il peut être remplacé en modifiant le fichier `/etc/pam.conf` et en changeant la valeur de l'option `nolnode` dans les lignes du module de gestion des comptes de Solaris Resource Manager. Le script "toutes sessions" n'est habituellement pas configuré. Vous pouvez cependant le configurer en ajoutant une option `everylogin=pathname` à tout module de gestion des comptes de Solaris Resource Manager dans le fichier `/etc/pam.conf`. Vous exécutez les scripts en tant d'utilisateur racine. L'entrée, la sortie et l'erreur standard sont fermées. Si un

script se termine avec une valeur non nulle, l'accès n'est pas accordé. Toutes les informations sont transmises sous forme de variables d'environnement, lesquelles sont dérivées directement des informations transmises au module PAM par le service :

USER	Nom de connexion fourni au programme. Il a été authentifié par consultation dans la table des mots de passe ; s'il est absent, le module de gestion des comptes a déjà envoyé un code d'erreur au module PAM.
UID	ID de l'utilisateur qui est authentifié. Dans le cas des services qui changent l'UID (par exemple "su"), il s'agit de l'UID de l'utilisateur qui appelle le service; dans le cas des services qui définissent l'UID (par exemple login), il s'agit de l'UID cible (celui de USER).
RHOST	Lors des tentatives d'accès dans un réseau, cette variable contient le nom de l'hôte à l'origine de la tentative. Autrement, sa valeur dépend de la mise en oeuvre.
SERVICE	Nom du service d'accès, par exemple rsh, login, ftp, etc.
TTY	Nom du télécype où le service est appelé. Certains services n'ayant pas (strictement parlant) de terminal de contrôle (comme ftp) donneront à cette variable une certaine valeur (par exemple ftp12345, où 12345 est l'identificateur de processus (PID) de ftpd) ; tandis que d'autres la laisseront vide ou la remplaceront par le nom du service.
DEBUG	Si "debug" a été précisé dans le fichier pam.conf, la variable DEBUG a la valeur "true" ; sinon, elle a la valeur "false". Comme aucune autre variable d'environnement n'est réglée, les scripts doivent, au besoin, définir leur propre variable PATH.

Le script par défaut "noeud limite absent" crée le noeud limite dans le groupe d'ordonnement par défaut ( other, si un tel utilisateur existe dans la table des mots de passe, ou sinon root) et rappelle par courrier à l'administrateur du système de déplacer le nouveau noeud limite à l'endroit approprié dans la hiérarchie d'ordonnement. Pour consulter un exemple de script, voir "Script "pas de noeud limite" par défaut" , page 99.

# Interaction du module PAM avec des groupes de périphériques

Le module PAM de Solaris Resource Manager consulte les noms de terminal et de service dans la hiérarchie des périphériques ; si des limites sont dépassées ou qu'un indicateur de périphérique a la valeur "activé", la permission n'est pas accordée.

Les catégories de périphérique examinées sont : `terminal` pour le nom du terminal, et `services` pour le type de service demandé. Par exemple, une tentative de connexion "rlogin" peut essayer d'utiliser un fichier dans le groupe de périphériques du réseau ; ainsi, les indicateurs vérifiés pour l'utilisateur (en supposant que tous les indicateurs ont la valeur `group`) sont indiqués ci-dessous. Les indicateurs suivants sont vérifiés dans l'ordre :

- `terminal.flag.network`
- `terminal.flag.all`
- `services.flag.rlogin`
- `services.flag.netsservices`

L'accès n'est accordé que si tous ces indicateurs ont la valeur "activé". En outre, les limites sont vérifiées en fonction des catégories correspondantes ( `terminal` et `services`).

## Gestion des sessions

Pour les services de connexion (créant une entrée dans le fichier `utmp`), les fonctions de gestion des sessions du module PAM sont appelées, ainsi que les fonctions de gestion des comptes si les deux sont configurées dans `/etc/pam.conf`.

La fonction de gestion des sessions de Solaris Resource Manager effectue le chargement des périphériques. Elle vérifie si l'utilisateur a dépassé la limite de temps de connexion ou si son indicateur `onelogin` est activé et que l'utilisateur est déjà en session ; le cas échéant, elle l'empêche.

Autrement, elle envoie un message au processus `limdaemon` afin de l'informer de la connexion et du coût configuré pour le terminal où la connexion est effectuée. Elle indique ensuite au noyau que le processus courant est un "processus d'en-tête de session", dont le processus `limdaemon` doit être informé lors de sa terminaison.

Le processus `limdaemon` fait alors le suivi des limites de temps de connexion, et génère des avertissements en cas de dépassement prochain.





## Procédure d'initialisation

---

Durant la procédure d'initialisation d'UNIX, diverses facettes de Solaris Resource Manager sont activées à divers moments. Voici les étapes principales :

- Lors du démarrage du noyau, divers paramètres sont chargés depuis le fichier `/etc/system`. Certains d'entre eux influencent Solaris Resource Manager. Ils sont détaillés ci-après.
- Lorsque le noyau poursuit son initialisation, après la création du processus 0 mais avant celle du processus 1, Solaris Resource Manager est initialisé. Cela inclut le chargement du module SHR et l'ordonnancement du processus 1 (le processus `init(1M)`) par Solaris Resource Manager. Cela est effectué en lançant `init(1M)` dans la classe d'ordonnancement SHR, plutôt que dans la classe d'ordonnancement par défaut.
- Initialement, le processus `init(1M)` et tous ses enfants sont reliés au noeud limite racine délégué.
- Lorsque le noyau est complètement initialisé, le système passe du mode mono-utilisateur à l'un des modes multi-utilisateur (généralement le niveau d'exécution 2 ou 3). Au début de cette procédure, le script `/etc/init.d/init.srm` est exécuté. Les actions effectuées par ce script sont décrites à la section "Événements de la séquence d'initialisation", page 46 et mettent en oeuvre les opérations normales de Solaris Resource Manager.

---

## Initialisation sans Solaris Resource Manager

S'il est nécessaire d'amorcer le système sans activer Solaris Resource Manager, il suffit de modifier la variable `initclass` dans le fichier `/etc/system` afin de la régler

au partage de temps (TS) plutôt qu'à SHR. Pour ce faire, vous pouvez utiliser l'option `-a` de la commande `boot(1M)`, et vous serez invité à spécifier un fichier système. Aux autres invites, appuyez simplement sur la touche Entrée afin d'accepter les valeurs par défaut, jusqu'à l'invite vous demandant le nom du fichier système. Tapez alors `etc/system.noshpload` (sans barre oblique initiale). Voici un exemple de cette procédure :

```
ok boot -a
Booting from: sd(0,0,0) -a
Enter filename [kernel/unix]:
Enter default directory for modules
[/platform/SUNW,UltraSPARC/kernel /kernel /usr/kernel]:
SunOS Release 5.6 Version ... [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1997, Sun Microsystems, Inc.
Name of system file [etc/system]: etc/system.noshpload
root filesystem type [ufs]:
Enter physical name of root device
[/sbus@1,f8000000/esp@0,800000/sd@3,0:a]:
```

Il faut noter que `/etc/system.noshpload` est simplement une copie de sauvegarde de `/etc/system` effectuée lors de l'installation de Solaris Resource Manager. Si vous avez ultérieurement modifié `/etc/system`, le fichier `/etc/system.noshpload` doit être maintenu en parallèle de manière à ce qu'il ne diffère que par l'occurrence de la modification de Solaris Resource Manager :

```
# diff /etc/system /etc/system.noshpload
< # enable srm
< set initclass="SHR"
```

---

## Événements de la séquence d'initialisation

La séquence des événements après l'initialisation du système lors du passage au mode multi-utilisateur est particulièrement importante dans Solaris Resource Manager. Les étapes ci-dessous présentent une séquence établissant correctement le système Solaris Resource Manager :

1. Configurez et activez Solaris Resource Manager au moyen de la commande `srmadm(1MSRM)`. À ce moment, la base de données des limites est ouverte, et l'ordonnanceur de Solaris Resource Manager est activé. Consultez la section "Activation de Solaris Resource Manager à l'aide de la commande `srmadm`",

page 48 pour des renseignements sur l'utilisation de la commande `srmadm(1MSRM)` afin d'activer Solaris Resource Manager.

2. Attribuez les noeuds limites "lost" et "idle".
3. Lancez le démon de Solaris Resource Manager. Consultez la section "Lancement du démon Solaris Resource Manager" , page 49 pour obtenir des renseignements sur cette procédure.
4. Lancez les autres démons du système sur un noeud limite approprié.

Le script utilisé par défaut dans les étapes 1 à 3 de la procédure ci-dessus est présenté à l'annexe.

---

## Processus démons du système

La liaison de démons (processus de maintenance du système normalement exécutés en permanence) à un noeud limite autre que le noeud limite racine est particulièrement importante. Les processus reliés au noeud limite racine ont un ordonnancement spécial et obtiennent toujours toutes les ressources de l'UC qu'ils demandent; il est donc déconseillé d'associer au noeud limite racine tout processus pouvant de faire un usage intensif de l'UC. En liant les démons à leur propre noeud limite, l'administrateur central peut leur attribuer une part appropriée de l'UC.

Au cours de la procédure d'initialisation, chaque nouveau processus hérite sa liaison au noeud limite de son processus père. Étant donné que le processus `init(1M)` est relié au noeud limite racine, tous les processus subséquents le sont aussi. Lorsque le script d'initialisation de Solaris Resource Manager est exécuté et que la base de données des noeuds limites est ouverte, les processus peuvent alors être reliés à d'autres noeuds limites, mais seulement lorsqu'un processus effectue un appel système `setuid(2)` explicite (comme le fait `login(1)`) ou demande explicitement à Solaris Resource Manager de se lier à un noeud limite particulier, comme le fait la commande `srmuser(1SRM)`. L'exécution d'un programme avec le bit de mode du fichier `setuid` activé n'entraîne pas de changement de liaison au noeud limite.

Ainsi, tous les programmes système lancés automatiquement lors du démarrage du système seront reliés au noeud limite racine. Cela est souvent indésirable, car tout processus relié au noeud limite racine faisant un usage intensif de l'UC perturbera fortement l'exécution des autres processus. Il est donc recommandé de lier explicitement tout processus démon lancé lors de la procédure d'initialisation à son propre noeud limite au moyen de la commande `srmuser(1SRM)`. Cela n'a aucun effet sur leur UID réelle ou effective.

En voici un exemple :

```
# /usr/srm/bin/srmuser Start in.named attached to the my_daemons lnode.
```

Ces lignes pourraient remplacer l'appel existant du démon `named(1M)` dans son script de démarrage. Pour ce faire, un compte utilisateur et un noeud limite pour "network" devraient préalablement être définis.

---

## Activation de Solaris Resource Manager à l'aide de la commande `srmadm`

La commande `srmadm(1MSRM)` permet à l'administrateur de contrôler le mode d'exploitation et la configuration globale du système Solaris Resource Manager. Cette commande est généralement employée lors de la transition au niveau d'exécution 2 ou 3 dans le script `init.d(4) /etc/init.d/init.srm` de Solaris Resource Manager afin de s'assurer que les valeurs de tous les paramètres seront appropriées lors du démarrage du système, et que le système Solaris Resource Manager sera activé avant que les utilisateurs ne puissent accéder au système. La commande `srmadm(1MSRM)` permet également d'administrer les paramètres globaux de Solaris Resource Manager. Consultez la page de manuel `srmadm(1MSRM)` pour connaître les paramètres réglables au moyen de la commande `srmadm`. Les commandes `srmadm(1MSRM)` lancées dans le script `init.d(4)` de Solaris Resource effectuent les actions suivantes :

- Ouverture de la base de données des limites. Jusqu'à ce moment, tout processus lancé est automatiquement relié à un noeud limite racine délégué. Le noeud limite racine délégué vous permet de vous assurer qu'un noeud limite est toujours disponible pour la liaison de processus, quel que soit l'état d'exploitation de Solaris Resource Manager. Pour cette raison, il est important que la base de données des limites soit ouverte avant le démarrage des processus non-racine. Lors de l'ouverture de la base de données des limites, les valeurs des attributs d'usage du noeud limite racine délégué sont ajoutées à leurs homologues dans le noeud limite racine. Cependant, avec cette méthode, les diminutions nettes d'usage ne sont pas prises en compte. Ainsi, les modifications d'usage antérieures à l'ouverture de la base de données des limites ne sont pas annulées.
- Application des limites.
- Définition des paramètres contrôlant le comportement de l'ordonnanceur SHR de Solaris Resource Manager, par exemple le taux de décroissance de l'usage.
- Activation de l'ordonnanceur de Solaris Resource Manager. Antérieurement, les processus appartenant à la classe d'ordonnement SHR étaient programmés en mode de recherche circulaire simple, et les attributions de ressources de l'UC définies dans le système Solaris Resource Manager n'avaient aucun effet.

La section "Paramètres globaux de Solaris Resource Manager - commande `srmadm`", page 36 décrit certains appels courants de la commande `srmadm(1MSRM)`.

---

# Lancement du démon Solaris Resource Manager

Le programme `limdaemon(1MSRM)` est le démon du mode utilisateur de Solaris Resource Manager. Il est normalement appelé lors de la transition au niveau d'exécution 2 ou 3 en tant que dernière étape du script `init.d(4)` de Solaris Resource Manager. Il ne faut pas le confondre avec le processus système `srmgr` (dans la classe `SYS`), lancé par le noyau. Le listage `ps(1)` suivant illustre ces deux processus :

```
# ps -efc | egrep 'limdaemon|srmgr'
root  4  0  SYS 60 18:42:14 ?        0:05 srmgr
root  92  1  SHR 19 18:42:32 ?        0:41 limdaemon
```

Le programme `limdaemon` remplit les fonctions suivantes :

- il reçoit les messages de notification et les transmet aux terminaux des utilisateurs destinataires ;
- il reçoit les messages de notification de connexion ou de déconnexion, et conserve un registre précis de toutes les sessions de Solaris Resource Manager en cours ;
- il met périodiquement à jour les temps de connexion de tous les utilisateurs actuellement en session sous Solaris Resource Manager (facultatif) ;
- il détecte les utilisateurs ayant atteint leur limite de temps de connexion, détruit le processus, et ferme leur session (facultatif) après un intervalle de grâce ;
- il consigne toutes les actions au moyen de `syslog(3)` dans `syslogd(1M)`.

Lorsqu'il est avisé des ouvertures de session de Solaris Resource Manager, `limdaemon` surveille le temps de connexion au terminal de tous les utilisateurs et le compare à leur limite de temps de connexion. Lorsque leur limite de temps de connexion est presque atteinte, ils reçoivent un message d'avertissement. Lorsque le temps est expiré, une période de grâce supplémentaire leur est accordée avant l'arrêt de tous leurs processus et la fermeture de leur session.

Le programme `limdaemon` diminue progressivement les usages de temps de connexion. La décroissance d'usage pour la catégorie de périphériques terminaux doit être effectuée si des limites de temps de connexion sont employées. Consultez la section "Options de la commande `limdaemon`", page 39 pour obtenir des détails sur les options de ligne de commande permettant de contrôler le comportement de `limdaemon`.



## Gestion des noeuds limites

---

Le système Solaris Resource Manager est conçu autour d'un ajout fondamental au noyau : une structure de niveau utilisateur appelée noeud limite (lnode). Chaque UID définie dans la table des mots de passe doit avoir un noeud limite correspondant (soit chaque UID unique renvoyée par appels `getpwent(3C)` successifs). Bien que cela ne soit pas recommandé, un noeud limite peut exister sans entrée correspondante dans la table des mots de passe. Les noeuds limites sont stockés sur disque et transférés vers et depuis la mémoire par le noyau. Les copies de noeuds en mémoire qui ont changé depuis leur lecture initiale à partir du disque sont réécrites dans le cadre des opérations de synchronisation système habituelles, ainsi que sur demande, lors de l'exécution de la commande `sync(1M)` et, lorsque nécessaire, afin de libérer de l'espace de cache pour la lecture d'autres noeuds limites. Un noeud limite est en fait un emplacement de taille fixe dans lequel nombre de données de niveau utilisateur peuvent être stockées et mises à jour.

Les noeuds limites sont gérés sous forme de hiérarchie arborescente, l'administrateur central étant la racine, et d'autres utilisateurs étant les chefs de groupes plus restreints. L'administrateur central est le superutilisateur (ou utilisateur racine) du système.

Les erreurs relatives aux noeuds limites, dont les orphelins et les boucles de groupe, sont abordés au Chapitre 9.

---

## Administration déléguée

L'administrateur central est le principal responsable de la gestion des noeuds limites. Solaris Resource Manager fait appel à plusieurs contrôles de ressource pouvant être assignés et gérés, et permet de sélectionner certains privilèges d'administration à assigner à des utilisateurs autres que l'administrateur, afin de répartir les tâches

d'administration des utilisateurs. Des privilèges d'administration peuvent être assignés en définissant l'indicateur *uselimadm* ou *admin* des utilisateurs voulus. Les utilisateurs dont l'indicateur *uselimadm* est activé jouissent des mêmes privilèges que le superutilisateur dans le programme `limadm(1MSRM)`. Un chef de groupe dont l'indicateur *admin* est activé est appelé un sous-administrateur et détient des privilèges (voir ci-dessous) relatifs aux utilisateurs au sein de son groupe d'ordonnement.

L'administrateur central contrôle la division globale des ressources système en créant et en assignant des limites aux groupes d'ordonnement dont la racine est mère. Les sous-administrateurs effectuent habituellement les mêmes types de contrôle des ressources, mais ils se limitent aux utilisateurs de leur groupe d'ordonnement. La division des ressources par le sous-administrateur se limite aux ressources attribuées au groupe (par exemple, les ressources allouées au noeud limite du chef de groupe). Notez qu'un sous-administrateur peut assigner un indicateur administratif à tout utilisateur de son groupe d'ordonnement, subdivisant ainsi ses propres charges administratives.

Un sous-administrateur peut effectuer les tâches suivantes :

1. Créer et supprimer des noeuds limites pour les utilisateurs de son groupe d'ordonnement.
2. Modifier les limites de ressource de tout utilisateur au sein de son groupe d'ordonnement.

Notez que même si un sous-administrateur peut fixer la limite d'une ressource à un niveau supérieur à la limite établie pour le groupe, les ressources consommées par les membres du groupe sont également imputées aux chefs de groupe, et les limites individuelles seront appliquées si un dépassement de la limite du chef de groupe est tenté.

3. Modifier les indicateurs de tout noeud limite dans son groupe d'ordonnement, à condition que l'indicateur ne soit pas assorti de la condition `noadmin`. Les affectations d'indicateur effectuées par les sous-administrateurs sont aussi limitées par le fait qu'un utilisateur ne peut se voir accorder un privilège qui n'est pas déjà attribué au sous-administrateur. Cette restriction vise à empêcher un sous-administrateur de contourner les mesures de sécurité dans Solaris Resource Manager.
4. Configurer n'importe lequel de ses propres attributs dotés de la condition `selfadmin`.

Les principaux outils du sous-administrateur sont les commandes `limadm(1MSRM)` et `limreport(1SRM)`. Le programme `limadm` exécute des opérations sur les limites, les indicateurs et d'autres attributs de Solaris Resource Manager pour un ou plusieurs utilisateurs. Combinés au générateur de rapports `limreport`, ces outils permettent à un groupe d'ordonnement de s'autogérer sans perturber l'affectation des ressources ni la gestion d'autres groupes d'ordonnement.



Le superutilisateur est exempté de toute limite de ressource, dispose toujours de privilèges d'administration complets sans égard au réglage de ses indicateurs. Il peut ajouter, supprimer et modifier les comptes utilisateurs, et peut modifier tout usage, limite ou indicateur de tout noeud limite utilisant le programme `limadm`.

## Sécurité

Solaris Resource Manager ayant d'importantes répercussions sur l'administration d'un système Solaris, son installation et sa maintenance doivent être effectuées de façon à assurer la sécurité du système.

L'administrateur dispose de plusieurs méthodes pour assurer la sécurité d'un système Solaris Resource Manager. Comme pour tout système Solaris, la plus importante consiste à assurer la confidentialité du mot de passe racine. Toute personne connaissant ce mot de passe jouit d'un accès illimité aux ressources système, tout comme l'administrateur central.

Des privilèges administratifs spéciaux peuvent être accordés à des utilisateurs dans Solaris Resource Manager par l'intermédiaire de certains indicateurs système au sein de leur noeud limite. Ces privilèges peuvent contribuer à accroître la sécurité du système en permettant de confier des tâches à des utilisateurs sans avoir à leur accorder des privilèges complets de superutilisateur.

Néanmoins, certains de ces privilèges doivent être accordés avec précaution car ils fournissent un vaste éventail de pouvoirs. Le mot de passe des utilisateurs disposant de privilèges spéciaux doit être protégé avec autant de soin que celui du superutilisateur.

Dans certaines circonstances, l'administrateur central risque d'augmenter la vulnérabilité du système s'il néglige l'importance devant être accordée à la manipulation de la structure de l'arbre d'ordonnancement. Il est essentiel que l'administrateur central comprenne comment modifier l'arbre d'ordonnancement correctement et y repérer les problèmes potentiels.

## Indicateurs *uselimadm* et *admin*

L'administrateur central peut attribuer des privilèges administratifs dans Solaris Resource Manager par l'intermédiaire des indicateurs *uselimadm* et *admin* du noeud limite d'un utilisateur. L'indicateur *uselimadm* de la commande `limadm(1MSRM)` accorde les mêmes privilèges que ceux de l'administrateur central. Lorsqu'il est défini pour un chef de groupe, l'indicateur *admin* lui fournit des privilèges administratifs sur les membres de son groupe mais ne lui permet pas de modifier le contenu des noeuds limites extérieurs à son groupe.

Les chefs de groupe dont l'indicateur *admin* est activé sont appelés des sous-administrateurs. Plusieurs mesures de sécurité doivent être prises dans Solaris

Resource Manager afin d'empêcher le mauvais usage des privilèges administratifs accordés aux sous-administrateurs. Pour en savoir davantage, reportez-vous à "Serveur d'applications type" , page 27 et "Programmes de maintenance des noeuds limites" , page 59.

Lorsqu'il supprime des noeuds limites, un sous-administrateur doit s'assurer que les sous-arbres sont supprimés à partir du dernier noeud limite en remontant. Si vous commencez au sommet du sous-arbre à supprimer, vous perdrez le contrôle des enfants des noeuds limites qui ont été supprimés, car ils deviendront orphelins lorsque leurs parents seront effacés. Une fois des noeuds limites devenus orphelins, le sous-administrateur ne peut les modifier, car ils se trouvent à l'extérieur du groupe d'ordonnement.

## Structure suggérée de noeud limite sous-administrateur

Un des problèmes auxquels les sous-administrateurs peuvent être confrontés est qu'ils partagent les mêmes limites de groupe que les membres. Par exemple, si une limite de processus est définie sur le noeud limite du chef de groupe, elle fixe le nombre de processus pouvant être utilisés par l'ensemble du groupe, incluant le chef de groupe. À moins de définir des limites additionnelles, tout utilisateur du groupe d'ordonnement qui excède sa limite de processus empêchera le sous-administrateur de créer de nouveaux processus. Pour éviter cela, l'administrateur doit définir des limites individuelles pour chaque membre du groupe. Pour que ces limites soient efficaces, il peut être nécessaire de les définir avec des contraintes excessives. De plus, le fait de forcer un sous-administrateur à gérer des limites individuelles va à l'encontre du but de Solaris Resource Manager qui vise à contrôler les ressources de manière hiérarchique. L'administrateur peut contourner ce problème en modifiant la structure des noeuds limites au sein de son groupe. Au lieu de placer les utilisateurs directement sous leur propre noeud limite, il peut créer un noeud limite "contrôle" sous le sien en tant que seul noeud limite enfant, puis rendre les utilisateurs enfants du noeud Contrôle, comme dans l'arbre ci-dessous.

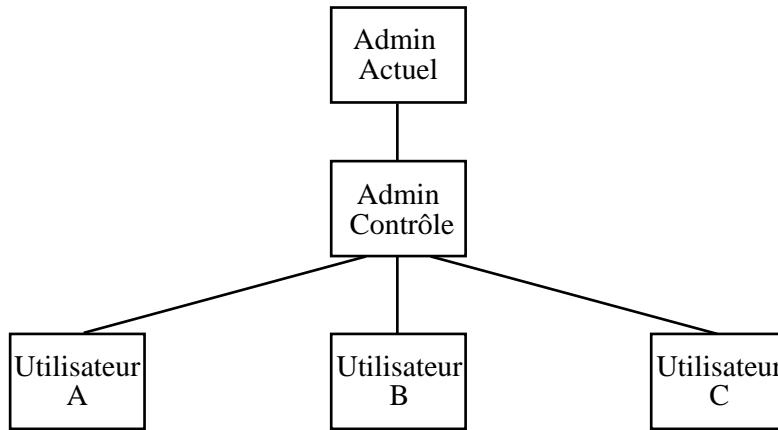


Figure 5-1 Structure de nœud limite de sous-administrateur

Dans l'exemple ci-dessus, l'UID du compte du sous-administrateur correspond à celle du nœud limite "Réel", père de l'arbre. Il s'agit du nœud limite dont l'indicateur *admin* serait activé. Un compte fictif serait créé pour le nœud limite "Contrôle". Il n'est pas nécessaire d'autoriser une ouverture de session sur ce compte. Les nœuds limites "A", "B" et "C" correspondent aux utilisateurs placés sous le contrôle du sous-administrateur.

Dans cet exemple, la limite de processus du nœud "Réel" pourrait être de 100, tandis que celle du nœud "Contrôle" pourrait être de 90, et les limites des utilisateurs individuels sont de 0. Avec cette configuration, même si les utilisateurs A, B et C utilisaient un total de 90 processus (maximum autorisé), le sous-administrateur pourrait quand même créer 10 processus. Ici, des utilisateurs peuvent quand même empêcher les autres de créer des processus. La seule façon de l'éviter est de fixer des limites individuelles pour les utilisateurs visés. Toujours dans notre exemple, ces limites pourraient être de 40 pour chaque utilisateur, ce qui assurerait une certaine souplesse tout en empêchant un seul utilisateur de priver les autres de ressources. Notez également que le sous-administrateur peut créer des nœuds limites additionnels pour des nouveaux utilisateurs en tant qu'enfants du nœud "Contrôle" et ce, sans avoir à rééquilibrer les limites.

---

## Base de données des limites

La base de données des limites contient les informations sur les utilisateurs et permet à Solaris Resource Manager d'effectuer le contrôle des ressources. Elle contient un nœud limite par UID, auquel on accède en utilisant l'UID comme index direct dans le fichier. S'il existe un nœud limite pour une UID numériquement élevée, la base de données des limites semblera grande. Cependant, si les UID des utilisateurs ne sont pas successives, la base de données des limites comportera de grands trous et, sur un

système de fichiers le permettant, elle pourrait être stockée sous forme de fichier réparti. Autrement dit, aucun bloc de disque ne sera assigné au stockage des sections "vides" du fichier. *Le système de fichiers ufs* accepte les fichiers répartis, mais pas le système de fichiers *tmpfs*. Pour en savoir davantage sur l'incidence des fichiers répartis sur l'enregistrement et la restauration de la base de données des limites, reportez-vous à la section "Enregistrement et restauration de la base de données des limites" , page 56.

Chaque fois qu'un utilisateur est créé, vous devez créer un noeud limite.

## Création de la base de données des limites

Le fichier de démarrage de Solaris Resource Manager (`/etc/init.d/init.srm`) génère une base de données des limites initiale à sa première exécution ou chaque fois qu'il est manquant.

La base de données des limites réside habituellement dans le répertoire `/var/srm`.

La base de données et son groupe doivent appartenir au superutilisateur, qui doit être le seul autorisé à la consulter. Aucune permission en écriture n'est requise puisque que seul du code noyau avec authentification de superutilisateur permet d'y écrire.



---

**Attention** - La sécurité du système pourrait être compromise si un utilisateur pouvait écrire dans la base de données des limites Solaris Resource Manager.

---

## Enregistrement et restauration de la base de données des limites

La base de données des limites pouvant être un fichier réparti, sa copie doit faire l'objet de précautions particulières. Le fichier occupera probablement beaucoup d'espace disque s'il est enregistré par un utilitaire ne prenant pas en charge les fichiers répartis, car les sections vides seront lues comme une suite de zéros et inscrites sous forme de vrais blocs de données au lieu de sections vides. Cela peut se produire si le fichier est copié, sauvegardé ou restauré par un utilitaire tel que `tar(1)`, `cpio(1)` ou `cp(1)` ; cependant, des programmes comme `ufsdump(1M)` et `ufsrestore(1M)` préserveront les trous.

La sauvegarde et la restauration de la base de données des limites peuvent également être effectuées à l'aide de `limreport(1SRM)` pour générer une version ASCII du fichier et de `limadm(1MSRM)` pour restaurer l'original depuis la version ASCII. Par exemple, la commande :

```
limreport 'flag.real' - lname preserve > /var/tmp/savelnodes
```

créera le fichier `/var/tmp/savelnodes`, qui contient une représentation ASCII des noeuds limites de chaque utilisateur dans la table des mots de passe. Notez que cette opération ne sauvegardera pas les noeuds limites n'ayant pas d'entrée correspondante dans la table des mots de passe. En règle générale, il ne devrait pas y avoir plus de noeuds limites que le nombre total d'UID dans la table des mots de passe.

La commande :

```
# limadm set -f - < /var/tmp/savelnodes
```

recréera les noeuds limites dont les données ont été sauvegardés. Cette commande ne supprimant pas les noeuds qui n'ont pas été sauvegardés, ces techniques peuvent également servir à sauvegarder et à restaurer des noeuds sélectionnés plutôt que l'ensemble de la base de données des limites.

“Commandes `limreport` et `limadm`”, page 62 explique l'utilisation des commandes `limreport(1SRM)` et `limadm(1MSRM)` plus en détail. Il est utile que l'administrateur se familiarise avec ces commandes pour sauvegarder et restaurer des noeuds limites, car elles peuvent servir lorsqu'une modification est apportée à l'interprétation de l'arbre des noeuds limites (telle que définie par le fichier de configuration).

Étant donné que le contenu de la base de données change régulièrement lors des opérations habituelles du système, il est recommandé d'effectuer les sauvegardes pendant que le système est au repos ou en mode mono-utilisateur. De même, la restauration de l'ensemble de la base de données des limites devrait être faite uniquement lorsque Solaris Resource Manager n'est pas utilisé, par exemple, en mode mono-utilisateur.

---

## Création et suppression de noeuds limites

Chaque fois qu'un utilisateur est créé, un noeud limite correspondant doit être créé, et ses limites et privilèges définis. L'utilisateur ne peut ouvrir une session tant que son noeud limite n'est pas créé. Lorsqu'il utilise Solaris Resource Manager, l'administrateur doit assurer la maintenance de la base de données en parallèle avec la base de données de mots de passe Solaris habituelle. La commande :

```
# limreport \ !flag.real - uid lname
```

peut être utilisée pour imprimer la liste des UID et des noms de compte des utilisateurs n'ayant pas de noeud limite correspondant.

Dans cette version, les noeuds limites ne sont pas créés et supprimés automatiquement par les commandes système utilisées pour créer et supprimer des comptes. Ces opérations doivent être effectuées par l'administrateur. Cependant, il est possible de créer des noeuds limites sur demande. Pour plus de détails, voir "Sous-système PAM", page 40.

De même, juste avant de supprimer un compte dans la table des mots de passe, le noeud limite correspondant doit être effacé de la base de données des limites avec la commande `limadm(1MSRM)`.

Si une UID est changée, le contenu de son noeud limite doit être copié vers le nouveau noeud correspondant au nouveau numéro, puis le noeud d'origine doit être supprimé. Reportez-vous à la section "Copie et déplacement de noeuds limites", page 63.

Les noeuds enfants doivent être reliés au nouveau noeud ou à un autre noeud père approprié. La commande :

```
# limreport 'sgroup==X' '%u\tsgroup=Y\n' uid | limadm set -u -f -
```

permet de détecter tous les noeuds limites ayant un groupe d'ordonnement père dont l'UID est X et d'en faire les enfants du noeud ayant l'UID Y.

Les étapes ci-après illustrent le changement d'UID d'un noeud limite de X à Y.

1. Enregistrement de l'état du noeud limite dont l'UID doit être changée :

```
# limreport 'iud==X' - lname preserve > /var/tmp/savelnodes.X
```

2. Remplacement de l'ancienne UID (X) correspondant à cet utilisateur par la nouvelle UID (Y) dans la table des mots de passe.
3. Création d'un noeud limite pour la nouvelle UID et restauration de l'état enregistré au préalable :

```
# limadm set -f /var/tmp/savelnode.X
```

4. Remplacement du groupe d'ordonnement de tous les noeuds limites enfants du noeud à modifier (UID X) par le nouveau noeud limite (UID Y) :

```
# limreport 'sgroup==X' '%u\tsgroup=Y\n' uid | limadm set -u -f -
```

5. Assurez-vous qu'aucun processus n'est relié à l'ancien noeud limite. Voir "Création et suppression de noeuds limites", page 57.
6. Utilisez la commande `chown(2)` pour remplacer le propriétaire de tous les fichiers de l'UID d'origine par celui de la nouvelle UID. Par exemple :

```
# find / -user X -print | xargs chown Y
```

7. Supprimez l'ancien noeud limite :

```
# limadm delete X
```

## Programmes de maintenance des noeuds limites

La commande `limadm` est le principal outil de maintenance des noeuds limites. Elle permet de changer les valeurs d'attribut de Solaris Resource Manager pour une liste de comptes utilisateur. Si aucun noeud limite n'existe pour un utilisateur, un noeud vide est d'abord créé par défaut. Voici les propriétés des nouveaux noeuds limites :

- *flag.real* est activé ;
- les attributs *cpu.shares* et *cpu.myshares* sont fixés à 1 ;
- les indicateurs *uselimadm* et *admin* sont fixés à "clear" ;
- tous les autres indicateurs sont fixés à "inherit" ;
- tous les attributs de limite et d'usage sont fixés à zéro.

Le groupe d'ordonnement du nouveau noeud limite est réglé à celui de l'appelant de `limadm` s'il s'agit d'un sous-administrateur ; sinon (si l'appelant est `root` ou que son indicateur *uselimadm* est défini), il est réglé à l'utilisateur Autre si un noeud limite existe pour ce compte d'utilisateur, ou au noeud limite racine si ce n'est pas le cas.

L'appelant de `limadm` doit disposer de privilèges d'administration suffisants pour effectuer les changements indiqués. Il doit s'agir du superutilisateur, avoir un indicateur *uselimadm* activé, ou s'agir d'un sous-administrateur modifiant uniquement les attributs des membres de son groupe d'ordonnement. Certaines restrictions s'appliquent à l'utilisation de `limadm` par un sous-administrateur : —

- il ne peut changer la valeur de ses propres attributs, à l'exception de ceux dotés de la condition *selfadmin* (par exemple, l'attribut *cpu.myshares*) ;
- l'attribut *sgroup* peut être affecté uniquement à un chef de groupe membre du groupe d'ordonnement de l'appelant ou à l'appelant même ;

- il ne peut changer les attributs des utilisateurs d'autres groupes d'ordonnement ;
- il ne peut modifier la valeur de tout attribut utilisé pour stocker les usages. Si cette restriction n'était pas établie, le sous-administrateur pourrait contourner les limites de groupe dans son propre noeud limite en diminuant l'usage de l'un de ses enfants, réduisant ainsi l'usage du groupe ;
- si la valeur d'un indicateur est contraire à sa valeur par défaut, il ne peut modifier cette valeur pour un membre du groupe, sauf pour la fixer à la même valeur contraire. Cette mesure assure que les sous-administrateurs dont des privilèges ont été explicitement interdits ne peuvent accorder ces privilèges à des utilisateurs sous leur contrôle.

La commande `limadm` permet à un administrateur de supprimer un noeud limite sans effacer le compte correspondant dans la table des mots de passe. Pour pouvoir utiliser `limadm`, l'appelant doit être le superutilisateur, ou avoir un indicateur `uselimadm` ou `admin` activé. Si seul l'indicateur `admin` de l'appelant est activé, il peut supprimer uniquement les noeuds limites des utilisateurs dont il est le chef de groupe.

---

## Unités

Dans Solaris Resource Manager, les valeurs sont représentées par l'un des trois types d'unités suivants :

### Proportionnée

L'unité proportionnée est le format par défaut, facilement lisible, qui permet d'afficher et d'entrer des valeurs. Elle contribue à éviter les erreurs des utilisateurs en réduisant le nombre de chiffres à entrer. —

### Brute (ou non proportionnée)

Les unités brutes sont les unités élémentaires de représentation d'une valeur. Par exemple, les unités brutes pour l'usage de la mémoire virtuelle sont des octets, tandis que les unités brutes pour le débit de la mémoire virtuelle sont des octets par seconde. Ces dernières sont surtout utilisées pour la facturation de l'usage, lorsque des quantités exactes sont nécessaires. —

### Interne

Les unités internes sont utilisées par Solaris Resource Manager pour stocker des attributs de mémoire en unités machine plutôt qu'en octets.



---

## Conversions

Les programmes Solaris Resource Manager convertissent les unités internes utilisées pour stocker des valeurs d'attribut, ce qui permet de toujours présenter des unités proportionnelles ou brutes à l'utilisateur. Sauf quelques exceptions, cela signifie que l'utilisateur n'a pas à se préoccuper des unités utilisées par Solaris Resource Manager.

Dans Solaris Resource Manager, les expressions exa, péta, téra, giga, méga et kilo sont utilisées pour représenter les puissances de deux et non de dix. Par exemple, un méga-octet compte 1 048 576 octets et non 1 000 000. La puissance de deux de chaque expression est 60 (exa), 50 (péta), 40 (téra), 30 (giga), 20 (méga) et 10 (kilo).

Les programmes constituant l'interface utilisateur principale du système Solaris Resource Manager sont `limadm(1MSRM)`, `liminfo(1SRM)` et `limreport(1SRM)`. Les conversions et mises à l'échelle effectuées sont expliquées en détail dans les sous-sections suivantes.

### Commande `limadm`

Lors d'un changement de valeurs d'attribut, `limadm` permet d'ajouter des caractères aux nombres : `[EPTGMK][B][.][wdhms]`. Les majuscules et les minuscules sont interchangeables. Si l'attribut exprime une quantité de stockage (attributs de mémoire) ou un débit de stockage, un caractère du premier groupe (EPTGMK) est permis. Cela est multiplié par le nombre d'octets compris dans un exa-octet (E), un péta-octet (P), un téra-octet (T), un giga-octet (G), un méga-octet (M) ou un kilo-octet (K). Le caractère B facultatif peut être ajouté pour faciliter la lecture, mais n'a aucun effet. Si l'attribut exprime le temps (date ou heure) ou le débit de stockage, un caractère du deuxième groupe est permis. Cela est multiplié par le nombre de secondes que compte une semaine (w), un jour (d), une heure (h), une minute (m) ou par le nombre de secondes (s), selon le cas. Un point facultatif peut séparer les unités de stockage et de temps (par exemple, mh, M.h et MB.h expriment tous des "méga-octet par heure"). Si le suffixe M porte à confusion, `limadm` tente de déduire sa signification selon le contexte. Si cela est impossible, la valeur méga est interprétée par défaut de préférence aux minutes.

Ces caractères de conversion sont utiles pour éviter les erreurs d'ordre de grandeur en entrant des nombres élevés, mais la quantité est stockée en unités internes, peu importe la méthode d'entrée utilisée.

Le caractère spécial `u` peut aussi être utilisé seul mais uniquement pour les valeurs d'attribut de mémoire. Il indique que le nombre est en unités machine (internes) et non en octets.

## Commande `liminfo`

La commande `liminfo(1SRM)` utilise les mêmes suffixes pour les rapports que `limadm(1MSRM)` emploie pour l'entrée (voir ci-dessus). Normalement, `liminfo` convertit les valeurs en formats proportionnels appropriés pour l'impression, mais l'option `-r` peut être utilisée pour forcer `liminfo` à imprimer les valeurs brutes. Par exemple, la mémoire est habituellement exprimée à l'aide d'une unité proportionnelle pertinente comme le méga-octet (p. ex. 102 Mo), mais l'option `-r` forcera l'impression en octets (p. ex. 106 954 752 octets).

## Commande `limreport`

La commande `limreport(1SRM)` indique toujours les valeurs sous forme brute. Si des valeurs proportionnelles sont requises, la conversion doit être énoncée explicitement dans l'expression utilisée pour afficher la valeur. Par exemple, pour afficher l'usage total de mémoire virtuelle pour tous les utilisateurs en kilo-octets arrondis au ko près :

```
# limreport 'flag.real' '%-8s %d KB\n' lname '(memory.usage+1k-1)/1k'
```

Comme on le voit dans cet exemple, il est possible d'utiliser des suffixes proportionnels pour les nombres dans des expressions, ce qui simplifie la conversion des unités brutes en valeurs proportionnelles.

Notez que les unités internes de certains attributs diffèrent de leur forme brute. Normalement, l'utilisateur n'a pas à s'en préoccuper, car tous les programmes Solaris Resource Manager convertissent en unités proportionnelles ou brutes mais, à titre d'exemple, certaines expressions de `limreport` qui précisent une correspondance exacte sur un nombre d'octets échoueront toujours si un nombre indiqué n'est pas un multiple intégral de l'unité interne pertinente.

---

# Manipulation de noeuds limites

## Commandes `limreport` et `limadm`

Les commandes `limreport(1SRM)` et `limadm(1MSRM)` fournissent à l'administrateur un moyen très simple d'enregistrer et de restaurer le contenu des noeuds limites de n'importe quel nombre d'utilisateurs. La commande `limreport` permet de sélectionner et d'extraire les noeuds limites à enregistrer, tandis que `limadm` permet de les restaurer. Les usages les plus courants pour cette combinaison de commandes sont la copie de noeuds limites et la modification de la structure des noeuds limites.

La commande `limreport` est un outil polyvalent pour sélectionner et afficher les attributs des utilisateurs. Elle fournit deux niveaux de sélection : la sélection de noeuds limites et la sélection d'attributs à afficher pour chaque noeud limite sélectionné. La sélection des noeuds limites s'effectue au moyen d'une expression de sélection, pouvant être une condition simple ou un ensemble de conditions reliées par des opérateurs logiques avec une syntaxe de type C. La sélection des attributs s'effectue en listant les noms symboliques des attributs voulus. Le mode d'affichage des attributs peut être précisé par une chaîne de commande de format similaire à la fonction `limreport` de C, avec des extensions pour traiter les types spéciaux de Solaris Resource Manager. Si une chaîne de commande de format '-' est précisée, `limreport` utilise les formats par défaut pour chaque attribut affiché. Pour en savoir davantage, voir `limreport(1SRM)`.

La commande `limadm` permet de changer le contenu d'attributs dans un noeud limite si le demandeur détient les privilèges requis. Les commandes de modification peuvent être précisées directement sur la ligne de commande ou en spécifiant le nom du fichier qui les contient (à l'aide de l'option `-f`).

`limreport` permet de générer des attributions de valeur d'attribut en utilisant la syntaxe `lim` (reportez-vous à l'identificateur "preserve" de la syntaxe `lim`), dont la sortie peut être entrée dans `limreport` à l'aide de l'option `-f`. Cette méthode permet à l'administrateur d'utiliser les deux programmes ensemble pour sélectionner le contenu à sauvegarder ou à restaurer dans la base de données des limites.

## Copie et déplacement de noeuds limites

La commande :

```
# limreport 'uid==X' - Y preserve | limadm set -u -f -
```

copie un noeud limite de l'UID X à l'UID Y. L'expression `'uid==X'` permet de sélectionner le noeud limite source. L'identificateur "preserve" force `limreport` à extraire toutes les valeurs d'attribut qui ne sont pas en lecture seule dans une syntaxe pouvant être acheminée à `limadm`. L'UID Y étant placée devant l'identificateur "preserve", il s'agit du premier élément de données transféré à `lim`, ce qui fournit la sélection du noeud limite cible.

Si le noeud limite source n'est plus nécessaire, il peut être enlevé avec `limadm`.

---

**Remarque** - La prudence est de mise si vous utilisez une concordance par UID en tant qu'expression de sélection `limreport`. Si plusieurs comptes partagent une UID, ils seront tous associés. Dans l'exemple ci-dessus, la même donnée de noeud limite sera préservée et chargée plusieurs fois. Dans le système Solaris, l'UID détient les comptes `root` et `smtp`.

---



## Ordonnanceur SHR

---

L'ordonnanceur SHR de Solaris Resource Manager est utilisé pour contrôler l'attribution des ressources de l'unité centrale. Le concept de parts permet aux administrateurs de contrôler facilement des droits relatifs aux ressources de l'UC pour les utilisateurs, les groupes et les applications. Ce concept est similaire à celui des actions d'une entreprise ; ce qui importe n'est pas la quantité détenue mais la proportion d'actions par rapport aux autres actionnaires.

---

### Description technique

Il existe quatre attributs par noeud limite associé à l'ordonnanceur de Solaris Resource Manager : *cpu.shares*, *cpu.myshares*, *cpu.usage* et *cpu.accrue*. La sortie de `liminfo(1SRM)` affiche ces attributs ainsi que d'autres valeurs utiles.

Sous Solaris Resource Manager, l'ordonnement est mis en oeuvre au moyen de la classe d'ordonnement SHR, qui comprend la prise en charge des commandes `dispadm(1M)`, `priocntl(1)`, `nice(1)` et `renice(1)`. Au niveau de l'appel système, SHR est compatible avec la classe d'ordonnement TS.

### Parts

L'attribut *cpu.shares* d'un utilisateur permet de répartir les droits à l'UC en fonction du père et des homologues actifs de l'utilisateur. L'attribut *cpu.myshares* est significatif uniquement si l'utilisateur a des utilisateurs enfants actifs ; il permet de déterminer la proportion de droits à l'UC en fonction de ces derniers.

Par exemple, si les utilisateurs A et B sont les seuls enfants du père P et que A, B et P ont chacun une part au sein du groupe P ( *cpu.shares* est fixé à 1 pour A et B, et

*cpu.myshares* est fixé à 1 pour P), chacun à droit à un tiers du total des ressources de l'UC accordées au groupe.

Les droits à l'UC d'un utilisateur sont donc tributaires des droits relatifs du père, ce qui dépend des valeurs relatives de *cpu.shares* du père sur ses homologues et de *cpu.myshares* du grand-père, et ainsi de suite en remontant l'arbre.

Aux fins de la gestion du système, les processus liés au noeud limite racine ne sont pas concernés par les attributs des parts. Tout processus lié au noeud limite racine obtient toujours la plupart des ressources de l'UC qu'il demande.

Il est important que les processus qui consomment beaucoup de ressources d'UC ne soient pas liés au noeud limite racine, à défaut de quoi l'exécution d'autres processus serait grandement ralentie. Pour l'éviter, les précautions ci-après doivent être prises :

- Le compte de l'administrateur central doit posséder sa propre UID, qui doit différer de celle du superutilisateur. Ce compte doit être utilisé pour ouvrir une session en vue d'effectuer des tâches non administratives. Si l'UID d'un superutilisateur est requise pour exécuter des fonctions administratives, l'administrateur central peut se servir de la commande `su(1)` afin de changer d'UID tout en demeurant relié à son propre noeud limite.
- La commande `srmuser(1SRM)` peut être utilisée dans les scripts `init.d(4)` pour relier tout processus démon à un noeud limite non racine. Tout processus lancé depuis le script de démarrage détient par défaut une UID racine et est relié au noeud limite racine. La commande "user" permet aux démons de conserver une UID racine tout en demeurant reliés à leur propre noeud limite, ce qui évite les problèmes si un démon accroît sa consommation de ressources de l'UC.

Les chefs de groupe de l'arbre d'ordonnancement n'ont pas tous à représenter les utilisateurs qui exécutent actuellement des processus, et il n'est pas nécessaire de leur allouer une part des ressources de l'UC. Ces noeuds limites peuvent être indiqués en fixant leur attribut *cpu.myshares* à zéro. L'attribut cumulatif d'UC dans un tel chef de groupe comprend néanmoins la charge totale pour tous les membres de son groupe.

## Part attribuée

Les attributs *cpu.shares* et *cpu.myshares* déterminent la part d'UC attribuée à chaque noeud limite actif en pourcentage. Les parts des utilisateurs inactifs n'ont aucune incidence sur la part attribuée. Si un seul utilisateur est actif, il disposera de 100 % des ressources de l'UC. Si seulement deux utilisateurs sont actifs et disposent de parts égales dans le même groupe, chacun recevra une part de 50 %. Pour en savoir davantage sur le calcul de la part attribuée, consultez la section "Calcul de la part attribuée", page 69.

## Usage et décroissance

La valeur de l'attribut *cpu.usage* augmente chaque fois qu'un processus relié au noeud limite est chargé pour un temps d'UC. Cette valeur décroît exponentiellement à un taux déterminé par le paramètre global de décroissance d'usage de Solaris Resource Manager. Le taux de décroissance (exprimé par une demi-vie en secondes) est défini par la commande `srmadm(1MSRM)`.

Bien que tous les processus soient dotés d'un noeud limite sans égard à leur classe d'ordonnancement actuelle, ceux qui se trouvent à l'extérieur de la classe d'ordonnancement SHR ne sont jamais chargés.

## Usage cumulatif

La valeur de l'attribut d'usage cumulatif augmente dans les mêmes proportions que celle de l'attribut d'usage, mais elle ne décroît pas. Elle représente donc l'usage cumulatif total pour tous les processus ayant été reliés au noeud limite et à ses membres depuis la dernière réinitialisation de l'attribut.

## Part effective

La part effective courante d'un noeud limite est déterminée par sa part attribuée et par l'attribut *cpu.usage*. L'ordonnanceur de Solaris Resource Manager ajuste les priorités de tous les processus reliés à un noeud limite de façon à rendre leur taux de traitement proportionnel à la part effective du noeud et inversement proportionnel au nombre de processus exécutables qui y sont reliés.

## Priorité de partage par processus (*sharepri*)

Chaque processus relié à un noeud limite intègre des données propres à Solaris Resource Manager et dont la maintenance est assurée par le noyau du système d'exploitation. Aux fins de l'ordonnancement, la plus importante de ces valeurs est *sharepri*. Les processus dont la valeur "sharepri" est la plus basse ont en tout temps priorité pour leur exécution par l'UC.

---

# Exemple d'attribution de parts

## Structure de l'arbre d'ordonnancement

Les paragraphes ci-après traitent de la structure de l'arbre d'ordonnancement, sujet auquel l'administrateur central doit porter une attention particulière :

- L'arbre d'ordonnancement est la structure utilisée par Solaris Resource Manager pour mettre en oeuvre une hiérarchie de contrôle des ressources et des privilèges. Si un sous-administrateur obtient le contrôle d'un sous-arbre de l'arbre d'ordonnancement auquel il n'a normalement pas accès, il peut accéder à des ressources et privilèges additionnels sans l'approbation de l'administrateur central. Cela peut se produire lorsque l'administrateur supprime un noeud limite en laissant un sous-arbre orphelin. Un sous-administrateur pourrait recréer le noeud manquant dans son propre groupe, obtenant ainsi le contrôle de la section orpheline de l'arbre.
- L'administrateur central peut exécuter la commande `limreport(1SRM)` pour repérer les sections orphelines de l'arbre par l'intermédiaire de l'identificateur d'orphelin intégré. Tout orphelin détecté doit être relié immédiatement. Cependant, la commande `limreport` peut également être utilisée par les sous-administrateurs pour repérer les sous-arbres orphelins et les ajouter à son groupe, tel qu'expliqué dans le paragraphe précédent, ce qui ouvre la voie à des violations potentielles de la sécurité.
- Lorsqu'un noeud limite est créé, il est constitué en grande partie de zéros, forçant ainsi la plupart des indicateurs à adopter la valeur par défaut "inherit" (héritage). Il s'agit du résultat souhaité pour la plupart des indicateurs, puisqu'ils sont utilisés pour indiquer les privilèges de périphérique. Les indicateurs mis à zéro de façon explicite lors de la création d'un noeud limite sont `uselimadm` et `admin`. Cette mesure vise à empêcher que les nouveaux utilisateurs obtiennent automatiquement des privilèges administratifs.

## Description de l'arbre

L'arbre ci-dessous illustre une structure comportant plusieurs chefs de groupe et utilisateurs. La racine de l'arbre est l'utilisateur racine. Un noeud limite de chef de groupe est indiqué par deux nombres entiers qui représentent les valeurs de ses attributs "shares" et "myshares" respectivement. Un noeud limite feuille est indiqué par un seul nombre entier, la valeur de son attribut "shares".



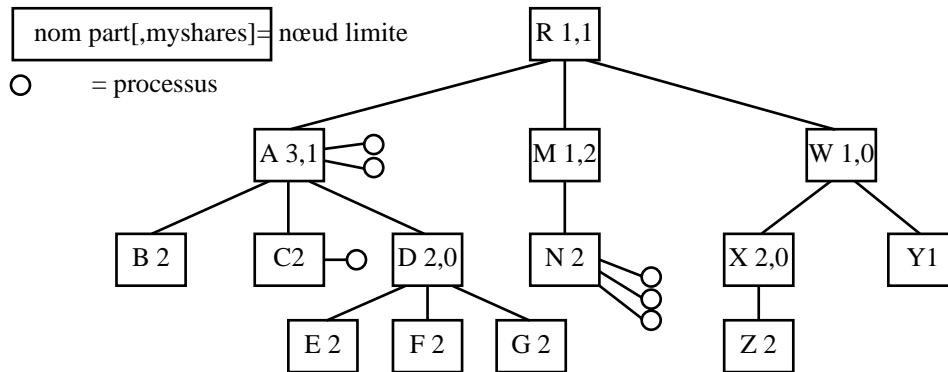


Figure 6-1 Structure de l'arbre d'ordonnancement

## Calcul de la part attribuée

Des processus sont actuellement reliés aux noeuds limites A, C et N. Au niveau supérieur, il suffirait de partager l'unité centrale entre A et M du fait qu'il n'y a aucun processus pour W ou un membre de ce groupe. Le rapport des parts entre A et M étant de 3:1, la part attribuée au niveau supérieur serait de 75 % au groupe A et 25 % au groupe M.

La part de 75 % accordée au groupe A serait partagée entre les utilisateurs actifs (A et C) en fonction du rapport de leurs parts au sein du groupe A (1/2). Notez que l'attribut "myshares" est utilisé pour déterminer les parts de A par rapport à ses enfants. Par conséquent, A obtiendrait un tiers de la part attribuée au groupe, et C les deux tiers. Le total attribué au groupe M serait accordé au noeud N, car il est le seul à comporter des processus.

La distribution globale des parts d'UC disponibles serait donc de 0,25 pour A, 0,5 pour C et 0,25 pour N.

Maintenant, supposons que les processus de A, C et N ont toujours besoin de l'UC et que le système dispose d'un maximum de deux processeurs. Dans un tel cas, Solaris Resource Manager ferait en sorte que chaque processus reçoive les pourcentages ci-après du total des ressources de l'UC : —

- pour les deux processus de A : 12,5 % chacun
- pour le processus de C : 50 %
- pour les trois processus de N : 8,3 % chacun

Le taux de progression de chaque processus est contrôlé de façon à ce que la cible soit atteinte pour chaque noeud limite. Sur un système doté de plus de deux unités centrales mais seulement de ces six processus exécutables, le processus de C ne parviendra pas à consommer sa part de 50 %, et le reste sera réparti proportionnellement entre A et N.

---

# Relation entre Solaris Resource Manager et la fonction `nice` de Solaris

La fonction `nice` de Solaris permet à un utilisateur de réduire la priorité d'un processus pour que les processus habituels ne soient pas ralentis par ceux qui ne sont pas urgents. Pour les utilisateurs de Solaris Resource Manager, cette fonction présente l'avantage de réduire la charge de travail pour le temps de traitement utilisé en basse priorité.

Solaris Resource Manager met en oeuvre cette fonctionnalité en permettant à l'administrateur central de modifier le taux de décroissance "sharepri" pour les processus visés par "nice". Le paramètre Solaris Resource Manager global de la commande `srmadm(1MSRM)` est utilisé pour définir les taux de décroissance applicables aux priorités des processus ayant des valeurs "nice" normale et maximum. Les taux de toutes les valeurs "nice" intermédiaires sont interpolés puis extrapolés à la valeur `nice` minimale. Par exemple, la priorité (`sharepri`) des processus normaux peut décroître avec une demi-vie de deux secondes, tandis que celle des processus ayant une valeur `nice` maximale peut décroître avec une demi-vie de 60 secondes.

Le résultat est que les processus visés par `nice` obtiennent une part des ressources d'UC inférieure à celle des autres processus sur le même noeud limite. Sous Solaris Resource Manager, `nice` a peu d'incidence sur les taux d'exécution des processus exécutés sur des noeuds différents, sauf si le nombre de processus dans la file d'attente des processus exécutables excède le nombre d'UC.

Sous Solaris Resource Manager, les processus ayant une valeur `nice` maximale (par exemple, ceux qui démarrent avec une commande `nice -19`) font l'objet d'un traitement spécial : ils obtiennent du temps d'UC uniquement si les autres processus n'en font pas la demande et seraient de toutes façons inactifs.

Pour en savoir davantage sur la relation entre Solaris Resource Manager et d'autres fonctions de contrôle des ressources, voir "Lien avec les autres fonctions de contrôle des ressources de Solaris" , page 4.

## Reconfiguration dynamique

La fonction de reconfiguration dynamique (RD) des serveurs Sun Enterprise permet aux utilisateurs d'ajouter et de supprimer dynamiquement des cartes systèmes comportant des ressources matérielles comme des processeurs, de la mémoire et des périphériques d'E-S. Aux fins de l'ordonnancement, Solaris Resource Manager effectue le suivi des ressources processeur disponibles et gère les changements en conséquence, redistribuant équitablement les ressources libres entre les utilisateurs et les processus admissibles.

Solaris Resource Manager contrôlant uniquement la taille de mémoire virtuelle des processus et non la mémoire physique consommée par les processus et les utilisateurs, l'incidence d'une opération RD sur la mémoire n'a aucun effet sur la vérification des limites de mémoire effectuée par Solaris Resource Manager.

---

## Noeud limite inactif

Le noeud limite inactif (`srmiddle`) est celui qui est assigné par l'administrateur central pour facturer tous les coûts d'UC inactive du noyau. À l'installation, l'utilisateur inactif (nom d'utilisateur `srmiddle`) a été créé avec une valeur de `41`. Le noeud limite inactif doit avoir zéro part afin d'assurer que ses processus sont exécutés uniquement si aucun autre processus n'est actif. Le noeud inactif est assigné à l'aide de la commande `srmadm(1MSRM)`.

Au démarrage, le noeud inactif par défaut est le noeud limite racine. Lors du passage en mode multi-utilisateurs, le script `init.d(4)` définit le noeud limite inactif comme étant celui du compte `srmiddle` si celui-ci existe. Cette procédure peut être personnalisée en précisant un autre noeud limite dans le script `/etc/init.d/init.srm`.

Si le noeud inactif n'est pas le noeud racine, il doit s'agir d'un enfant direct de celui-ci.

---

## Noeud limite autre

Le noeud limite autre (`srmother`) est celui qui est assigné par l'administrateur système en tant que noeud limite père par défaut pour les nouveaux utilisateurs créés après l'installation initiale (où le noeud limite père par défaut est la racine). Le noeud autre est créé automatiquement à l'installation avec une valeur par défaut de `1` part, ce qui assure que les noeuds limites qui y sont connectés pourront accéder à l'unité centrale.

Le noeud autre ne doit avoir aucune limite de ressources, une part d'UC de `1` ou plus, et aucun privilège spécial.

---

## Noeud limite perdu

Sous Solaris Resource Manager, l'appel système `setuid(2SRM)` a comme effet secondaire de relier le processus appelant à un nouveau noeud limite. Si le changement de liaison échoue, ce qui se produit habituellement parce que le nouveau noeud n'existe pas, le processus est alors relié au noeud perdu (`srmlost`), ce dernier étant créé à l'installation de Solaris Resource Manager. Si cette liaison échoue aussi ou qu'aucun noeud perdu n'a été assigné, la fonction `setuid(2SRM)` n'est pas touchée et le processus demeure sur son noeud limite actuel.

Le script `init.srm` définit le noeud perdu pendant le passage au mode multi-utilisateurs. Cette procédure peut être contournée en précisant le noeud limite à utiliser dans le fichier `/etc/init.d/init.srm`. Pour éviter les failles de sécurité, le noeud perdu doit avoir une part d'UC de 1 et aucun privilège spécial (si vous modifiez les valeurs, tenez compte des besoins de cet utilisateur lors des changements).

## Limites de mémoire, de mémoire de processus et de comptage de processus

---

Solaris Resource Manager permet à un administrateur de contrôler :

- La quantité de mémoire virtuelle détenue par des utilisateurs ou par une application (il s'agit d'une mesure de l'usage de mémoire virtuelle de tous les processus au niveau du noeud limite).
- La quantité de mémoire virtuelle détenue par un groupe d'utilisateurs ou par une application (il s'agit d'une mesure de l'usage de mémoire virtuelle de tous les processus reliés au chef de groupe ou à ses enfants, effectuée au niveau du noeud limite du chef de groupe).
- La quantité de mémoire virtuelle détenue par un processus.
- Le nombre de processus pouvant être exécutés simultanément par un utilisateur/ noeud limite.

---

## Attributs pour le contrôle de la mémoire virtuelle (total)

Plusieurs attributs système sont utilisés pour enregistrer l'usage de mémoire virtuelle et lui assigner des limites.

Les attributs système permettant de gérer l'usage de la mémoire sont les suivants :

`memory.myusage`

L'attribut `memory.myusage` d'un noeud limite est égal à la somme de l'usage de la mémoire

virtuelle de tous les processus actuellement reliés à ce noeud.

`memory.usage`

L'attribut `memory.usage` d'un noeud limite est égal à la somme de son attribut `memory.myusage` et des attributs `memory.usage` de ses noeuds enfants.

`memory.limit`

L'attribut `memory.limit` est appliqué à l'attribut `memory.usage`. Il limite l'usage conjoint de la mémoire par tous les processus reliés au noeud limite et aux noeuds membres.

---

## Attribut pour le contrôle de la mémoire de processus (par processus)

L'attribut système ci-dessous permet d'enregistrer l'usage des processus et d'y assigner des limites :

`memory.plimit`

L'attribut `memory.plimit` d'un noeud limite est une limite par processus appliquée séparément à l'usage de mémoire de chaque processus qui est relié à ce noeud ou à ses membres.

---

## Description technique des limites de mémoire

Chaque fois qu'un processus tente d'accroître sa taille de mémoire, il est assujéti aux limites de mémoire (totale et par processus) du noeud limite auquel il est relié. Un processus dispose de cinq méthodes pour tenter d'accroître sa taille de mémoire :

1. L'appel d'une routine d'attribution telle que `malloc(3C)`, qui lance un appel système `sbrk(2)`. Si une limite de mémoire est dépassée, l'appel renvoie une erreur portant le code `ENOMEM`.

2. L'expansion d'une pile provoquant une défaillance de pile, ce qui cause normalement le transfert d'une page de mémoire additionnelle au processus. Si une limite de mémoire est dépassée, un signal SIGSEGV est envoyé au processus.
3. Utilisation de `mmap(2)`.
4. Utilisation de `fork(2)`. L'espace adresse enfant est dupliqué tout en demeurant la propriété du processus père. Lors de la duplication, le nouvel espace d'adressage ne dépassera pas `memory.plimit` puisque le père doit déjà se trouver dans cette limite. Cependant, l'attribution est tributaire de `memory.limit`.
5. Utilisation de `exec(2)`. Lors d'une commande `exec`, l'usage mémoire décroît lorsque l'ancien espace adresse est supprimé. Cependant, si l'espace d'adressage du nouveau programme est supérieur et entraîne le dépassement de la limite, `exec` peut échouer.

---

## Reconfiguration dynamique et limites de mémoire virtuelle

La fonction de reconfiguration dynamique (RD) des systèmes Enterprise 10000 a peu d'effets sur les limites de mémoire virtuelle imposées aux noeuds limites. Plus précisément, la RD permet d'ajouter ou de supprimer de la mémoire physique du système pendant son fonctionnement (le pool de mémoire virtuelle du système comprend toute la mémoire physique et l'espace d'échange configuré dans le système). De plus, la commande `swap(1M)` peut être exécutée sur tout système Solaris pour ajouter (`-a`) ou supprimer (`-d`) de l'espace d'échange sur le système. De cette façon, la mémoire virtuelle totale peut augmenter et diminuer pendant le fonctionnement du système.

Cette procédure a un effet indirect sur les limites de mémoire virtuelle imposées par Solaris Resource Manager. La mémoire virtuelle étant gérée en unités absolues (plutôt que par parts), les limites effectives ne changent pas lorsque les ressources du système sont modifiées pendant son fonctionnement (notez que cet effet est différent de celui de la reconfiguration dynamique des processeurs, décrite à la section "Reconfiguration dynamique", page 70).

---

## Attributs pour le contrôle du comptage de processus

Les attributs système utilisés pour gérer l'usage des processus (nombre de processus) sont les suivants :

<code>process.myusage</code>	L'attribut <code>process.myusage</code> d'un noeud limite est égal au nombre de processus reliés à ce noeud.
<code>process.usage</code>	L'attribut <code>process.usage</code> d'un noeud limite est égal à la somme de son attribut <code>process.myusage</code> et des attributs <code>process.usage</code> de ses noeuds enfants.
<code>process.limit</code>	L'attribut <code>process.limit</code> est une limite appliquée à l'attribut <code>process.usage</code> d'un noeud limite afin de limiter le nombre de processus pouvant être reliés simultanément à ce noeud et à tous les noeuds membres.

---

## Description technique du comptage de processus

Les appels système `fork(2)`, `fork1(2)` et `vfork(2)` sont utilisés pour créer des processus. Si la limite de processus était dépassée, l'appel système échouerait et renverrait une erreur EAGAIN. La plupart des programmes interprètent EAGAIN comme un manque temporaire de ressources système et réexécutent `fork` après une courte pause. Si l'échec de `fork` est causé par une limite de Solaris Resource Manager, cela peut générer une boucle qui sera exécutée indéfiniment du fait que EAGAIN sera renvoyé à chaque tentative jusqu'à ce que la limite fixée pour le noeud concerné soit atteinte.



## Données d'usage

---

Solaris Resource Manager permet à l'administrateur d'effectuer une collecte précise des valeurs cumulatives d'usage pour les ressources du système, des applications et des utilisateurs. De concert avec les fonctions et utilitaires fournis par Solaris Resource Manager, ces données peuvent servir de base pour le développement d'un système de facturation en fonction de l'usage des ressources.

---

### Attributs cumulatifs

Les attributs cumulatifs d'un noeud limite sont utilisés pour stocker des informations sur l'accroissement de l'usage des ressources. Par exemple, l'attribut `cpu accrue` contient l'usage cumulatif de l'UC pour tous les noeuds limites du groupe, ainsi que celui du noeud en cours. Lorsque les attributs cumulatifs d'un noeud limite sont actualisés, les changements sont également appliqués à son père (comme pour les changements apportés à l'attribut d'usage), et ainsi de suite jusqu'au noeud racine. L'usage cumulatif de chaque niveau de l'arbre est donc la somme de l'usage cumulatif du noeud limite et de celui de ses enfants, s'ils existent.

---

### Facturation

L'administrateur doit décider quels sont les noeuds à facturer pour l'usage des ressources. Par exemple, il peut se préoccuper uniquement de facturer des services entiers et ainsi facturer uniquement les chefs de groupe ou les groupes des niveaux supérieurs, dont l'usage cumulatif comprendra celui de tous les noeuds limites des niveaux inférieurs des différents services.

Pour pouvoir mettre en oeuvre un système de facturation, l'administrateur doit définir une fonction d'évaluation des coûts pour chaque ressource facturée. Il peut s'agir d'une simple relation linéaire (où le coût unitaire demeure le même, quel que soit le montant utilisé) ou encore, d'une relation non linéaire, par exemple, une fonction en escalier ou une courbe où le coût unitaire varie en fonction de l'usage.

Pour déterminer la fonction d'évaluation pour chaque ressource, l'administrateur doit garder à l'esprit qu'en plus de contrôler l'affectation des coûts par rapport à l'augmentation de l'usage des ressources, elle pourrait aussi avoir une incidence sur la façon dont un utilisateur se sert de la ressource. Par exemple, si la fonction d'évaluation pour l'usage de la mémoire virtuelle accroît le coût unitaire lorsque l'usage augmente, les utilisateurs veilleront à économiser l'usage de la mémoire virtuelle. Par conséquent, l'administrateur peut contrôler le comportement des utilisateurs grâce à une stratégie pertinente d'attribution des coûts.

Chaque ressource compte un seul attribut cumulatif, qui contient l'usage cumulatif pour la ressource basé sur l'attribut d'usage de celle-ci. Cela signifie qu'il n'y a pas d'usage cumulatif correspondant à l'attribut `myusage`. Dans le cas des chefs de groupe, il n'y a pas d'usage cumulatif pour l'utilisateur seul, car l'attribut cumulatif contient l'usage cumulatif du groupe. Cela ne concerne pas les noeuds limites sans enfant et les noeuds limites feuille, puisque `myusage` et l'attribut d'usage sont identiques. Si une facture est requise pour l'usage cumulatif individuel d'un chef de groupe, elle doit être calculée d'après le total du groupe moins la somme des totaux de tous les enfants du groupe.

## Commande `liminfo`

La sortie par défaut de la commande `liminfo(1SRM)` est destinée aux utilisateurs qui désirent connaître leurs usages actuels, leurs limites, leurs privilèges, etc. `liminfo` permet en outre aux administrateurs de connaître les attributs d'autres utilisateurs. Différents formats de rapport sont disponibles, et des options permettent de rendre la sortie de `liminfo` compatible pour être traitée par un pipeline de filtres. Pour des informations détaillées sur les options de ligne de commande et leur signification et pour connaître les champs pouvant être affichés, voir `liminfo(1SRM)`.

## Commande `limreport`

La commande `limreport(1SRM)` permet à l'administrateur d'obtenir des informations sur n'importe quel attribut de n'importe quel utilisateur, y compris sur l'attribut cumulatif. Elle fournit un outil polyvalent pour sélectionner les informations à afficher sur les noeuds limites voulus.

Par exemple, la commande :

```
% limreport 'cpu.accrue!=0' '%u %u %10d\n' uid lname cpu.accrue
```

sélectionne tous les noeuds limites ayant n'importe quel usage cumulatif dans le domaine usr et indique l'UID et l'attribut d'usage cumulatif de chacun des noeuds sélectionnés. Pour trier ces valeurs par `cpu.accrue` et lister uniquement les dix premiers utilisateurs, il suffit de transmettre le résultat à une commande "sort", comme ceci :

```
% limreport 'cpu.accrue!=0' '%u %u %10d\n' uid lname cpu.accrue | sort -2n | head
```

## Commande `limadm`

La commande `limadm(1MSRM)` peut être utilisée dans un système de facturation pour remettre à zéro les attributs cumulatifs après leur facturation. Par exemple, la commande :

```
# limreport 1 '%u\tcpu.accrue=0,mem.accrue=0\n' uid | limadm set -u -f
```

utilise `limreport(1SRM)` pour générer la liste des commandes transmises à `limadm`. Chaque noeud limite est sélectionné et l'attribut cumulatif de chacun est remis à zéro.

L'administrateur doit choisir avec soin le moment voulu pour remettre l'attribut cumulatif d'un noeud limite à zéro, qui dépendra de la stratégie de facturation. Par exemple, si des factures sont produites pour les groupes puis pour chaque membre, les attributs cumulatifs des membres ne doivent pas être remis à zéro avant la production des deux factures. Cependant, si aucune facture individuelle n'est produite, les attributs cumulatifs des membres du groupe doivent être remis à zéro en même temps que celui du chef de groupe, et ce, même s'ils n'ont pas été utilisés individuellement.



## Dépannage

---

Cette section est conçue pour aider les administrateurs à diagnostiquer les problèmes relatif à l'utilisation de Solaris Resource Manager.

---

### Un utilisateur ne peut pas ouvrir une session

- Il n'y a aucun noeud limite correspondant à son UID. Aucun administrateur n'a configuré de noeud limite pour le compte de cet utilisateur. Le problème est signalé par un message indiquant qu'aucune information sur les limites n'est disponible. Reportez-vous à la section "Noeuds limites orphelins" , page 86.
- L'indicateur `nologin` ou `noattach` de l'utilisateur est activé.
- L'indicateur `onelogin` de l'utilisateur est activé, et celui-ci a déjà une session en cours sur un autre terminal ou une autre fenêtre.
- L'utilisateur a atteint sa limite de temps de connexion. Il doit attendre que l'usage diminue avant de recommencer, ou l'administrateur peut changer l'attribut `terminal.usage` ou `terminal.limit` de l'utilisateur pour accroître le temps de connexion.
- Le noeud limite de l'utilisateur est devenu orphelin suite à la suppression de son noeud père. Voir "Noeuds limites orphelins" , page 86.

Aucune des restrictions de Solaris Resource Manager mentionnées ci-dessus ne s'applique à l'utilisateur racine.

---

## Un utilisateur n'est pas informé lorsqu'il atteint ses limites

Pendant le fonctionnement normal de Solaris Resource Manager, l'utilisateur en session reçoit des messages chaque fois qu'une limite est dépassée. L'utilisateur peut ne pas être au courant du problème et le système semblera fonctionner anormalement, mais l'administrateur système sera informé du problème.

La transmission des messages est effectuée par le programme démon `limdaemon` de Solaris Resource Manager. Si les messages ne sont pas transmis, l'administrateur peut étudier plusieurs possibilités : --

- La fenêtre de console est masquée. Si l'utilisateur ouvre une session dans une fenêtre puis ouvre d'autres fenêtres qui masquent la première, il ne verra pas les messages qui y sont affichés.
- Le programme `limdaemon(1MSRM)` n'est pas en cours d'exécution.
- `limdaemon` est incapable d'allouer dynamiquement de la mémoire additionnelle pour maintenir ses structures internes. Si cela se produit, `limdaemon` affiche un message de diagnostic sur la console système dès le premier échec. Il continue ensuite à essayer d'obtenir de la mémoire mais n'affiche plus de messages.
- Le fichier `utmp` est endommagé ou manquant. `limdaemon` utilise ce fichier pour déterminer les terminaux où un utilisateur est en session afin d'y envoyer des messages. Si le fichier `utmp` est endommagé ou manquant, un message d'erreur est signalé à la console et la transmission des messages est interrompue.
- `limdaemon` est incapable de transmettre un message en raison d'une restriction système. Par exemple, si `limdaemon` ne peut pas ouvrir une fenêtre sur un terminal afin de transmettre un message, celui-ci est annulé.—

---

## Impossible de changer le groupe d'un utilisateur

L'attribut `sgroup` indique le père d'un noeud limite dans l'arbre d'ordonnancement. Cette hiérarchie permet de réguler l'usage des ressources et d'ordonner l'UC. Pour cette raison, la modification de l'attribut `sgroup` fait l'objet de plusieurs mesures de sécurité afin d'éviter les erreurs de manipulation ou le contournement de Solaris Resource Manager.

Pour modifier l'attribut `sgroup`, l'utilisateur doit posséder l'un des privilèges suivants :

- être le superutilisateur ;
- avoir un indicateur `uselimadm` activé ;
- avoir un indicateur `admin` activé et être chef de groupe pour le noeud limite à modifier.

Un noeud limite orphelin ne peut être le père d'autres noeuds limites. Voir "Noeuds limites orphelins" , page 86.

---

## Base de données des limites endommagée

Une base de données des limites endommagée est un problème important pour l'administrateur. Aucun symptôme ne permet de déterminer avec certitude si la base a été endommagée, mais certains indices peuvent indiquer une telle situation.

Si la base de données est endommagée, l'administrateur doit en restaurer une version valide et rétablir la configuration correspondante. Si les dommages sont limités à une partie restreinte de la base de données, l'administrateur peut parvenir à enregistrer le contenu de tous les autres noeuds limites, puis les restaurer dans une nouvelle base de données des limites à l'aide des commandes `limreport(1SRM)` et `limadm(1MSRM)`.

Pour en savoir davantage sur la détection et la correction d'une base de données des limites endommagée, voir "Base de données des limites endommagée" , page 89.

---

## Temps de connexion d'un terminal non actualisé

La cause probable est que le programme `limdaemon(1MSRM)` n'est pas en cours d'exécution. `limdaemon` met périodiquement à jour les attributs d'usage et cumulatif dans la catégorie de terminal pour tous les utilisateurs en session. Normalement, il devrait être démarré depuis le script `init.d(4)` de Solaris Resource Manager.

---

## Des utilisateurs dépassent souvent des limites

Il existe plusieurs causes probables : –

- La limite administrative de l'utilisateur est trop basse pour ses besoins.
- L'attribut d'usage ne décroît pas. L'administrateur est chargé de s'assurer que le décroissement est appliqué aux catégories de périphérique pour toutes les ressources renouvelables (incluant la catégorie de terminal). Normalement, cela est effectué en exécutant régulièrement la commande `limdaemon(1MSRM)`. Si le décroissement n'est pas appliqué à une ressource renouvelable, l'attribut d'usage de cette ressource continue à croître jusqu'à ce que sa limite soit atteinte.
- L'intervalle entre les exécutions de `limdecay` est trop long. La fréquence d'exécution de `limdaemon(1MSRM)` doit être définie en fonction de la granularité de l'intervalle de décroissement le plus court.
- L'attribut de décroissement d'une ressource renouvelable est trop petit ou l'intervalle est trop grand. Si le décroissement d'une ressource renouvelable pendant un intervalle donné est fixé à une valeur inférieure au taux de consommation habituel de cette ressource, l'attribut d'usage augmentera progressivement jusqu'à ce que la limite soit atteinte.

---

## Le système est lent

### Processus du noeud limite racine

Aux fins de la gestion du système, les processus reliés au noeud limite racine obtiennent pratiquement toutes les ressources d'UC demandées. Par conséquent, si un processus d'UC est relié au noeud racine, il peut monopoliser et ralentir, voire même arrêter les processus des autres noeuds limites.

Pour éviter ce genre de problème, les précautions ci-après peuvent être prises :

- Pour l'utilisation normale, l'administrateur doit toujours ouvrir une session sur l'un de ses propres noeuds et non sur le noeud racine. S'il doit se relier au noeud racine, il ne doit pas exécuter d'applications faisant usage intensif de l'UC, par exemple, des compilateurs. Pour se servir d'une UID de superutilisateur sans se relier au noeud racine, l'administrateur peut utiliser la commande `su(1)`.
- Les scripts `init.d(4)` devraient être modifiés pour utiliser le programme `srmsuser(1SRM)` en vue de relier tous les démons à des noeuds limites qui leur appartiennent, ce qui évite de les relier (par héritage) au noeud limite racine.



Un programme exécuté en tant que `setuid-root` n'est pas automatiquement relié au noeud limite. Normalement, le processus demeure relié au noeud limite du père qui l'a créé, et seule l'UID en cours est changée.

## Ressources d'unité centrale non contrôlées par Solaris Resource Manager

Solaris Resource Manager contrôle uniquement l'usage de l'UC par les processus de la classe d'ordonnancement SHR. Si des demandes excessives d'une priorité supérieure sont faites par d'autres classes d'ordonnancement, spécialement les classes temps réel (RT) et système (SYS), SHR peut ordonnancer seulement avec les ressources restantes de l'UC.

L'utilisation de la classe RT génère un conflit avec les fonctions de gestion de système de Solaris Resource Manager. Les processus en temps réel obtiennent un accès complet au système afin qu'ils puissent transmettre des réponses en temps réel (habituellement de l'ordre de quelques microsecondes). Les processus exécutés dans la classe SHR ont par définition une priorité inférieure à celle de tout processus en temps réel, sur lesquels Solaris Resource Manager n'a aucun contrôle. Les processus en temps réel peuvent facilement consommer *toutes* les ressources disponibles, privant Solaris Resource Manager de ressources à attribuer aux autres processus.

le serveur NFS est l'un des services exécuté *exclusivement* dans la classe SYS. Solaris Resource Manager ne peut contrôler les démons NFS, car ils sont exécutés dans SYS. La capacité de Solaris Resource Manager à attribuer des ressources de processeur peut être réduite sur les systèmes offrant un service NFS étendu.

Même si des processus exécutent du code de noyau (à l'intérieur d'un appel système), les règles de priorité de tranche de temps habituelles ne s'appliquent pas. La plupart des appels système exécuteront seulement une certaine partie de leurs tâches avant d'atteindre un point de préemption. Cependant, si le système subit une charge élevée d'appels système plus intensifs, cela peut causer une diminution de la réactivité globale, ce qui est hors de la portée de contrôle d'une classe d'ordonnancement.

Si le système manque de mémoire réelle, le goulot d'étranglement d'E-S résultant de l'accroissement du taux d'erreurs de page et des échanges de processus augmente la consommation des ressources de l'UC par le noyau. Des temps d'attente d'E-S importants peuvent indiquer des pertes de capacité d'UC. Encore une fois, cela est hors de la portée d'une classe d'ordonnancement.

La classe d'ordonnancement SHR est un ordonnanceur à temps partagé (TS) qui utilise la même gamme de priorité globale que les ordonnanceurs TS et IA. Il n'est pas recommandé de combiner l'utilisation de SHR avec TS et IA, sauf pour transférer tous les processus dans ou depuis la classe SHR. L'utilisation du système avec une combinaison de processus des classes SHR et TS diminuera la qualité de l'ordonnancement dans les deux classes. Pour cette raison, Solaris Resource Manager

empêche les processus non racine de se déplacer ou de déplacer d'autres processus vers les classes TS ou IA. La classe RT fait appel à une autre gamme de priorité et peut être utilisée avec la classe SHR de la même façon que les classes TS et IA.

Si des processus exécutés par racine contiennent du code utilisant l'appel système `prctl(2)` directement au lieu de la routine de bibliothèque `setpriority(3C)` pour ajuster les priorités des processus, ils peuvent transférer les processus cibles dans une autre classe d'ordonnement (habituellement TS). Le code de la routine `setpriority(3C)` tient compte du fait que l'interface `prctl(2)` avec SHR est binairement compatible avec celle de TS, et contourne ainsi le problème. L'option `-c` de `ps(1)` ou l'option `-d` de `prctl(1)` peut être utilisée pour afficher la classe d'ordonnement des processus.

Le même problème se manifeste avec les processus de privilège racine qui utilisent `prctl(2)` explicitement pour gérer l'appartenance de classe d'ordonnement des processus.

---

## Messages inhabituels

Des messages sont envoyés à tout utilisateur ayant atteint une limite. Par conséquent, si une limite de groupe est atteinte, le chef de groupe et tous les utilisateurs des niveaux inférieurs de la hiérarchie d'ordonnement du groupe recevront un message d'avertissement.

Si une limite est atteinte alors qu'un utilisateur est relié à un autre noeud limite, cet utilisateur ne recevra pas de message, mais tous les autres le recevront. La source du problème peut ne pas être apparente pour le groupe concerné.

---

## Noeuds limites orphelins

Un noeud limite orphelin n'a pas de noeud limite père. Cela doit être pris en compte par l'administrateur, car Solaris Resource Manager empêche les processus de se relier à un noeud limite orphelin ou dont l'arbre d'ordonnement comporte un ascendant orphelin.

Le noyau vérifie les changements apportés à l'attribut `sgroup` afin d'éviter la création d'orphelins suite à des modifications non valides au père du groupe d'ordonnement.

Le principal effet d'un noeud limite orphelin est qu'il devient impossible d'y relier des processus. Aucun processus n'y étant relié, le noeud limite ne peut pas être utilisé pour ouvrir une session, et toute tentative avec le compte correspondant échouera.

La méthode la plus facile pour l'administrateur de détecter les noeuds limites orphelins consiste à utiliser la commande `limreport(1SRM)` avec l'identificateur d'orphelin intégré. La commande :

```
% limreport orphan - uid sgroup lname
```

indiquera l'UID, le père du groupe d'ordonnement et le compte des utilisateurs ayant des noeuds limites orphelins. L'attribut `sgroup` peut être utilisé pour repérer les noeuds limites se trouvant à la racine d'une section orpheline de l'arbre.

Lorsque l'administrateur découvre un noeud orphelin, la première étape consiste à repérer la racine de la section orpheline dans l'arbre d'ordonnement puisqu'il s'agit du noeud limite devant être relié. Si la racine de la section orpheline n'est pas identifiée correctement, seule une partie de la section orpheline sera reliée à l'arbre.

Une fois la racine de la section orpheline identifiée, l'administrateur disposant de privilèges suffisants peut utiliser la commande `limadm(1MSRM)` pour associer l'attribut `sgroup` du noeud limite orphelin de premier niveau à un noeud limite valide dans l'arbre. Le noeud orphelin sera relié à l'arbre en tant que membre du groupe dont le chef est le noeud limite valide. `limadm` vérifie si le père du nouveau groupe d'ordonnement peut être activé, ce qui assure que le noeud limite modifié ne sera plus orphelin.

L'administrateur peut aussi créer un utilisateur dont l'UID est identique à celle de l'attribut `sgroup` du noeud orphelin, ce qui reliera automatiquement la section orpheline de l'arbre.

---

## Boucles de groupe

Lorsqu'un noeud limite est activé, tous ses parents jusqu'au noeud limite racine le sont aussi. Si, ce faisant, l'un des noeuds limites a un père ayant déjà été détecté, cela signifie que le noyau a découvert une boucle de groupe.

Si la base de données des limites est endommagée, une boucle de groupe peut être créée (l'un des ascendants d'un noeud limite est également l'un de ses enfants). Le cas échéant, le noyau relie automatiquement la boucle à l'arbre d'ordonnement en la brisant de façon arbitraire pour la relier en tant que groupe sous le noeud limite racine. Lorsque cela se produit, le noeud limite auquel la boucle est reliée sur l'arbre d'ordonnement devient le chef d'un groupe de premier niveau. Il est possible que des membres de ce groupe héritent de privilèges ou de limites supérieurs à ceux auxquels ils auraient normalement droit.

## Causes

Les boucles de groupe sont évitées par `limadm(1MSRM)` lors de la configuration de parents de groupe d'ordonnement. La seule cause d'une boucle de groupe est une base de données de limites endommagées. Il s'agit d'un problème grave pouvant générer toutes sortes de défaillances sous Solaris Resource Manager du fait que la base de données des limites est essentielle à son fonctionnement.

## Effets

Lorsque le noyau détecte une boucle de groupe, il relie silencieusement l'un des noeuds limites de la boucle directement sous le noeud limite racine. La boucle n'ayant ni début ni fin, le noyau ne peut déterminer quel est le noeud limite supérieur.

## Correction

Ce problème se corrige de lui-même dans la structure de l'arbre d'ordonnement puisque le noyau relie le noeud limite au noeud racine. La liaison étant effectuée à un point arbitraire de la boucle, l'administrateur doit déterminer où le noeud limite devrait être relié et vérifier également le point de connexion de tous les autres membres de la boucle.

Le résultat d'une réparation automatique de boucle de groupe peut être affiché en dressant la liste des noeuds limites enfants du noeud racine. La commande :

```
% limreport 'sgroup==0' - uid lname
```

affiche la liste de tous les noeuds limites dont le père est le noeud limite racine. Si des noeuds limites de cette liste ne devraient pas être des enfants du noeud limite racine, cela peut indiquer qu'ils sont au premier niveau d'une boucle de groupe ayant été reliée sous le noeud racine.

Pour l'administrateur, la principale préoccupation lorsqu'une boucle de groupe est détectée est que de nombreux problèmes beaucoup plus graves pourraient se manifester du fait que la base de données des limites est endommagée. Si l'administrateur soupçonne une base de données des limites endommagée, il doit en vérifier la validité afin de confirmer le problème et prendre les mesures qui s'imposent. Pour savoir comment détecter et corriger une base de données des limites endommagée, voir "Reprise après une panne", page 89.

## Reprise après une panne

La défaillance d'un système Solaris est source de nombreuses préoccupations pour l'administrateur, et d'autres facteurs doivent être considérés lorsqu'un système Solaris Resource Manager est utilisé, soit :

- la possibilité que la base de données des limites ait été endommagée par une défaillance de disque ou un autre incident matériel ;
- le fonctionnement de `limdaemon(1MSRM)` pendant une panne de système, notamment en ce qui concerne la facturation du temps de connexion.

Les sections ci-après traitent de ces sujets en détail et fournissent des suggestions pour corriger la situation.

### Base de données des limites endommagée

Les fonctions de maintenance de la base de données des limites de Solaris Resource Manager sont très robustes et les dommages sont peu probables. Cependant, si cela se produit, il s'agit d'un problème important pour l'administrateur, car cette base de données est essentielle au fonctionnement de Solaris Resource Manager. Tout dommage potentiel doit faire l'objet d'une recherche minutieuse et être réparé s'il est confirmé.

### *Symptômes*

Aucun symptôme ne permet de déterminer avec certitude si la base de données des limites a été endommagée, mais certains indices peuvent indiquer une telle situation :

- La détection d'une boucle de groupe par Solaris Resource Manager. Différentes fonctions de Solaris Resource Manager permettent d'éviter les boucles de groupe ; seul un attribut `sgroup` endommagé peut en être la source. Pour en savoir davantage, voir "Boucles de groupe" , page 87.
- L'affichage du message "No limits information available" lorsqu'un utilisateur tente d'ouvrir une session et que son compte est rejeté. Cela peut se produire si les dommages à la base de données des limites provoquent la désactivation de l'attribut `flag.real`, ce qui supprime le noeud limite de l'utilisateur ainsi que tout noeud limite orphelin (pour en savoir davantage, reportez-vous à la section Noeuds limites orphelins). Notez que le message "No limits information available" apparaîtra également si aucun noeud limite n'a été créé pour le compte ou s'il a été supprimé intentionnellement; il ne signifie donc pas nécessairement que la base de données des limites a été endommagée.
- Des valeurs non valides apparaissent soudainement dans les attributs d'usage ou de limites. Cela peut faire en sorte que des utilisateurs atteignent subitement des limites.

- Des utilisateurs signalent des pertes de privilèges ou l'obtention de privilèges inhabituels, ce qui est causé par des indicateurs de privilèges altérés.

Lorsque l'administrateur soupçonne que la base de données des limites a été endommagée, la façon la plus sûre de le confirmer consiste à exécuter `limreport(1SRM)` afin de demander la liste des noeuds limites ayant des attributs dont les valeurs ne se situent pas dans leur plage de valeurs habituelle. Cette commande permet également de dresser la liste des noeuds limites dont l'indicateur `flag.real` est désactivé, ce qui indique la présence, dans la table des mots de passe, de comptes n'ayant pas de noeud limite.

### *Solution*

Si des dommages sont détectés, l'administrateur doit restaurer une version valide de la base de données des limites. Si les dommages sont limités à une partie restreinte de la base de données, l'administrateur peut parvenir à enregistrer le contenu de tous les autres noeuds limites, puis les restaurer dans une nouvelle base de données des limites à l'aide des commandes `limreport(1SRM)` et `limadm(1MSRM)`. Si une copie récente de la base de données n'est pas disponible, cela est préférable, puisque la nouvelle base de données contiendrait les attributs d'usage et cumulatifs les plus récents. Pour connaître la procédure de sauvegarde et de restauration de la base de données des limites, consultez la section Gestion des noeuds limites. Pour les cas simples de noeuds limites manquants, il peut être suffisant de simplement les recréer à l'aide la commande `limadm`.

### Perte de temps de connexion par `limdaemon`

Si `limdaemon(1MSRM)` est interrompu, tous les utilisateurs en session ne sont plus facturés pour leur temps de connexion. En outre, lorsque `limdaemon(1MSRM)` est redémarré, les utilisateurs en session continue à utiliser leur terminal gratuitement. Cela s'explique par le fait que le démon utilise les avis d'ouverture de session fournis par `login(1)` pour établir un relevé de session Solaris Resource Manager depuis les structures internes utilisées pour calculer les frais de temps de connexion. Par conséquent, chaque fois que le démon démarre, aucune session Solaris Resource Manager n'est établie avant la réception du premier avis.

Cela n'est habituellement pas un problème si l'interruption de `limdaemon` est causée par une panne du système, celle-ci entraînant également l'arrêt des autres processus. Aucune session ne peut donc reprendre avant le redémarrage du système.

Si `limdaemon(1MSRM)` est arrêté pour une autre raison, l'administrateur a deux choix :

1. Redémarrer le démon immédiatement et ne pas tenir compte des pertes de facturation de temps de terminal des utilisateurs déjà en session, auquel cas un

utilisateur pourrait utiliser gratuitement un terminal jusqu'à ce qu'il soit détecté et que sa session soit fermée.

2. Mettre le système en mode mono-utilisateur, puis de nouveau en mode multi-utilisateurs, ce qui met fin à toutes les sessions en cours et empêche les utilisateurs d'ouvrir une session de nouveau avant le redémarrage du démon.





## Messages d'erreur

---

Les messages d'erreur du système Solaris Resource Manager sont les suivants :

### **memory limit reached**

L'usage de mémoire virtuelle (`memory.usage`) d'un noeud limite a atteint la limite applicable (`memory.limit`).

### **process limit reached**

Le nombre de processus (`process.usage`) d'un noeud limite a atteint la limite applicable (`process.limit`).

### **per-process memory limit reached**

L'usage de mémoire virtuelle d'un processus a atteint la limite applicable (`memory.plimit`).

### **Inode attach failed in `setuid`**

Si le système Solaris Resource Manager est installé activé, l'appel système `setuid`, en plus de sa fonction standard, relie le processus appelant au noeud limite associé à sa nouvelle UID réelle. Si la liaison échoue, c'est habituellement parce qu'il n'y a pas de noeud limite associé à la nouvelle UID.

### **currently barred from logging in**

`flag.nologin` est activé pour l'utilisateur au moment où il tente d'ouvrir une session.

**already logged in - only one login allowed**

`flag.onelogin` est activé pour l'utilisateur, et celui-ci a déjà ouvert une session sur un autre terminal.

**no permission to use this terminal**

`terminal.flag.all`, `terminal.flag.console`, `terminal.hardwired` ou `terminal.flag.network` est active pour l'utilisateur, et celui-ci tente d'ouvrir une session sur un terminal donné.

**terminal connect time limit reached**

Le temps de connexion terminal (`terminal.usage`) a atteint la limite applicable (`terminal.limit`).

# Exemples de script Solaris Resource Manager

---

---

## Script de démarrage

Le script de démarrage ci-après est fourni avec le système (fichier /etc/init.d/init.srm) et est exécuté avec un argument de démarrage lorsque le système passe au niveau d'exécution 2 ou 3 (mode multi-utilisateurs). Il est également exécuté avec un argument d'arrêt lors de l'arrêt du système.

```
#!/bin/sh
#
# Copyright (c) 1998 by Sun Microsystems, Inc.
# All rights reserved.
#
# Copyright 1995-1997 Softway Pty. Ltd.
#
# Start/stop Solaris Resource Manager v1.0
#
#ident    "@(#)init.srm 1.17 98/10/27 SMI"

#####
# Default values.

DATADIR=/var/srm
ShareDb=$DATADIR/srmDB
LimdaemonOptions=
ChargeOptionsOn="limits=y:share=y:adjgroups=y:limshare=y"
ChargeOptionsOff="limits=n:share=n:adjgroups=n:limshare=n"

LostLnode=srmlost
IdleLnode=srmidle
```

```

#####

# ECHO=echo # For a verbose startup and shutdown
ECHO=: # For a quiet startup and shutdown

SRMDIR=/usr/srm
SRMBIN=$SRMDIR/bin
SRMSBIN=$SRMDIR/sbin
ETCSRMR=/etc/srm

PATH=/sbin:/usr/sbin:/bin:$PATH:$SRMSBIN:$SRMBIN
export PATH
case "$1" in
'start')
if [ ! -x $SRMBIN/srmadm ]; then
echo "Solaris Resource Manager *not* installed." \
"Missing srmadm command."
exit
fi

# Only bother if sched/SHR is loaded.
if [ ` $SRMBIN/srmadm ` != yes ]
then
#
# Usually this is because /etc/system doesn't have the usual
# set initclass="SHR"
# or at least a set extraclass="SHR"
#
echo "Solaris Resource Manager *not* loaded."
exit
else
echo "Enabling Solaris Resource Manager v1.0."
if [ ` $SRMBIN/srmadm show fileopen ` = yes ]; then
echo "SRM database file already open - stopping first."
limdaemon -k
sleep 2
srmadm set $ChargeOptionsOff
sync
srmadm set fileopen=n
$ECHO "SRM inactive"
fi
$ECHO "Starting SRM..."
fi

# Check the limconf file.
if [ ! -s $ETCSRMR/limconf ]; then
echo "SRM - file $ETCSRMR/limconf is missing " >&2
echo "SRM not started."
exit 1
fi

if [ ! -f "$ShareDb" ]; then
echo "SRM database '$ShareDb' not present - " \
"creating empty database"
if [ ! -d "$DATADIR" ]; then
mkdir "$DATADIR"
chmod 400 "$DATADIR"
chown root "$DATADIR"
chgrp root "$DATADIR"

```

```

fi
touch "$ShareDb" ||
{
echo "Failed to create '$ShareDb'" >&2
echo "SRM not started"
exit 1
}
chmod 400 "$ShareDb"
chown root "$ShareDb"
chgrp root "$ShareDb"
fi

CreateLnodes=0
if [ ! -s "$ShareDb" ]; then
$ECHO "SRM Warning: Using empty database" >&2
CreateLnodes=1
fi

$ECHO "SRM starting ... \c"

# Open Lnode file.
srmadm set -f "$ShareDb" fileopen=y
if [ $? != 0 ]; then
echo
echo "srmadm set -f $ShareDb failed" >&2
echo "SRM not started"
exit 1
fi

# Set SRM global options.
srmadm set $ChargeOptionsOn
if [ $? != 0 ]; then
echo
echo "srmadm set $ChargeOptionsOn failed" >&2
echo "SRM not completely enabled"
exit 1
fi

# Create if needed the other lnode.
liminfo other 2>/dev/null | grep "^Login name: *other " >/dev/null 2>&1
if [ $? -ne 0 ]; then
# If user "other" exists but has no lnode, create one.
limadm set cpu.shares=1 other 2>/dev/null
limadm set sgroup=root other 2>/dev/null
fi

# Create if needed, and set the lost lnode.
if [ x"$LostLnode" != x ]; then
liminfo "$LostLnode" 2>/dev/null | \
grep "^Login name: *$LostLnode " >/dev/null 2>&1
if [ $? -ne 0 ]; then
limadm set cpu.shares=1 "$LostLnode"
limadm set sgroup=root "$LostLnode"
fi
srmadm set lost="$LostLnode" ||
$ECHO "SRM - Warning: No user '$LostLnode' for lost lnode"
fi

# Create if needed, and set the idle lnode.
if [ x"$IdleLnode" != x ]; then

```

```

        liminfo "$IdleLnode" 2>/dev/null | \
grep "^Login name: *$IdleLnode " >/dev/null 2>&1
if [ $? -ne 0 ]; then
    limadm set cpu.shares=0 "$IdleLnode"
    limadm set sgroup=root "$IdleLnode"
fi
srmadm set idle="$IdleLnode" ||
$ECHO "SRM - Warning: No user '$IdleLnode' for idle lnode"
fi

# If creating SRM database, set up existing users.
if [ "$CreateLnodes" -eq 1 ]; then
    echo "SRM - creating user lnodes; may take a while"
    # We now want to catch any other users which were not found
    # on the filesystems. First we need to decide what the maximum
    # uid value we will create an l-node entry for in the database.
    # We choose less than the uid for 'nobody' so that we can try
    # and minimise the apparent size of the database (which is a sparse
    # file). If the user 'nobody' does not exist then we just have
    # to take our chances with using all possible uid values.
    # Unfortunately all this means that there are certain circumstances
    # where not all users will be taken into account.

MaxUID=`awk -F: "\\\$1==\\"nobody\\" { print \\$3 - 100 }" /etc/passwd`
if [ $? -eq 0 -a x"$MaxUID" != x ] ; then
    Cond="uid >= 0 && uid < $MaxUID && !flag.real"
else
    Cond="uid >= 0 && !flag.real"
fi
UIDS=`limreport "$Cond" '%d\n' uid | wc -l`
if [ $UIDS -gt 0 ]; then
    $ECHO "$UIDS other lnodes to be created " \
        "due to passwd entries"
    limreport "$Cond" 'limadm set -u sgroup=0:cpu.shares=1 %d\necho " uid %7d\r\c"\n' uid uid | sh

    echo
fi
fi

limdaemon $LimdaemonOptions

echo "Solaris Resource Manager v1.0 Enabled."
;;

'stop')
# SRM shutdown should be done as late as possible before
# filesystems are unmounted.
if [ -x $SRMBIN/srmadm ] && $SRMBIN/srmadm show fileopen > /dev/null
then
    limdaemon -k
    sleep 2
    srmadm set $ChargeOptionsOff
    srmadm set fileopen=n
    sync
    $ECHO "Solaris Resource Manager Disabled"
fi
;;

*)
echo "Usage: $0 {start|stop}"

```

```
;;
esac
```

## Script “pas de noeud limite” par défaut

Ce script crée le noeud limite dans le groupe d’ordonnement par défaut (`other` si cet utilisateur existe dans la table des mots de passe, `root` autrement) et transmet un message à l’administrateur système pour lui rappeler de transférer le nouveau noeud limite à l’endroit approprié dans la hiérarchie d’ordonnement.

```
#!/bin/sh
#
#ident  "@(#)nolnode.sh 1.6 98/10/28 SMI"
#
# Copyright (c) 1998 by Sun Microsystems, Inc.
# All rights reserved.
#
# Copyright 1995-1997 Softway Pty. Ltd.
#
# A script that called by the PAM module to create a lnode
#

PATH=/usr/srm/bin:/sbin:/bin export PATH
LOCALE=C export LOCALE
if [ "$DEBUG" = "true" ]
then
    exec >> /tmp/nolnodelog 2>&1
    echo
    date
    echo "Attempting to create lnode for $USER"
else
    exec > /dev/null 2>&1
fi

err=`limadm set -u cpu.shares=1 "$UID" 2>&1`
if [ $? -eq 0 ]
then
    cat <<-EOF | /usr/lib/sendmail root
    Subject: New lnode created for "$USER"

    Remember to change scheduling group and shares for
    "$USER".

    `limreport lname=="$USER"' 'Currently in group "%s" with %d shares\\n' sgroupname cpu.shares`
EOF
else
    cat <<-EOF | /usr/lib/sendmail root
    Subject: Could not create lnode for "$USER"

    after "$SERVICE" attempt on tty "$TTY", uid "$UID",
    rhost \ "$RHOST" \ ,
    limadm said \ "$err" \
EOF
    exit 1 # deny access
fi
# permit access
exit 0
```





# Glossaire

---

Plusieurs nouveaux concepts font leur apparition avec Solaris Resource Manager, et certains éléments chevauchent des concepts propres à d'autres parties de Solaris.

Les termes ci-après sont définis dans le but de simplifier le présent document et d'éviter la confusion.

<b>actif</b>	Se dit d'un noeud limite auquel des processus sont reliés (ou à ses descendants.) Un noeud limite ne peut être supprimé pendant qu'il est actif.
<b>administrateur</b>	Personne chargée de la maintenance du système. Solaris Resource Manager fournit des fonctions permettant à l'administrateur de déléguer certaines tâches sans avoir à céder des privilèges de superutilisateur. Voir aussi utilisateur administratif, superutilisateur, utilisateur uselimadm et "Administration déléguée", page 51.
<b>administrateur central</b>	Utilisateur racine (ou superutilisateur) du système. Le noeud limite racine est toujours le premier de l'arbre d'ordonnement. L'administrateur central est chargé des tâches globales de gestion des utilisateurs et des ressources, mais il peut déléguer certaines tâches à des utilisateurs ordinaires en leur accordant les privilèges administratifs requis. Habituellement, l'administrateur central détermine l'attribution des ressources aux groupes enfants du noeud limite racine et accorde des privilèges administratifs aux chefs de chacun de ces groupes, ce qui le libère de la majeure partie du fardeau administratif.
<b>arbre d'ordonnement</b>	Arborescence ayant à sa tête le noeud limite racine et comportant des références particulières à la relation père-enfant entre noeuds limites, l'attribution des parts d'UC et la façon dont l'ordonneur Solaris Resource Manager détermine les taux d'exécution des processus.

<b>ascendant</b>	Se dit d'un noeud limite qui est l'ascendant d'un autre si les références successives à l'attribut sgroup, à partir du premier noeud limite, renvoient à l'autre. Dans un tel cas, ce dernier est un descendant, ou membre, du premier.
<b>attribut</b>	Champs de données d'un noeud limite. Vu que tous les noeuds limites partagent la même structure interne, tous les utilisateurs partagent le même ensemble d'attributs. Les attributs peuvent être relatifs au système, à l'utilisateur (programmes en mode utilisateur uniquement) ou au domaine. Les divers types d'attribut diffèrent quant aux numéros de champ qui leur sont attribués. Les attributs système sont utilisés directement par le noyau, par exemple les variables numériques contrôlant des ressources comme les processus, la quantité de mémoire et les indicateurs qui gèrent les privilèges système au niveau du noyau. L'administrateur peut ajouter des attributs d'utilisateur en tout temps et modifier les attributs existants s'ils ne perturbent pas les programmes qui les utilisent. Les attributs de domaine ne sont pas déclarés dans le fichier source de configuration, car la déclaration d'un domaine le définit implicitement.
<b>attribut d'utilisateur</b>	Voir attribut.
<b>avis</b>	Message envoyé au démon <code>limdaemon(1MSRM)</code> de Solaris Resource Manager. Certains avis ont une signification spéciale pour <code>limdaemon</code> .
<b>base de données des noeuds limites</b>	Copie sur disque de tous les noeuds limites utilisés par Solaris Resource Manager, classés par UID.
<b>boucle de groupe</b>	Voir "Boucles de groupe", page 87.
<b>chef de groupe</b>	Noeud limite placé à la racine d'un groupe. Solaris Resource Manager permet d'accorder certains privilèges administratifs aux chefs de groupe pour confier, par exemple, à un directeur d'un service, l'entière responsabilité de la création et de la suppression de groupes et d'utilisateurs et l'attribution des ressources au sein de son service. Les chefs de groupe ne sont pas considérés comme membres de leur groupe.
<b>cumulatif</b>	Les ressources fixes et renouvelables peuvent être dotées d'un attribut d'usage cumulatif qui est l'entier de l'attribut d'usage correspondant dans le temps.

<b>décroissance</b>	Diminution périodique de l'usage d'une ressource renouvelable. Pour toutes les ressources (à l'exception de l'usage d'unité centrale), la décroissance est calculée d'après un taux fixe qui doit être soustrait de l'attribut d'usage sur une base régulière. Pour l'usage d'UC, une décroissance exponentielle (multiplicative) est utilisée.
<b>droit</b>	Quantité de temps d'UC attribuée à un utilisateur.
<b>enfant</b>	Tous les noeuds limites situés directement sous un autre dans l'arbre d'ordonnancement sont ses enfants. Un noeud limite est l'enfant d'un autre si la valeur de l'attribut <code>sgroup</code> du premier noeud est l'UID du second. Ce dernier est appelé père ou chef de groupe du premier.
<b>groupe</b>	Dans Solaris Resource Manager, terme désignant un groupe d'ordonnancement. Voir Groupe d'ordonnancement.
<b>groupe d'ordonnancement</b>	Solaris Resource Manager permet de classer tous les utilisateurs à l'aide d'une hiérarchie de groupes couvrant l'ensemble du système et reflétant la structure des services qui l'utilisent. L'expression "groupe d'ordonnancement" a été préférée à "groupe" afin d'éviter toute confusion avec le concept de groupe UNIX existant, et ce, même si ces groupes d'ordonnancement sont utilisés pour nombre d'autres tâches que l'ordonnancement. Les groupes Solaris Resource Manager n'ont aucun lien avec les groupes définis dans le fichier <code>/etc/group</code> de UNIX.  Un groupe d'ordonnancement à n'importe quel niveau de la hiérarchie peut être traité comme un utilisateur. Autrement dit, les limites de ressource attribuées à un groupe d'ordonnancement s'appliquent à l'usage net de tous les groupes et utilisateurs au sein de ce groupe.
<b>groupe de premier niveau</b>	Groupe ayant la racine comme chef de groupe.
<b>homologue</b>	Les homologues d'un noeud limite sont les autres noeuds du même groupe d'ordonnancement, à l'exception du père.
<b>indicateur</b>	Attribut similaire à une variable booléenne, mais qui peut avoir l'une des quatre valeurs suivantes : <code>set</code> , <code>clear</code> , <code>group</code> ou <code>inherit</code> . Solaris Resource Manager utilise des indicateurs pour contrôler les privilèges.
<b>inherit</b>	L'une des valeurs possibles d'un attribut d'indicateur. Lorsqu'un indicateur avec une valeur <code>inherit</code> immédiate est évalué, le même indicateur du noeud limite père est aussi évalué afin de déterminer

la valeur réelle. Ce processus est récursif. Si l'indicateur est réglé à `inherit` sur le noeud limite racine, la valeur finale est déterminée d'après la valeur par défaut. Le résultat de l'évaluation d'un indicateur est toujours `set` ou `clear`.

<b>limite</b>	Attribut numérique associé à un attribut d'usage. L'usage d'une ressource par un utilisateur est contrôlé afin d'éviter le dépassement des limites applicables à cette ressource. Il y a deux types de limite : stricte et variable. Une limite stricte cause l'échec des tentatives de consommation ou d'attribution des ressources si l'usage résultant entraînerait son dépassement. Une limite variable ne restreint généralement pas l'usage directement mais fournit plutôt un repère permettant d'informer l'utilisateur afin qu'il réduise son usage. Une limite de zéro est un cas spécial, correspondant à aucune limite.
<b>mode utilisateur</b>	Mode de fonctionnement sous lequel le code est exécuté par les programmes et processus habituels d'un système UNIX. Le mode noyau est utilisé par les appels système, les pilotes de périphérique et l'ordonnanceur de classe SYS. Certains composants de Solaris Resource Manager sont exécutés en mode utilisateur et d'autres en mode noyau.
<b>noeud limite</b>	Structure de taille fixe utilisée par Solaris Resource Manager pour la maintenance de toutes les données utilisateur requises en plus des données stockées dans la table des mots de passe. Elle est stockée sur disque dans le fichier du noeud limite et lue ou modifiée par le noyau au besoin. Il ne peut y avoir plus d'un noeud limite pour chaque UID. Différents comptes ayant la même UID utilisent le même noeud limite.
<b>noeud limite autre</b>	Si un compte existe pour un utilisateur nommé "autre" (other) et que ce compte a un noeud limite, celui-ci sera utilisé par défaut pour le père des noeuds limites créés par racine ou par les utilisateurs de <code>uselimadm</code> au moyen de la commande <code>limadm(1MSRM)</code> .
<b>noeud limite inactif</b>	Noeud limite dont le temps de processeur inutilisé est cumulé, ce qui peut être utile à des fins de comptabilité. Le nom d'utilisateur par défaut est <code>srmidle</code> et son UID est 41.
<b>noeud limite perdu</b>	Noeud limite spécial qui est utilisé lorsque l'appel système <code>setuid(2SRM)</code> ne parvient pas à relier un processus au noeud correspondant à l'UID cible de l'appel, habituellement parce que le noeud limite n'existe pas.
<b>noeud limite racine</b>	Noeud de l'UID 0. Ce noeud limite est à la racine de l'arbre des noeuds limites, et tous les autres noeuds sont ses membres.

<b>noyau</b>	Partie résidente du système d'exploitation. Le noyau prend en charge les appels système, les systèmes de fichiers et l'ordonnancement des processus. Solaris Resource Manager se compose de deux modules noyau et de plusieurs raccords, ainsi que de programmes utilisateur (non noyau) et de fonctions de bibliothèque.
<b>numéro de champ</b>	Emplacement de grille utilisé par les attributs précisés dans le fichier de configuration.
<b>orphelin</b>	Noeud limite n'ayant pas de noeud père, ce qui signifie que l'UID précisée dans l'attribut sgroup du noeud ne correspond pas à un noeud limite.
<b>parts</b>	Expression utilisée pour définir la proportion de droits à l'UC accordée à un noeud limite par rapport à son père et à ses homologues. Ce concept est semblable à celui des actions d'une entreprise ; ce qui importe n'est pas la quantité détenue, mais bien la proportion d'actions par rapport aux autres actionnaires.
<b>part attribuée</b>	Fraction des ressources d'UC disponibles qui serait accordée à un utilisateur à long terme avec une configuration donnée de hiérarchie d'arbre de noeuds limites, de parts et de noeuds limites actifs.
<b>part courante</b>	Voir Chapitre 6.
<b>père</b>	Chef de groupe d'un noeud limite dans l'arbre d'ordonnancement.
<b>relié</b>	Lorsqu'un utilisateur ouvre une session, son processus de module PAM se relie au noeud limite correspondant à son UID. Tout processus engendré par la suite est relié par défaut au même noeud limite. Le noeud limite auquel un processus est relié détermine les limites, les parts de ressources d'UC, les privilèges et d'autres éléments du processus.
<b>ressource fixe</b>	Ressource dont la quantité totale est limitée.
<b>ressource renouvelable</b>	Ressource continuellement disponible, par exemple, l'usage de l'unité centrale ou le temps de connexion.
<b>session Solaris Resource Manager</b>	Toute ouverture de session détectée par Solaris Resource Manager et qui nécessite une coopération entre Solaris Resource Manager et les divers programmes passerelles responsables de l'authentification des utilisateurs et des droits d'accès.

<b>sous-administrateur</b>	Chef de groupe ayant des privilèges administratifs sur les membres de son groupe. Un chef de groupe devient sous-administrateur lorsque son indicateur admin est activé, ce qui lui permet de créer et de supprimer des noeuds limites d'utilisateurs au sein de son groupe, de contrôler des ressources et l'attribution des privilèges dans son groupe et de déléguer des tâches administratives aux chefs de son groupe.
<b>superutilisateur</b>	Utilisateur connaissant le mot de passe racine. Les processus sont dotés de privilèges de superutilisateur lorsqu'ils sont exécutés avec une UID de 0.
<b>unités</b>	Quantité unitaire d'une ressource. Dans Solaris Resource Manager, les valeurs sont représentés selon trois types d'unité, soit : proportionnée, brute ou interne.
<b>usage</b>	Attribut numérique augmentant chaque fois que l'utilisateur consomme des ressources ou s'en voit attribué. L'usage des ressources fixes décroît chaque fois que la ressource est libérée. L'usage des ressources renouvelables diminue à chaque décroissance.
<b>utilisateur administratif</b>	Utilisateur ayant un noeud limite dont l'indicateur "admin" est activé. Un tel utilisateur peut créer, supprimer et modifier des noeuds limites au sein de son groupe d'ordonnement.
<b>utilisateur uselimadm</b>	Utilisateur dont l'indicateur <code>uselimadm</code> est défini sur le noeud limite, ce qui lui accorde les mêmes privilèges administratifs que le superutilisateur.