

Tuning système sous Solaris 2.x (SunOS 5.x)

Référence A31



une division de
Sun Microsystems France S.A.
Service Formation
143 bis, avenue de Verdun
92 442 Issy les Moulineaux Cedex
Tel 16 1 41 33 17 17
Fax 16 1 41 33 17 20

Sun Microsystems France S.A.
Siège social
13, av. Morane Saulnier - B.P. 53
78 142 VELIZY Cedex
Tel 16 1 30 67 50 00
Fax 16 1 30 67 53 00

Révision B, Avril 1998
Document non révisable

© 1993 Sun Microsystems, Inc.—Printed in the United States of America.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc. and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's Font Suppliers.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, [ALL OTHER SUN TRADEMARKS REFERRED TO IN THE PRODUCT OR DOCUMENT] are trademarks or registered trademarks of Sun Microsystems, Inc. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. [ATtribution of other third party trademarks mentioned significantly throughout product or documentation]. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark and product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Table des matières



Mise en oeuvre d'une politique de tuning..... 1-1

Introduction	1-2
Besoin.....	1-2
Connaissances préliminaires.....	1-2
Connaissance de l'environnement.....	1-2
Besoin.....	1-3
Performances attendues des applications	1-4
Impacts des divers composants	1-4
Qu'est-ce que le tuning ?.....	1-6
Algorithme général.....	1-6
Travail de l'administrateur.....	1-6
Analyse des résultats.....	1-8
Isolation des applications	1-8
Détection des ralentissements.....	1-8
Quand mettre en oeuvre une politique de tuning	1-10
Preliminaire.....	1-10
En exploitation.....	1-10
Définition de l'environnement.....	1-12
Site clients/serveur	1-12
Type de clients.....	1-12
Type de serveurs	1-12
Connaissance du matériel.....	1-12
Connaissance du logiciel.....	1-12
Capacités du matériel.....	1-16
Type de processeurs	1-16
Type de machines	1-16
Gamme des serveurs	1-16
Gamme des périphériques.....	1-16
Capacités des matériels	1-45
Les périphériques disques	1-58
Les types de bus SCSI.....	1-58
Les autres types incluant un protocole SCSI.....	1-58
Les périphériques disques	1-71
Capacités des logiciels.....	1-72
Les versions de systèmes d'exploitation traités.....	1-72
Les applicatifs liés au système d'exploitation.....	1-72
les versions d'OS couvertes	1-73
Performances	1-97
Techniques de surveillance.....	1-100
Isolation des applications	1-100
Algorithme de tuning.....	1-100
Modifications possibles.....	1-102
Logicielles et matérielles	1-102
Logicielles et matérielles	1-103

Mécanismes internes	2-1
Mécanismes internes	2-2
Pré-requis nécessaire	2-2
Mécanismes mis en oeuvre.....	2-2
Gestion du noyau.....	2-4
Fonctionnement interne	2-6
Temps d'exécution.....	2-6
Gestion des processeurs.....	2-8
Les interruptions.....	2-10
Implantation mémoire d'un processus.....	2-16
Cycle de vie d'un processus	2-18
Variables associées aux processus.....	2-20
Client/Serveur concurrent et Threads.....	2-24
Commutation de contexte.....	2-24
Les priorités	2-26
Gestion du swap.....	2-30
Gestion des accès disques	2-46
Types d'accès	2-46
Mécanismes internes	2-46
Description physique des disques.....	2-46
Description des types de systèmes de fichiers.....	2-46
Les systèmes de fichiers natifs	2-67
Interaction entre les produits	2-72
Applicatifs base de données.....	2-74
Gestion des bases de données.....	2-75
Applicatifs base de données.....	2-76
Place de la base dans le système.....	2-76
Les mécanismes internes Unix nécessaires	2-76
Fonctionnement d'une base de données	2-76
Utilisation des ressources disques.....	2-79
Mécanismes internes Unix.....	2-80
Mécanismes internes à la base de données	2-88
Gestion du réseau	2-90
Les bandes passantes.....	2-90
Les types de transports	2-90
Les charges des divers applicatifs	2-90
Les types de transports	2-95
Le protocole	2-102
Cas du client	2-102
Cas du serveur.....	2-102
Cachefs.....	2-116
Principes	2-116
Intérêt en matière d'applications réseau	2-116
Principes	2-117
Objectifs.....	2-117
Ressources.....	2-118



Application NFS.....	2-126
Application HTTP.....	2-130
Vue rapide sur les problèmes de développement.....	2-132
La surveillance	3-1
Présentation des outils de surveillance.....	3-2
Importance des outils	3-2
Les outils récapitulatifs	3-2
Les outils de surveillance quotidienne	3-2
Surveillance des applications	3-6
Surveillance de SunOS	3-6
Intervalle de surveillance.....	3-8
Les commandes Berkeley.....	3-10
Les autres commandes Berkeley.....	3-36
Les commandes SVR3	3-40
Les commandes de surveillance liées au développement	3-50
L'accounting	3-52
Qu'est-ce que l'accounting.....	3-52
Mise en œuvre de l'accounting.....	3-54
Les outils freewares	3-72
top.....	3-72
nfswatch	3-72
proctool.....	3-72
Adrian Monitor	3-72
Les autres outils.....	3-82
Les outils intégrés dans les logiciels.....	3-82
Les outils tierce-partie	3-82
Le protocole de remonté des informations	3-82
Rappels sur SNMP	3-84
Manager et agent.....	3-85
Produit tierce-partie.....	3-86
Détection des problèmes.....	4-1
Algorithme de tuning.....	4-2
Détection des problèmes.....	4-4
Dysfonctionnement d'un applicatif.....	4-4
Faibles performances.....	4-4
Dysfonctionnement d'un applicatif.....	4-6
Dysfonctionnement lié au noyau.....	4-6
Dysfonctionnement lié à l'applicatif	4-6
Faibles performances.....	4-12
CPU	4-12
Nombre de CPU.....	4-12

Attente sur les entrées/sorties	4-12
Mémoire	4-14
Processus	4-19
Swap.....	4-20
Swap par processus	4-20
Cache disque.....	4-29
Disque	4-33
Système de fichiers	4-34
Réseau.....	4-39
NFS.....	4-40
Bases de données.....	4-50
Serveur WEB.....	4-66

Les interventions	5-1
Introduction	5-2
Visualiser les valeurs des variables	5-2
Modifier les valeurs	5-2
Vérifier la modification	5-2
Action sur le noyau.....	5-4
Visualiser les valeurs des variables.....	5-4
Modifier la valeur	5-4
Vérifier la modification	5-4
Visualiser les valeurs des variables.....	5-6
Modifier le contenu de /etc/system.....	5-17
Modifier la valeur des paramètres	5-19
Affectation d'une valeur à un paramètre	5-19
Affectation d'une valeur à une variable d'un module	5-19
Cas des IPC	5-20
Action sur les processus.....	5-26
Visualiser les valeurs des variables	5-26
Modifier la valeur	5-26
Vérifier la modification	5-26
Visualiser les valeurs des priorités.....	5-28
Modifier la valeur	5-31
Action sur les disques.....	5-32
Visualiser les valeurs des variables.....	5-32
Modifier la valeur	5-32
Vérifier la modification	5-32
Visualiser les valeurs des variables.....	5-33
Action sur le réseau	5-40
Visualiser les valeurs des variables.....	5-40
Modifier la valeur	5-40
Vérifier la modification	5-40
Action sur les utilisateurs	5-44



Visualiser les valeurs des variables	5-44
Modifier la valeur	5-44
Vérifier la modification	5-44
Etude de cas	6-1
Introduction	6-2
Cas de la machine desktop	6-4
Description de la machine	6-4
Les choix liés au système d'exploitation.....	6-4
Les choix liés aux applicatifs	6-4
Cas du serveur générique	6-14
Description de la machine	6-14
Les choix liés au système d'exploitation.....	6-14
Les choix liés aux applicatifs	6-14
Cas du serveur de calcul	6-22
Description de la machine	6-22
Les choix liés au système d'exploitation.....	6-22
Les choix liés aux applicatifs	6-22
Cas du serveur NFS	6-30
Description de la machine	6-30
Les choix liés au système d'exploitation.....	6-30
Les choix liés aux applicatifs	6-30
Cas du serveur de base de données	6-56
Description de la machine	6-56
Les choix liés au système d'exploitation.....	6-56
Les choix liés aux applicatifs	6-56
Cas du serveur WEB.....	6-70
Description de la machine	6-70
Les choix liés au système d'exploitation.....	6-70
Les choix liés aux applicatifs	6-70
Annexe A : Scripts et fichiers Réseau	A-1
Fichier S69inet.....	A-2
Snoop	A-4
Annexe B : Paramètres de configuration.....	B-1
Index.....	I-1

Mise en oeuvre d'une politique de tuning



Objectifs

Les sujets couverts par ce chapitre seront les suivants :

- qu'est-ce que le tuning ?
- quand mettre en oeuvre une politique de tuning,
- définition de l'environnement,
- les capacités du matériel,
- les capacités des logiciels,
- les techniques de surveillance,
- les modifications possibles.



Introduction

Besoin

Amélioration des performances des machines

Connaissances préliminaires

L'environnement matériel

L'environnement logiciel

Connaissance de l'environnement

Les types de services proposés

Introduction

Besoin

- Amélioration des performances des machines

Le tuning n'a pas pour seul but l'amélioration des performances des machines, il permet aussi la supervision des charges actuelles, et propose une prévision des charges futures.

Il permet de vérifier l'adéquation du matériel et du logiciel en fonction des besoins de chaque application.

Connaissances préliminaires

- L'environnement matériel

Le tuning nécessite une connaissance préalable des capacités matérielles des machines disponibles.

- L'environnement logiciel

Il nécessite aussi une connaissance des mécanismes « internes » des couches en présence pour pouvoir intervenir sur chaque entité. Il demande des connaissances préalables sur le système, le réseau et les diverses applications mises en oeuvre dans l'exploitation quotidienne.

Connaissance de l'environnement

- Les types de services proposés

Le support couvre les études de performances d'un serveur de calcul, d'un serveur d'espace disque (local et NFS), d'un serveur de base de données et d'un serveur WEB.



Introduction

Performances attendues des applications

Impacts des divers composants

Utilisateurs

Applications

Configuration du noyau

Configuration du système

Réseau

Introduction

Performances attendues des applications

Dans le milieu informatique, la performance d'un équipement comprend sa capacité à effectuer une tâche en consommant un minimum de ressource et avec le temps de réponse le meilleur.

Impacts des divers composants

Les facteurs intervenant sur le temps de réponse sont :

- Les utilisateurs

La façon dont le personnel utilise les systèmes et le réseau est typique de chaque site. Il est nécessaire de comprendre leurs habitudes pour leur proposer des performances en adéquation avec leurs besoins.

- Les applications

Il est nécessaire de connaître la liste des applications utilisées de façon quotidienne, ainsi que les ressources dont elles ont besoin tant localement que sur le réseau.

- La configuration du noyau

Il est nécessaire de configurer le noyau en fonction des besoins précis des applicatifs des machines serveurs.

- La configuration du système

La configuration du noyau doit être complétée par la configuration de tout le système, c'est-à-dire l'espace mémoire, la zone de swap, et la configuration des disques.

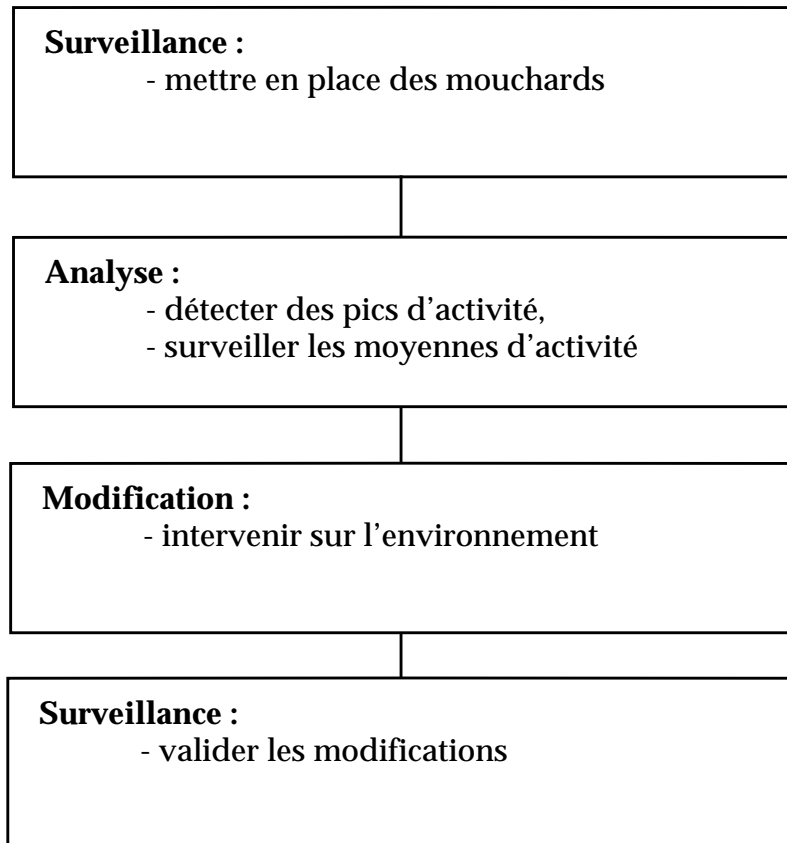
- Le réseau

Le réseau prend aussi une part importante dans la configuration d'un site client serveur. Nous veillerons donc à étudier spécifiquement cette ressource.



Qu'est-ce que le tuning ?

Algorithme général



Travail de l'administrateur

Tâche quotidienne

Qu'est-ce que le tuning ?

Algorithme général

■ Surveillance

Pour mettre en oeuvre une politique de tuning, il est nécessaire de disposer de données caractéristiques des charges des diverses applications. Le premier travail de l'administrateur sera donc de valider des mouchards pour relever un maximum d'informations.

■ Analyse

Une fois les données récupérées, il va être nécessaire de les analyser. Cette analyse repose sur une connaissance de certains mécanismes internes aux systèmes et aux applications.

■ Modification

A la suite de l'analyse, l'administrateur modifiera des paramètres liés aux équipements de son site.

■ Surveillance

La surveillance devra reprendre tant pour valider les modifications effectuées que pour anticiper sur des charges futures.

Travail de l'administrateur

Le travail de tuning de l'administrateur se doit d'être quotidien, tant sur le relevé d'informations que sur l'analyse des résultats.



Analyse des résultats

Isolation des applications

Se prémunir sur l'interaction entre applications

Prendre en compte toutes les composantes des applications

Détection des ralentissements

Les composantes de l'environnement

- le disque
- le système et l'application
- le réseau

Analyse des résultats

Isolation des applications

La première tâche va consister à se prémunir de l'interaction entre applications. De plus, chaque application devra être détaillée pour isoler les parties purement locales au serveur, les parties dépendant du client et le poids de l'interaction réseau lors du dialogue.

Détection des ralentissements

- Les composantes de l'environnement

Le premier travail va consister à mettre en évidence le goulet d'étranglement subit par le système. Les éléments suivants seront donc étudiés :

- le disque
- le système et l'application
- le réseau

Chacune de ces composantes interagissent les unes sur les autres ce qui rend plus complexe la détermination du problème de base. Au vue des divers temps de réponse des 3 composantes citées ci-dessus, il est nécessaire de commencer par l'étude des temps de réponse du disque qui est souvent l'élément le plus lent.



Quand mettre en oeuvre une politique de tuning

Preliminaire

Définition des besoins

Quantification des charges possibles

Fonctionnement d'une application

En exploitation

Conformité aux besoins

Amélioration des performances

Prévoir les montées en charge

Quand mettre en oeuvre une politique de tuning

Préliminaire

■ Définition des besoins

La connaissance de l'environnement est fondamentale pour mener à bien une politique de tuning. Cette dernière doit être entreprise le plus rapidement possible sur le site pour pouvoir avoir des éléments comparatifs et pour intervenir au plus tôt si une baisse de performance est détectée.

■ Quantification des charges possibles

Le tuning devrait commencer lors du choix de l'équipement et lors de la configuration initiale de ce dernier. Il permet d'adapter les applications aux équipements et aux utilisations prévues par le cahier des charges.

■ Fonctionnement d'une application

Une des composantes du tuning est aussi la résolution de problèmes système permettant à une application de fonctionner dans un environnement précis.

En exploitation

Il est, bien entendu nécessaire de continuer cette tâche lors de l'exploitation quotidienne. Elle permettra d'obtenir les meilleures performances des équipements et de prévoir les montées en charge ou l'installation de nouvelles applications.



Définition de l'environnement

Site clients/serveur

Travail sur le client

Travail sur le serveur

Type de clients

Importance du graphique

Type de serveurs

Travail principal

Types de services

Connaissance du matériel

Les plates-formes

Connaissance du logiciel

Les versions de systèmes d'exploitation

Les versions des applications

Définition de l'environnement

Site clients/serveur

L'environnement de base du site est important. S'il est de type « site centralisé », la surveillance a lieu sur la machine centrale. S'il est de type « client/serveur », il est nécessaire de prendre en compte cette composante pour détecter les goulets d'étranglement de l'ensemble.

Il sera donc nécessaire d'effectuer une analyse tant sur le serveur que sur le client. Le travail principal se centralisera toute fois sur le serveur.

Type de clients

- Importance du graphique

Actuellement, les besoins liés aux clients sont essentiellement basés sur la charge réseau induite par les applications et par les potentialités graphiques des postes clients.

Type de serveurs

- Travail principal

Le travail le plus important aura lieu sur cette machine. Il est recommandé de pouvoir isoler les services pour effectuer un travail le plus pertinent.

- Types de services

Chaque service proposé sera analysé séparément. Certains étant antagonistes, une répartition des charges sur plusieurs machines peut être souhaitable.



Définition de l'environnement

Site clients/serveur

Travail sur le client

Travail sur le serveur

Type de clients

Importance du graphique

Type de serveurs

Travail principal

Types de services

Connaissance du matériel

Les plates-formes

Connaissance du logiciel

Les versions de systèmes d'exploitation

Les versions des applications

Définition de l'environnement

Connaissance du matériel

- Les plates-formes

Il est nécessaire de connaître les types de matériels dont dispose le site ainsi que les performances propres de ces matériels (type de processeurs, types de contrôleurs, la mémoire présente), pour évaluer les performances brutes de la machine et tirer le meilleur partie du matériel.

Connaissance du logiciel

- Les versions de systèmes d'exploitation

La connaissance des versions des systèmes d'exploitation, et des produits associés (les gestionnaires d'espace disque, de systèmes de fichiers) est fondamentale pour modifier au mieux les caractéristiques du système. Chaque version de produit possède des spécificités qui lui sont propres.

- Les versions des applications

Il en est de même pour les logiciels (base de données, etc.) portés par le serveur ainsi que les versions des protocoles utilisés entre le client et le serveur.



Capacités du matériel

Type de processeurs

Type de machines

Gamme des serveurs

Les serveurs Ultra 1

Les serveurs Enterprise

Starfire Enterprise 10000

Gamme des périphériques

SSA

Sun StorEdge A3000

Sun StorEdge A5000

Capacités du matériel

Type de processeurs

Nous allons commencer par étudier les types de processeurs disponibles sur les serveurs.

Type de machines

Les implémentations des serveurs seront ensuite détaillées.

La gamme des serveurs

- Les serveurs Ultra 1

Le serveur Ultra™ Entreprise™ 1, Ultra™ Entreprise™ 150 et 450 seront étudiés.

- Les serveurs Enterprise

Les serveurs Ultra™ Entreprise™ 3000, Ultra™ Entreprise™ 4000, Ultra™ Entreprise™ 5000, et Ultra™ Entreprise™ 6000 seront étudiés.

- Starfire Enterprise 10000

Cette implémentation sera détaillée.

La gamme des périphériques

Nous étudierons les types de connexions pour les périphériques disques, ainsi que les périphériques suivants : SSA, A 3000 et A 5000.

Capacités du matériel

Type de processeurs

Super Sparc

Hyper Sparc

Ultra Sparc



Machine multi-processeurs

Equilibrage de la mémoire

Zone cache

Capacités du matériel

Type de processeurs

■ Super Sparc

Ce processeur est disponible sur les machines d'architecture sun4m et sur les premières machines sun4d. Ses caractéristiques sont les suivantes :

- processeur peu rapide (85 M Hz),
- effectue 4 instructions en 1 coup d'horloge,
- cache interne de 1 à 2 M octets.

Il est adapté aux stations de travail de type clientes qui n'exécutent qu'une seule application (utilisation maximale de la zone cache).

■ Hyper Sparc

Ce matériel est disponible sur les machines de type sun4d (2000E), ses caractéristiques sont les suivantes :

- processeur plus rapide (150 M Hz),
- effectue 1 instructions en 1 coup d'horloge,
- cache interne de 36 à 512 K octets.

Ce processeur est adapté à un travail de serveur, où les contextes switches vont être importants.

■ Ultra Sparc

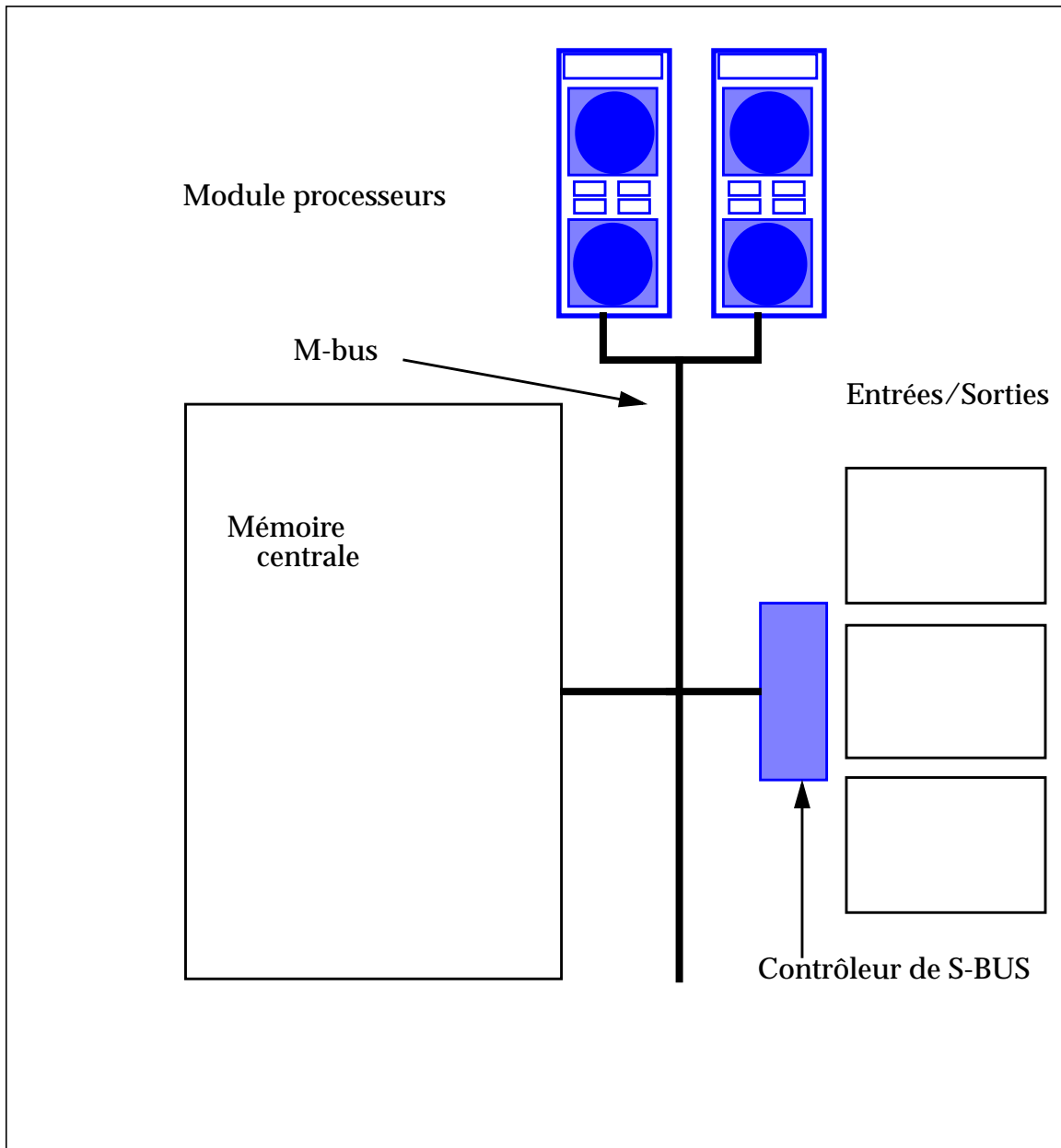
Ce matériel est disponible sur les machines de type sun4u. Il possède les caractéristiques suivantes :

- processeur plus rapide (143 à 250 M Hz),
- 64 bits interne,
- dispose d'un mode burst, nécessite 1 M octets de cache pour obtenir de meilleurs performances.

Capacités du matériel

Type de machines

Schéma structurel des serveurs de la gamme sun4m



Capacités du matériel

Type de machines

- Schéma structurel des serveurs de la gamme sun4m

Ces matériels possèdent :

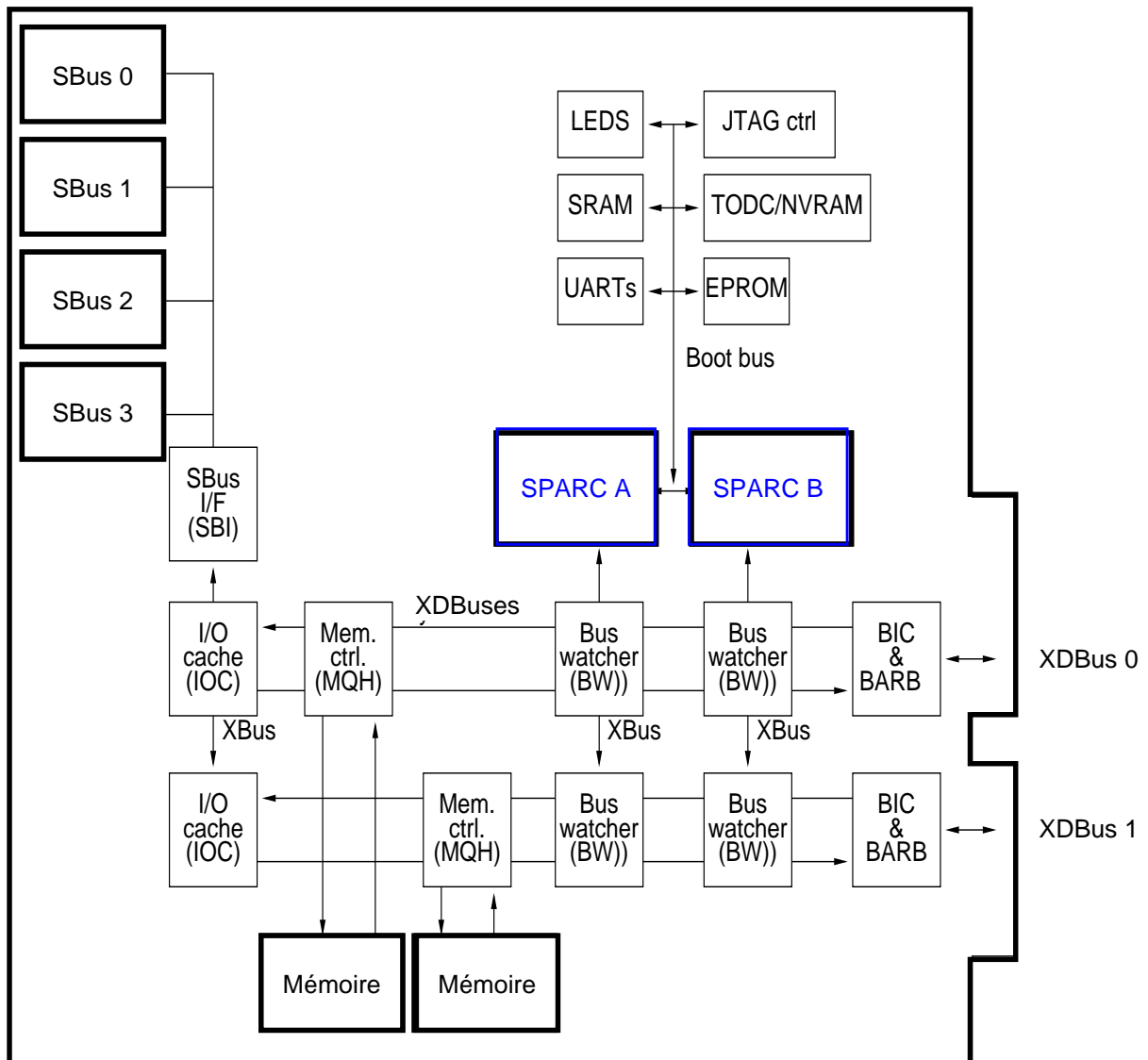
- une carte système unique,
- un ou des processeurs de type Super Sparc,
- un bus interne M-Bus de 64 bits séquenté entre 40 et 50 M Hz, en fonction des implémentations,
- un bus de gestion des entrées/sorties (S-Bus) séquenté entre 20 et 25 M Hz.



Capacités du matériel

Type de machines

Schéma structurel des serveurs de la gamme sun4d



Capacités du matériel

Type de machines

- Schéma structurel des serveurs de la gamme sun4d

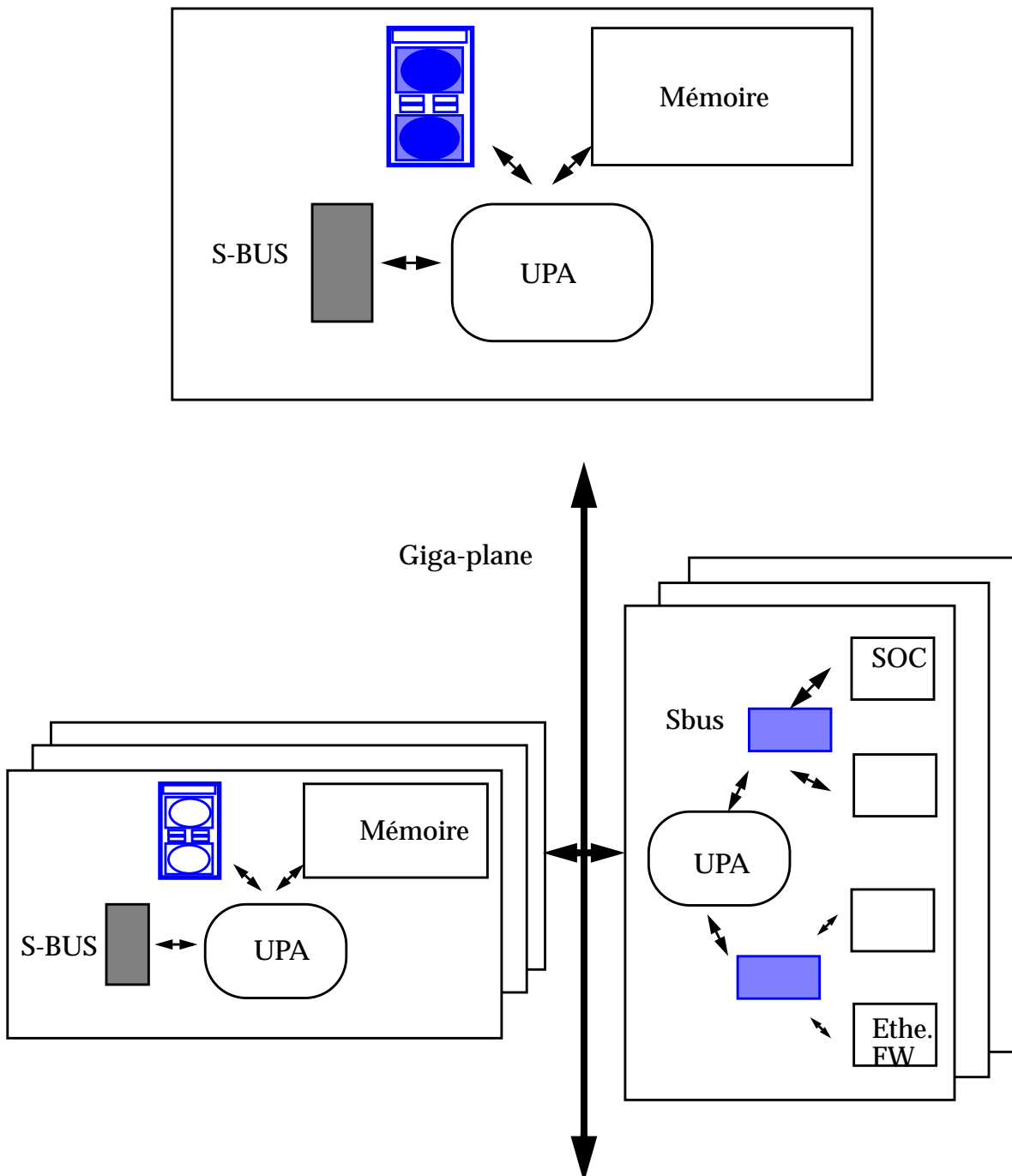
Ces matériels possèdent :

- plusieurs cartes système,
- un ou des processeurs de type Hyper Sparc (2000 E),
- un bus interne XD-Bus de 64 bits séquencé à 40 M Hz,
- un bus de gestion des entrées/sorties (S-Bus) séquencé à 20 M Hz.

Capacités du matériel

Type de machines

Schéma structurel des serveurs de la gamme sun4u



Capacités du matériel

Type de machines

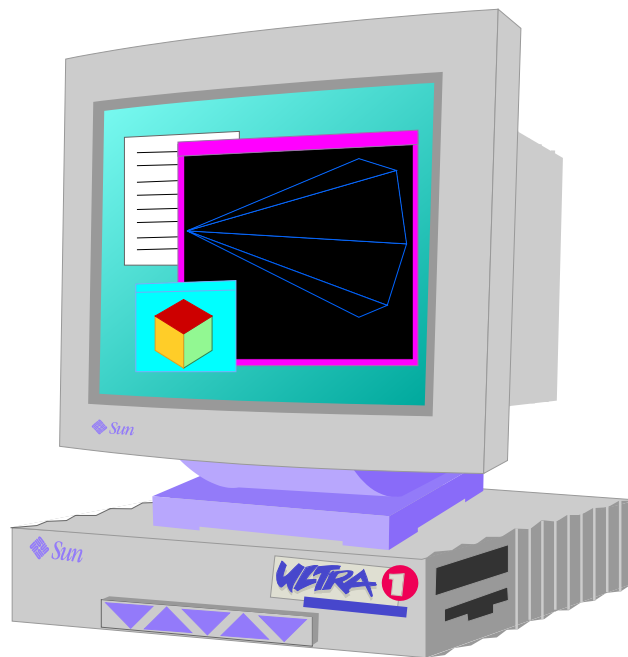
- Schéma structurel des serveurs de la gamme sun4u

Ces matériels possèdent :

- plusieurs cartes système,
- un ou des processeurs de type Ultra Sparc,
- un bus interne XD-Bus de 64 bits séquencé à 40 M Hz,
- un bus de gestion des entrées/sorties (S-Bus) séquencé à 20 M Hz.
- le bus Giga-plane propose un débit de 2,5 G octets/s et est séquencé à 83.3 M Hz.
- un contrôleur Intelligent Fast Wide SCSI-2 20 Mo/s,
- deux connexions FiberChannel à 50 Mo/s, permettant par exemple des connexions performantes vers un sous-système disque tel que le SPARCStorage Array. Ces éléments sont répartis sur deux bus SBus bufferisés, permettant chacun des débits de 200 Mo/s crête en mode burst.
- certaines versions disposent de 2 emplacements **PCI** (66 MHz/64 bits) qui incluent :
 - un contrôleur 10/100bT (Fast Ethernet),
 - un contrôleur Intelligent Fast Wide SCSI-2 20 Mo/s,
 - sur un contrôleur standard PCI à 33 MHz.

Capacités du matériel

Ultra™ 1 and Ultra™ Enterprise™ 1





Capacités du matériel

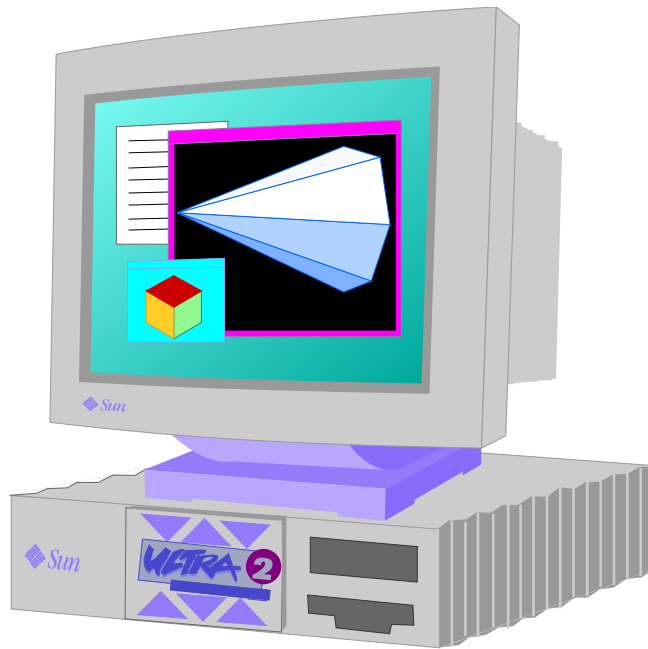
Ultra™ 1 and Ultra™ Enterprise™ 1

Cette gamme de machines sont basées sur des processeurs UltraSPARC cadencés à 143 MHz, 167 MHz ou 200 MHz :

- mémoire : de 32 Mo à 1 Go,
- 10 Mb/sec IEEE 802.3 Ethernet paire torsadée (10 Base-T) sur le modèle 170, et 100-10 Mb/s (100 Base-T) pour la série Creator ou interfaces AUI (en option),
- Fast, 10 MB/sec single-ended SCSI-2 pour le modèle 170, et Fast Wide SCSI-2 à 20 Mo/s pour les autres modèles Ultra 1,
- deux slots d'extension Sbus de largeur 64 bits réels (hormis l'accélérateur graphique),
- deux disques internes 3,5 " 2.1 Go,
- CD-ROM SunCD 12 interne,
- lecteur de disquette 3.5" interne (en option),
- jusqu'à 42 Go de capacité disque externe en utilisant les "desktops storage pack". Supporte également les lecteurs QIC, 8mm 14 Go et 4mm DAT.

Capacités du matériel

Ultra™ 2 and Ultra™ Enterprise™ 2



Capacités du matériel

Ultra™ 2 and Ultra™ Enterprise™ 2

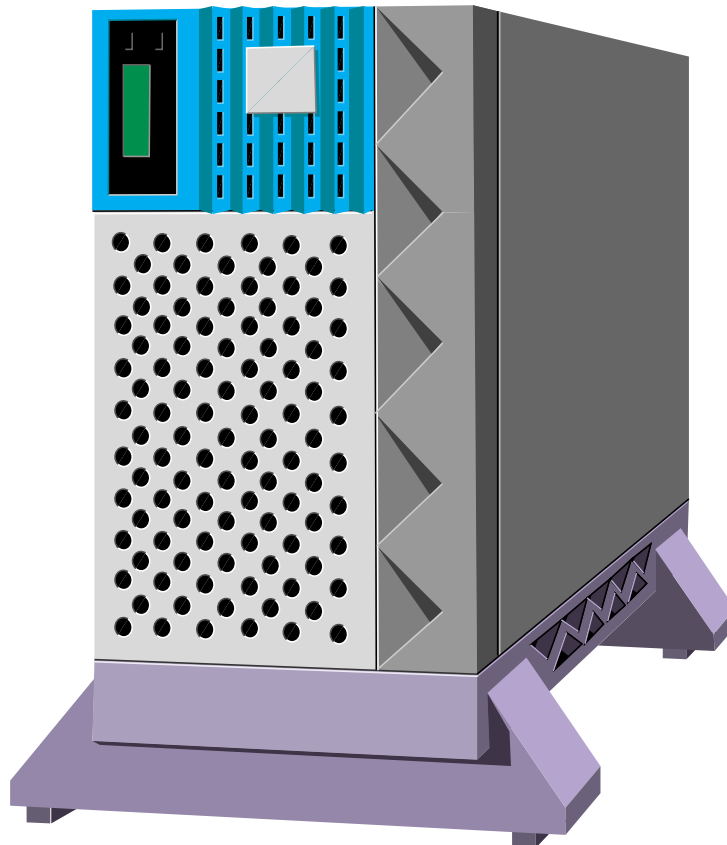
Les systèmes Ultra 2 constituent la nouvelle gamme de stations de bureau multiprocesseurs basés sur le processeur *UltraSPARC*. Ils se déclinent en plusieurs modèles à base de processeurs UltraSPARC cadencés à 167, 200 ou 300 MHz :

- modèle 2170 : bi-processeur à 167 MHz, 512 Ko de cache, 4 slots SBus, 2 slots UPA, Fast Ethernet 100 Mb/s, Fast Wide SCSI 20 Mo/s,
- modèle 1200 : mono-processeur à 200 MHz, 1 Mo de cache, 4 slots SBus, 2 slots UPA, Fast Ethernet 100 Mb/s, Fast Wide SCSI 20 Mo/s,
- modèle 2200 : bi-processeur à 200 MHz, 1 Mo de cache, 4 slots SBus, 2 slots UPA, Fast Ethernet 100 Mb/s, Fast Wide SCSI 20 Mo/s,
- modèle 1300 : mono-processeur à 300 MHz, 2 Mo de cache, 4 slots SBus, 2 slots UPA, Fast Ethernet 100 Mb/s, Fast Wide SCSI 20 Mo/s,
- modèle 2300 : mono-processeur à 300 MHz, 2 Mo de cache, 4 slots SBus, 2 slots UPA, Fast Ethernet 100 Mb/s, Fast Wide SCSI 20 Mo/s,
- mémoire de 128 Mo à 2 Go.



Capacités du matériel

Ultra™ Enterprise™ 150



Capacités du matériel

Ultra™ Enterprise™ 150

Ce serveur se présente sous un packaging au format tour (26.4 x 68.6 x 57.5 cm³). Ce modèle est prévu pour intégrer un grand nombre d'éléments :

- 1 processeur à 167 MHz, 512 Ko de cache,
- de 32 Mo à 1 Go de mémoire centrale,
- jusqu'à 25 Go de disques en interne (12 disques hot plug),
- jusqu'à 349 Go de disques en externe.

La bande passante, point à point, dans le crossbar UPA atteint 1,3 Go/s, lorsqu'il est cadencé à 83.3 Mhz, synchrone des CPU à 167 Mhz.

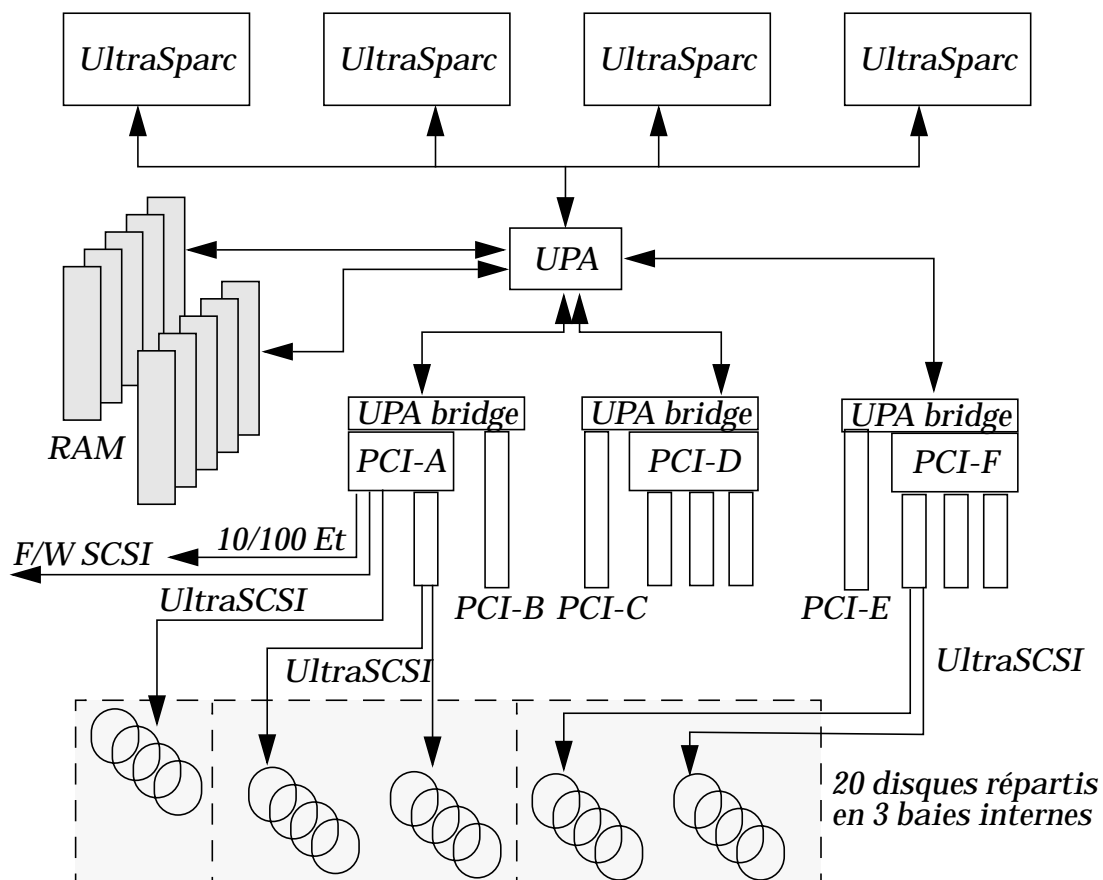
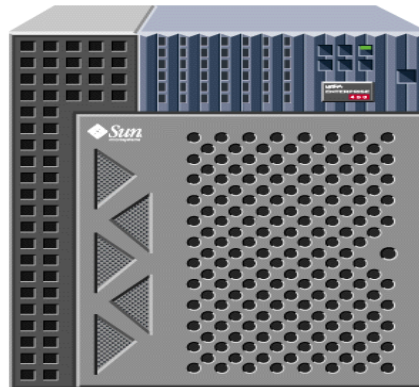
Connectivité

Le serveur Enterprise 150 est équipé, en standard :

- d'un CD-Rom quadruple vitesse,
- d'un lecteur de disquettes 3"5,
- d'un lecteur DAT 4mm en option.

Capacités du matériel

Ultra™ Enterprise™ 450



Capacités du matériel

Ultra™ Enterprise™ 450

Comprenant de un à 4 processeurs UltraSPARC, cadencés à 300 Mhz, ce serveur offre une capacité mémoire maximum de 4 Go, une capacité interne de 20 disques de 4.2 Go, et six bus PCI autorisant un débit d'entrées/sorties de 600 Mo/sec.

L'interconnexion entre processeurs, mémoire et canaux d'entrées/sorties est basée par un crossbar UPA. Ce crossbar permet de délivrer un flot de données en parallèle à un débit pouvant atteindre 1.6 Go/sec. Cette architecture, permet de réduire les temps de latence et d'optimiser l'utilisation des ressources du système. Les performances restent donc homogènes, même lors d'une montée en charge.

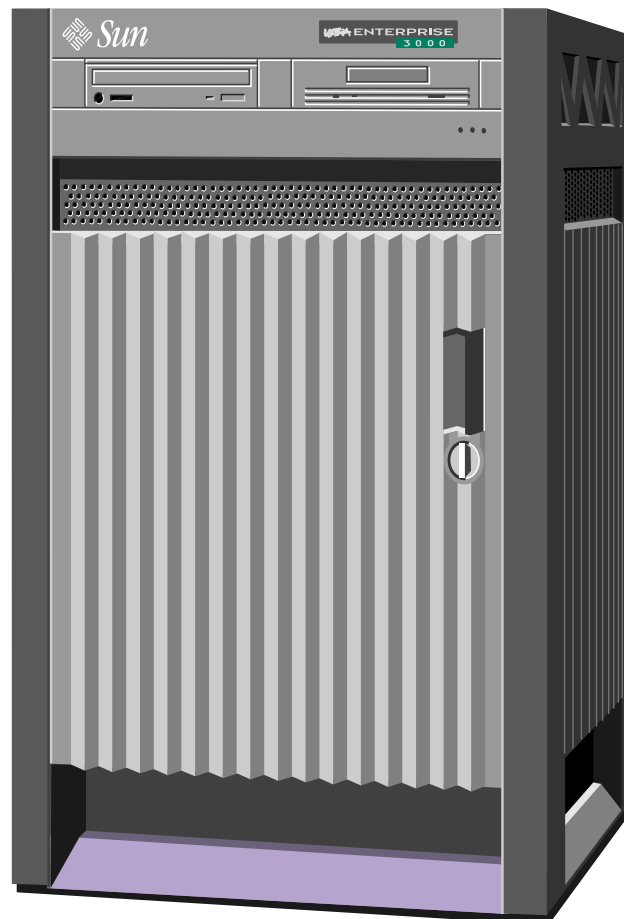
Caractéristiques

- 6 bus PCI supportant 3 Slots PCI 64 bits/66 Mhz, 3 slots PCI 32 bits/33 Mhz, et 4 slots PCI 32 ou 64 bits/33 Mhz,
- 1 port Ethernet/Fast Ethernet avec connecteurs RJ45 100 Base T et MII,
- 1 port F/W SCSI-2 pouvant supporter jusqu'à 4 lecteurs de bandes externes,
- 2 ports série [synchrone 50 à 384 Kbauds, asynchrone 50 à 460.8 Kbauds],
- 1 lecteur de CDROM interne,
- 5 Bus Ultra SCSI-3 40 Mo/s,
- jusqu'à 20 disques internes "hot plug" de 4.2Go/7200 rpm, également répartis sur les 5 bus UltraSCSI-3.
- jusqu'à 6 To de disques en externe.



Capacités du matériel

Ultra™ Enterprise™ 3000



Capacités du matériel

Ultra™ Enterprise™ 3000

Le serveur Enterprise 3000 offre 4 emplacements pour des cartes systèmes et/ou des cartes d'entrées/sorties, 6 processeurs au maximum.

Le serveur se présente sous un packaging au format tour (43 x 60 x 65 cm³). Ce modèle est prévu pour intégrer un grand nombre d'éléments, et offre :

- de 1 à 6 processeurs,
- de 64 Mo à 6 Go de mémoire centrale,
- jusqu'à 40 Go de disques en interne (10 disques),
- jusqu'à 2 To de disques en externe,
- jusqu'à 3 alimentations.

C'est une machine à architecture SMP, c'est-à-dire une architecture MultiProcesseurs Symétrique, optimisée par le système d'exploitation SOLARIS.

Les processeurs mis en oeuvre dans le serveur sont de type UltraSPARC cadencés à 167 MHz (1 Mo de cache) ou 250 MHz (1 Mo ou 4 Mo de cache), la plate-forme peut recevoir un maximum de 6 processeurs.

Fonctionnalités de sécurisation

Ses composants sont hot-plugs et permettent donc des raccordements à chaud.



Capacités du matériel

Ultra™ Enterprise™ 4000





Capacités du matériel

Ultra™ Enterprise™ 4000

Le serveur Enterprise 4000 offre 8 emplacements pour des cartes systèmes et/ou des cartes d'entrées/sorties, 14 processeurs maximum.

Ce serveur se présente sous un packaging au format compact (50 x 56 x 34 cm³). Ce modèle est prévu pour intégrer un grand nombre d'éléments, et offre :

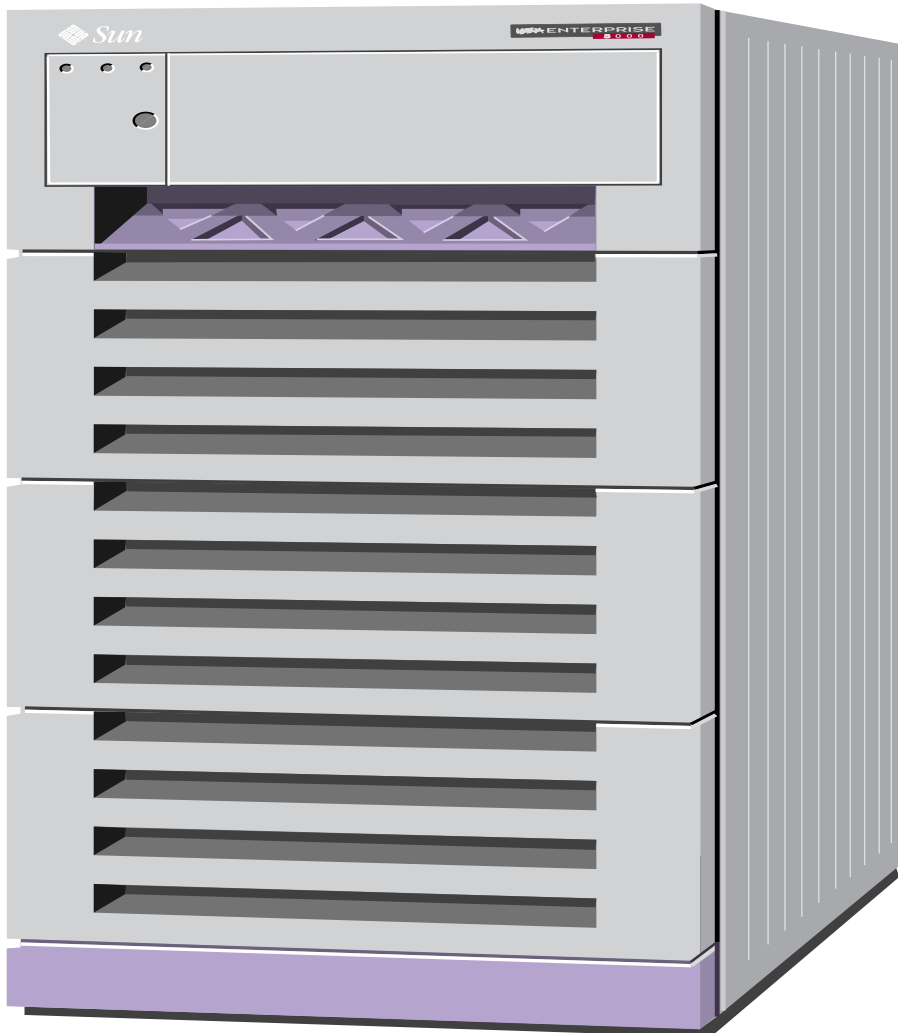
- de 1 à 14 processeurs,
- de 64 Mo à 14 Go de mémoire centrale,
- jusqu'à 16 Go de disques en interne (8 disques),
- jusqu'à 4 To de disques en externe,
- jusqu'à 4 alimentations.

C'est une machine à architecture SMP, c'est-à-dire une architecture MultiProcesseurs Symétrique.

Les processeurs mis en oeuvre dans le serveur sont de type UltraSPARC cadencés à 167 MHz (1 Mo de cache) ou 250 MHz (1 Mo ou 4 Mo de cache), la plate-forme peut recevoir un maximum de 14 processeurs.

Capacités du matériel

Ultra™ Enterprise™ 5000





Capacités du matériel

Ultra™ Enterprise™ 5000

Le serveur Enterprise 5000 se positionne comme un serveur d'entreprise offrant 8 emplacements pour des cartes systèmes et/ou des cartes d'entrées/sorties, 14 processeurs maximum.

Ce serveur se présente sous un packaging au format d'une armoire rack 56" (77 x 99 x 143 cm³). Ce modèle est prévu pour intégrer un grand nombre d'éléments, et offre :

- de 1 à 14 processeurs,
- de 64 Mo à 14 Go de mémoire centrale,
- jusqu'à 216 Go de disques en interne (3 SPARCstorage Arrays),
- jusqu'à 6 To de disques en externe,
- jusqu'à 4 alimentations.

Les processeurs mis en oeuvre dans le serveur sont de type UltraSPARC cadencés à 167 MHz (1 Mo de cache) ou 250 MHz (1 Mo ou 4 Mo de cache), la plate-forme peut recevoir un maximum de 14 processeurs.

C'est une machine à architecture SMP, c'est-à-dire une architecture MultiProcesseurs Symétrique.



Capacités des matériels

Ultra™ Enterprise™ 6000



Capacités des matériels

Ultra™ Enterprise™ 6000

Le serveur Enterprise 6000 se positionne comme un serveur d'entreprise offrant 16 emplacements pour des cartes systèmes et/ou des cartes d'entrées/sorties, 30 processeurs maximum.

Ce serveur se présente sous un packaging au format d'une armoire rack 56" (77 x 99 x 143 cm³). Ce modèle est prévu pour intégrer un grand nombre d'éléments, et offre :

- de 2 à 30 processeurs,
- de 64 Mo à 30 Go de mémoire centrale,
- jusqu'à 162 Go de disques en interne (2 SPARCstorage Arrays),
- jusqu'à 10 To de disques en externe,
- jusqu'à 8 alimentations.

Les processeurs mis en oeuvre dans le serveur sont de type UltraSPARC cadencés à 167 MHz (1 Mo de cache) ou 250 MHz (1 Mo ou 4 Mo de cache), la plate-forme peut recevoir un maximum de 30 processeurs.

C'est une machine à architecture SMP, c'est-à-dire une architecture MultiProcesseurs Symétrique.



Capacités des matériels

Ultra™ Enterprise™ 10 000



Capacités des matériels

Ultra™ Enterprise™ 10 000

Le serveur Enterprise 10000 est adapté au déploiement d'applications critiques conséquentes dans un environnement réseau.

Basé sur des technologies de type mainframes (partitionnement en machine virtuelles), l'Enterprise 10000 permet d'adresser l'ensemble des besoins liés aux applications critiques tout en conservant une grande faculté d'évolution et de disponibilité de services.

Ce serveur se présente sous un packaging au format d'une armoire rack (127 x 99 x 180 cm³). Ce modèle est prévu pour intégrer un grand nombre d'éléments, et offre :

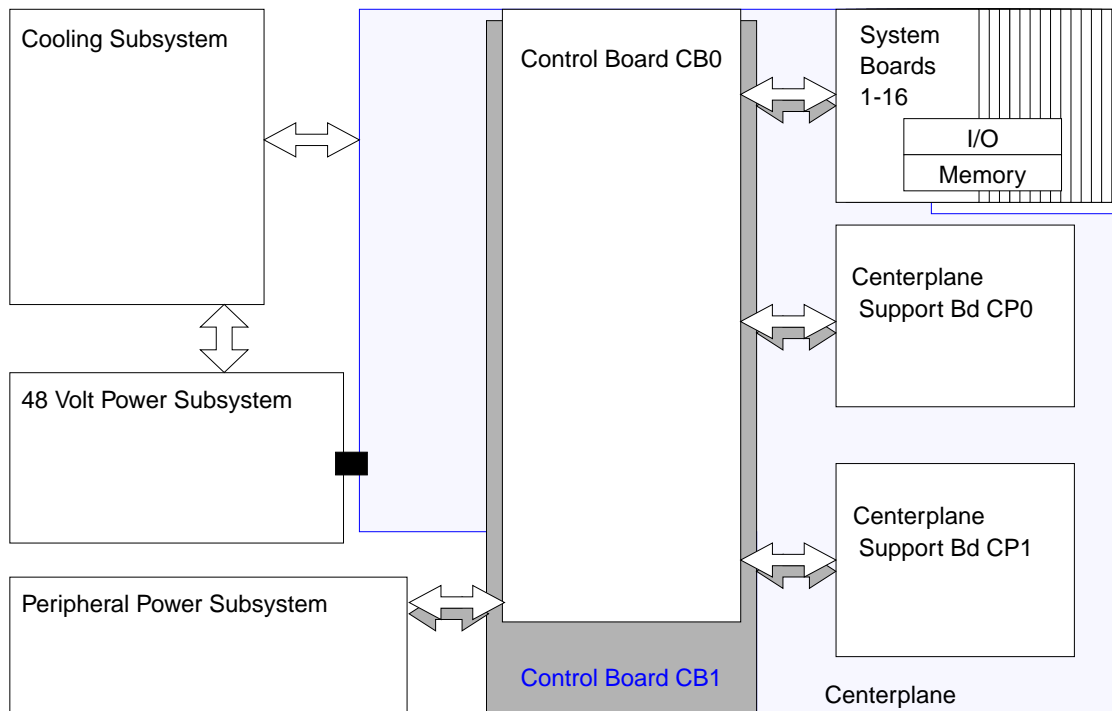
- de 16 à 64 processeurs,
- de 2 Go à 64 Go de mémoire centrale,
- bande passante système de 12,8 Go,
- jusqu'à 191 Go de disques en interne,
- jusqu'à 20 To de disques en externe,
- jusqu'à 8 alimentations.

C'est une machine à architecture SMP, c'est-à-dire une architecture MultiProcesseurs Symétrique.



Capacités des matériels

Ultra™ Enterprise™ 10 000





Capacités des matériels

Ultra™ Enterprise™ 10 000

L'architecture de l'Enterprise 10000 s'articule autour d'une interconnexion de type *Cross-Bar*, gérée par une ou deux *Control Board*.

Les *system Board* (16 au maximum), hébergeant la mémoire, les processeurs et les canaux I/O, sont connectées au cross-bar.

L'Enterprise 10000 peut être configuré de façon à atteindre des niveaux de disponibilité de type MainFrame. Ceci est possible grâce à la redondance des composants (alimentations, ventilations, ...) mais aussi avec des dispositifs spécifiques comme :

- *Hot-Swap* de system module,
- *Alternate Path* des I/O,
- *System Service Processor* (station de travail d'administration),
- *Dynamic System Domains*.

Carte système

Une carte système peut recevoir :

- 1 à 4 UltraSPARC à 250 Mhz,
- 1 Mo ou 4 Mo de mémoire cache externe par processeur,
- 0 à 4 Go de mémoire RAM,
- 0 à 2 modules SBus pouvant recevoir chacun 2 contrôleurs (SCSI, réseaux, Fibre optique,...).



Capacités des matériels

Visualisation de la configuration

uname -a

prtconf -v

sysdef -i

prtdiag

sunvts

symon

Capacités des matériels

Visualisation de la configuration

Les commandes permettant de visualiser la configuration d'un serveur sont les suivantes :

■ `uname -a`

Cette commande permet d'obtenir le type de machine et la sous-architecture.

■ `prtconf -v`

Cette commande permet de visualiser la description matérielle de la machine. L'interface `sunvts` est d'abord plus lisible que la sortie de cette commande.

■ `sysdef -i`

Cette commande permet de visualiser la description matérielle de la machine, elle est complémentaire de la commande précédente. Nous ferons la même remarque sur la sortie de cette commande.

■ `prtdiag`

Cette commande permet un diagnostic sur les machines, elle fournit une sortie succincte de la configuration de la machine.

■ `sunvts`

Cette commande permet un diagnostic sur les machines, elle fournit une sortie de la configuration de la machine.

■ `symon`

Cette commande permet un diagnostic sur les machines, elle fournit une sortie de la configuration de la machine, mais elle n'est disponible que sur les serveurs Enterprise.



Capacités des matériels

Visualisation de la configuration

uname -a

```
# uname -a
SunOS eureka 5.5 Generic sun4u sparc SUNW,Ultra-1
#
```

prtconf -v

```
# prtconf -v | more
System Configuration: Sun Microsystems sun4u
Memory size: 64 Megabytes
System Peripherals (Software Nodes):

SUNW,Ultra-1
  System software properties:
    name <relative-addressing> length <0> -- <no value>.
    name <MMU_PAGEOFFSET> length <4>
    value <0x00001fff>.
```

sysdef -i

```
# sysdef -i | more
* Hostid
  80784ceb
*
* sun4u Configuration
*
* Devices
*
packages (driver not attached)
  terminal-emulator (driver not attached)
    value <0x00001fff>.
  name <MMU_PAGESIZE> length <4>
```

Capacités des matériels

Visualisation de la configuration

■ `uname -a`

Cette commande permet d'obtenir des informations sur l'architecture, la sous-architecture et le type de machine.

■ `prtconf -v`

Cette commande permet d'obtenir des informations sur l'architecture, la sous-architecture et le type de machine. Elle fournit toute la configuration matérielle de l'équipement.

■ `sysdef -i`

Cette commande fournit toute la configuration matérielle de l'équipement, ainsi que la valeur de certaines valeurs positionnées dans le noyau (pour les IPC, par exemple).



Capacités des matériels

Visualisation de la configuration

prtdiag

example% /usr/platform/sun4d/sbin/prtdiag

System Configuration: Sun Microsystems sun4d SPARCserver 1000

System clock frequency: 40 MHz

Memory size: 64Mb

Number of XDBuses: 1

CPU Units:	Frequency		Cache-Size		Memory Units: Group Size			
	A: MHz	MB	B: MHz	MB	0: MB	1: MB	2: MB	3: MB
Board0:	50	1.0	50	1.0	32	0	0	0
Board1:	50	1.0	50	1.0	32	0	0	0

=====**SBus Cards**=====

Board0:	0:	dma/esp(scsi)	'SUNW,500-2015'
		lebuffer/le(network)	'SUNW,500-2015'
	1:	<empty>	
	2:	cgsix	'SUNW,501-1672'
	3:	<empty>	
Board1:	0:	dma/esp(scsi)	'SUNW,500-2015'
		lebuffer/le(network)	'SUNW,500-2015'
	1:	<empty>	
	2:	<empty>	
	3:	<empty>	

No failures found in System

=====



Capacités des matériels

Visualisation de la configuration

- `prtdiag`

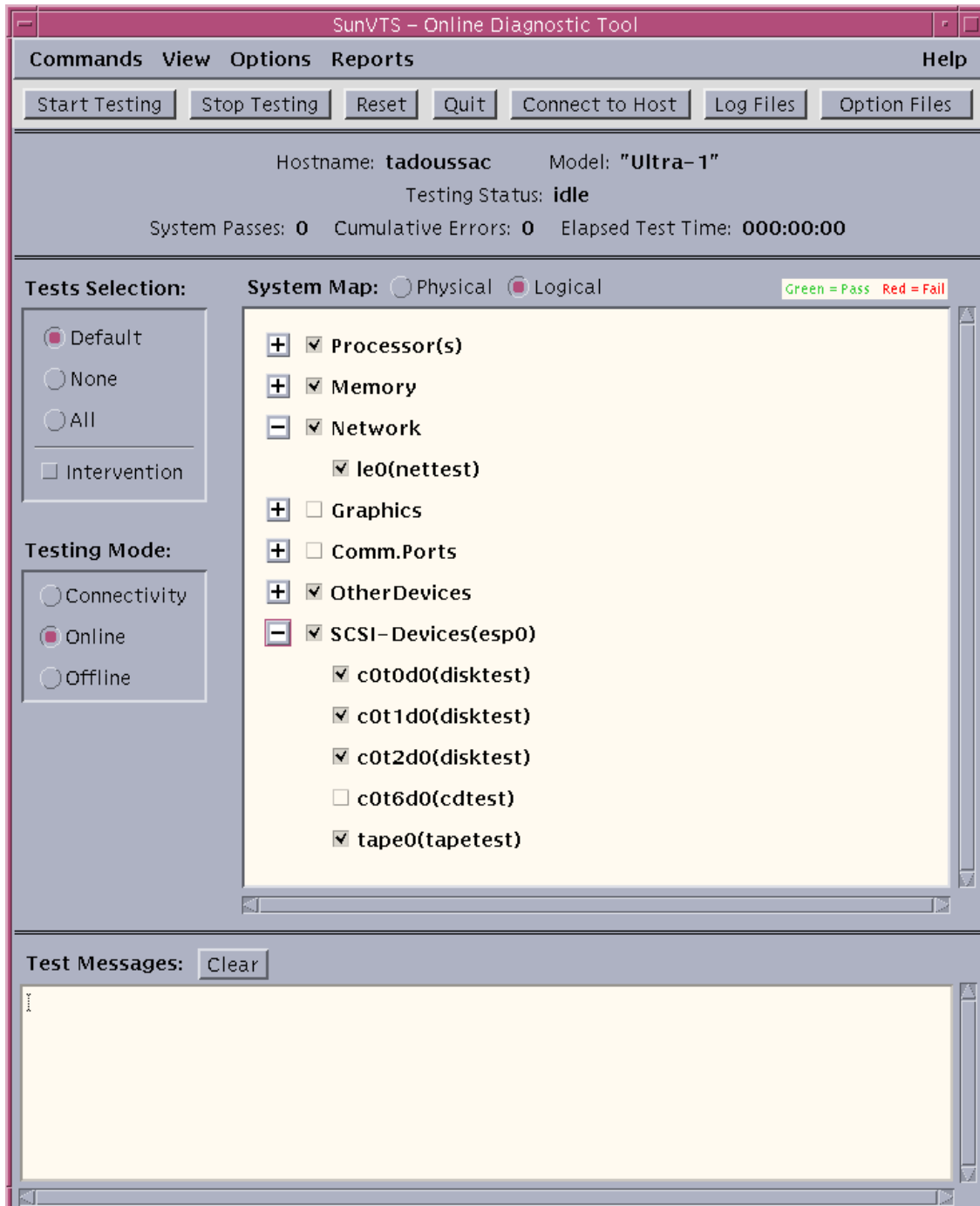
Cette commande permet d'effectuer un diagnostic sur les machines de type sun4m, sun4d et sun4u. Elle propose aussi une visualisation de la configuration des machines.



Capacités des matériels

Visualisation de la configuration

sunvts





Capacités des matériels

Visualisation de la configuration

■ sunvts

Cette commande de diagnostic permet aussi de visualiser la configuration des machines :

```
Processor(s)
system(systest)
  System Configuration= Sun Microsystems sun4u
  Memory size= 32 Megabytes
  System clock-frequency= 72 MHz
cpu-unit0(fputest)
  Type:SPARC V9 based FPU
  clock-frequency: The sparc processor operates at 143 MHz
Memory
kmem(vmem)
  Total Swap: 82MB
mem(pmem)
  Memory Size:32MB
Network
le0(nettest)
  Host_Name: tadoussac
  Host Address: 150.20.200.20
  Host ID: 807af25a
  Domain Name:
Graphics
cgsix0(cg6)
  GX
Comm.Ports
zs0(sptest)
  Port a -- zs0 /dev/term/a : /devices/ ... a
  Port b -- zs1 /dev/term/b : /devices/ ... b
OtherDevices
bpp0(bpptest)
  Logical name: bpp0
diskette(disktest)
  Controller:Intel 82077
sound0(audio)
  Audio Device Type: CS4231 On-board version b
SCSI-Devices(esp0)
c0t0d0(disktest)
  Capacity: 1002.09MB
  Controller: esp0
  Vendor: QUANTUM
  SUN Id: FB1080J SUN1.05
  Firmware Rev: 630D
  Serial Number: 9602421913
c0t1d0(disktest)
  Capacity: 1.98GB
```

Capacités des matériels

Visualisation de la configuration

symon



Capacités des matériels

Visualisation de la configuration

- symon

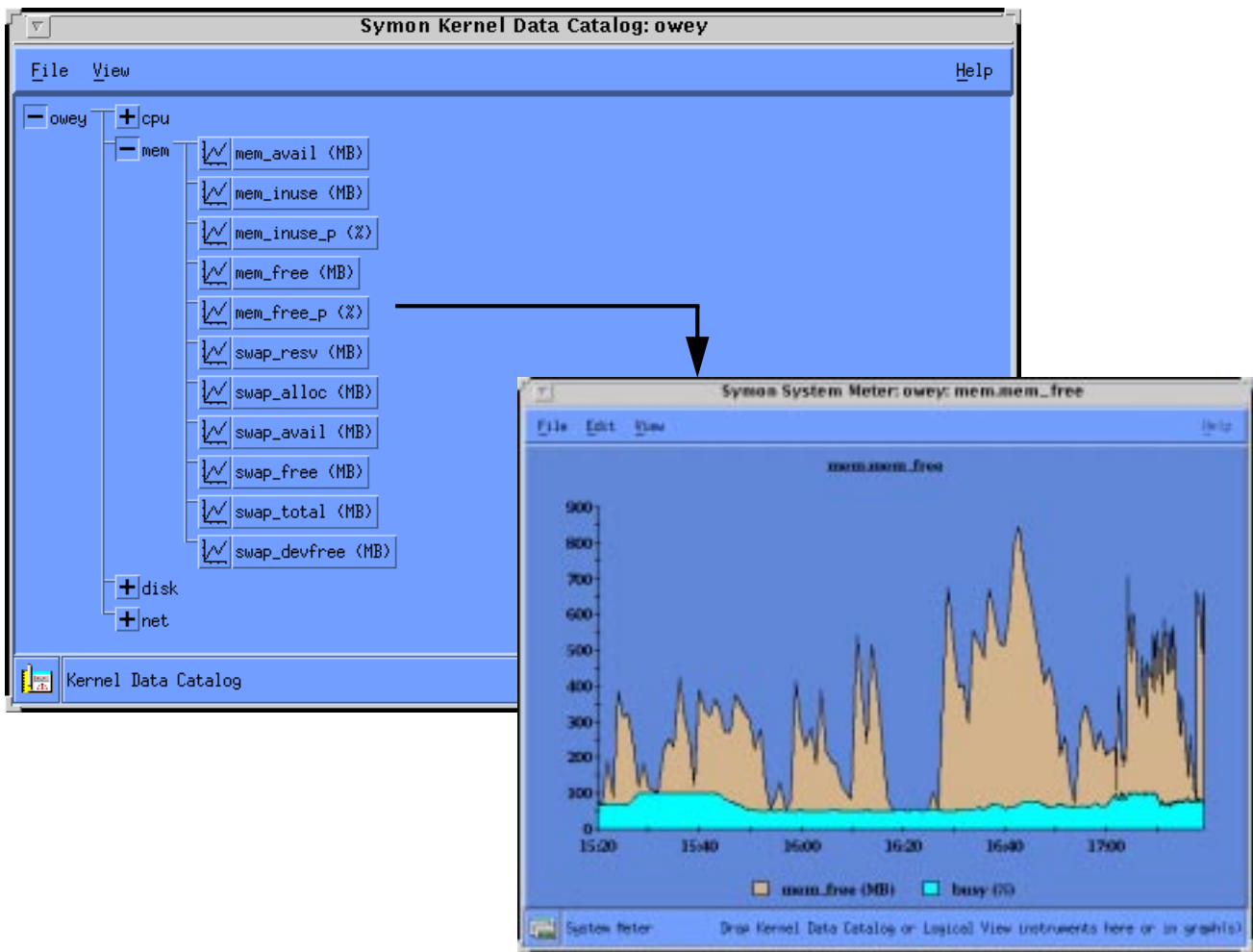
Le dernier membre de la famille Solstice, Solstice SyMON, a été co-développé avec AIM technologie. C'est un outil de surveillance pour les serveurs Enterprise X000 mais aussi des Ultra Enterprise 2, 150 et pour les serveurs 1000 et 2000 depuis la version 1.1 de SyMON.

Solstice SyMON 1.1 permet à l'utilisateur de visualiser simplement des informations sur le système au travers une interface graphique. L'interface graphique peut être installée sur plusieurs machines, ainsi plusieurs personnes peuvent surveiller le même serveur.

Capacités des matériels

Visualisation de la configuration

symon



Capacités des matériels

Visualisation de la configuration

Les consoles, au nombre de sept, se décomposent dans l'ordre suivant :

- les évènements,
- la log système,
- la vue physique du système,
- la vue logique du système,
- les performances et données sur le fonctionnement du système d'exploitation,
- les informations sur les processus,
- le diagnostique en ligne.

Le Kernel Data Catalog offre la possibilité à l'administrateur d'évaluer les performances du serveur sur une période donnée. Les performances sont classées par catégories : CPU, mémoire, disque et réseau.

Les goulets d'étranglement peuvent ainsi être identifiés, et les insuffisances hardware peuvent être anticipées.



Capacités des matériels

Les périphériques disques

SOC

SCSI

Les types de bus SCSI

SCSI-2

fast SCSI-2

differential SCSI-2

single-ended wide SCSI-2

differential wide SCSI-2

SCSI-3

Les autres types incluant un protocole SCSI

Fiber Channel-Arbitrated Loop FC-AL

Capacités des matériels

Les périphériques disques

Deux types de connexions sont disponibles pour les périphériques disques : on dispose de canaux fibres optiques (SOC) et de canaux SCSI. La fibre optique permet un transfert synchrone de 25 M octets par seconde.

Les types de bus SCSI

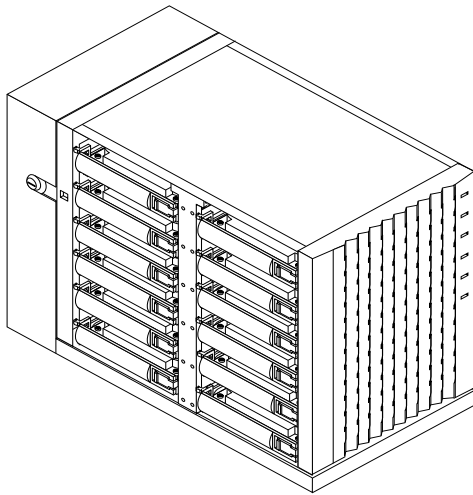
Types	Vitesses
SCSI-2 bus 8 bits 7 targets	5 Mo/s
fast SCSI-2 bus 8 bits 7 targets	10 Mo/s
single-ended wide SCSI-2 bus 16 bits 15 targets	20 Mo/s
differential SCSI-2 bus 8 bits 7 targets	10 Mo/s
differential wide SCSI-2 bus 16 bits 15 targets	20 Mo/s
SCSI-3	40 Mo/s
SOC	25 Mo/s
FC-AL	100 Mo/s



Capacités des matériels

Les périphériques disques

Sun StorEdge Multi-pack



SCSI ID 2	SCSI ID 10
SCSI ID 3	SCSI ID 11
SCSI ID 4	SCSI ID 12
SCSI ID 5	SCSI ID 13
SCSI ID 8	SCSI ID 14
SCSI ID 9	SCSI ID 15

Capacités des matériels

Les périphériques disques

Multi-pack

Unipack :

- 2.1-GB fast/wide SCSI-2, 7200-RPM hard disk
- 4.2-GB fast/wide SCSI-2, 7200-RPM hard disk
- 9.1-GB fast/wide SCSI-2, 7200-RPM hard disk
- 2.5-GB QIC tape drive
- 4- to 8-GB (compressed) 4 mm DDS-2 tape drive
- 12- to 24-GB (compressed) 4mm DDS-3 tape drive
- 7- to 14-GB (compressed) 8 mm tape drive
- 20- to 40-GB (compressed) 8mm tape drive
- SunCD™ 4 CD-ROM drive

Multipack (12) :

- 2, 6, 12 disques 1-inch de 2,1 GB 7200-rpm, 25,2 GB max
- 2, 6, 12 disques 1-inch de 4,2 GB 7200-rpm, 50,4 GB max

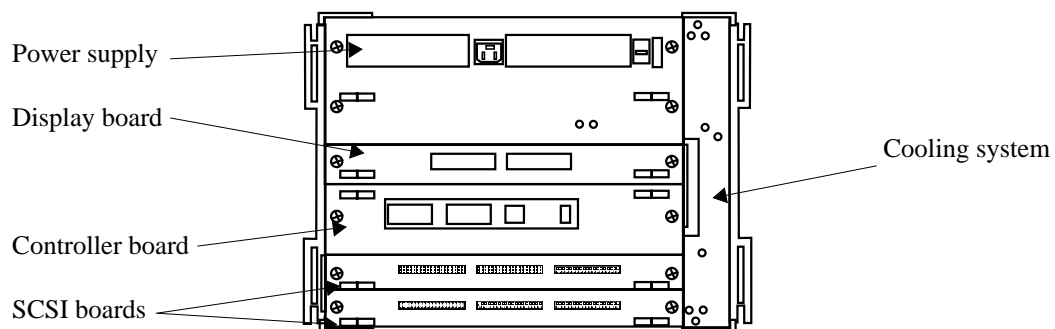
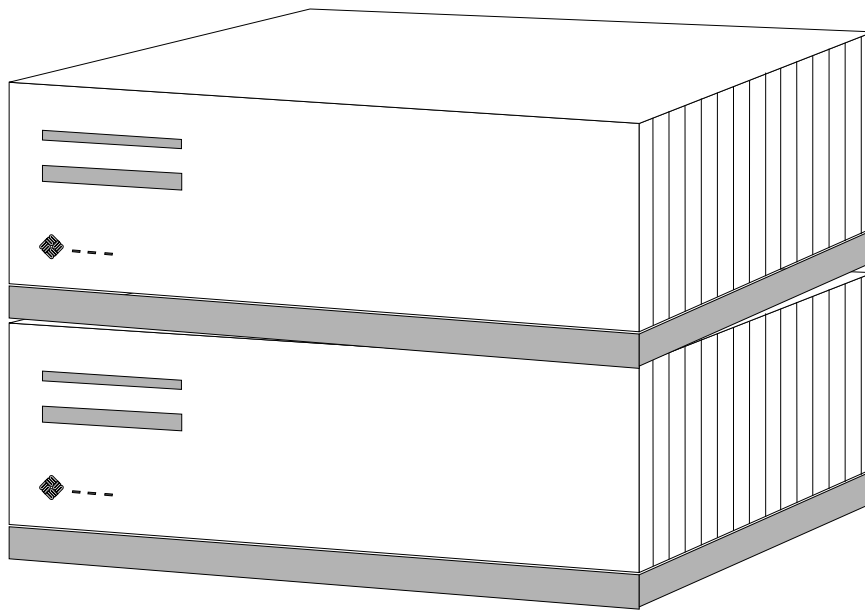
Multipack (6) :

- 2, 4, 6 disques 1.6-inch de 4.2-GB 5400-rpm, 25,2 GB max
- 2, 4, 6 disques 1.6-inch de 9.1-GB 7200-rpm, 54,6 GB max

Capacités des matériels

Les périphériques disques

Sparc Storage Array



Capacités des matériels

Les périphériques disques

Sparc Storage Array

Ce périphérique supporte de 6 à 30 disques 3.5 in. (demi-hauteur) par SPARCstorage Array. Chaque coffret comprend trois tiroirs pouvant contenir jusqu'à 10 disques. Le modèle 100 comprend des disques de 535 M octets, le modèle 101 des disques de 1.05 G octets :

- contient un bloc d'alimentation fiabilisé,
- contient des contrôleurs de fibre optique,
- est géré par le logiciel Volume Manager,
- propose deux connexions fibre optique par carte SBus.

Model 112

- 3 tiroirs de 10 disques de 2,1 GB,
- 2 contrôleurs fast/wide SCSI-2 par tiroir WARM PLUG,
- 2 SOC, 2 hosts
- capacité 63 GB max

Model 114

- 3 tiroirs de 10 disques de 4,2 GB,
- 2 contrôleurs fast/wide SCSI-2 par tiroir WARM PLUG,
- 2 SOC, 2 hosts
- capacité 126 GB max

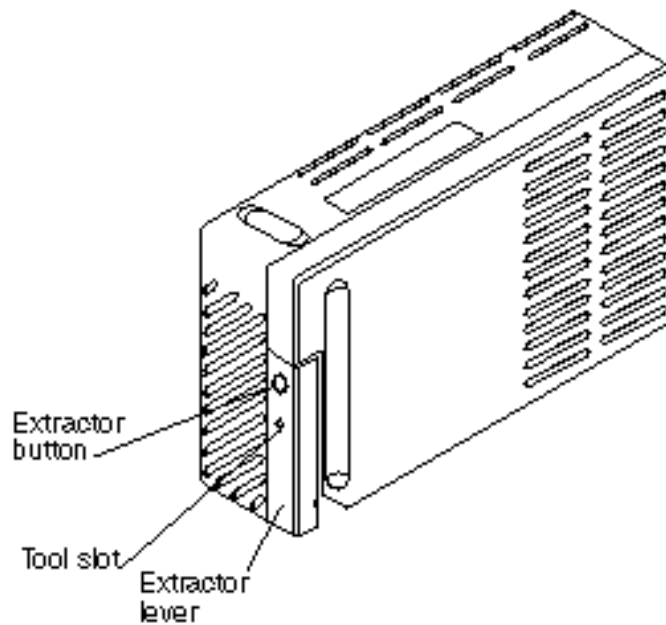
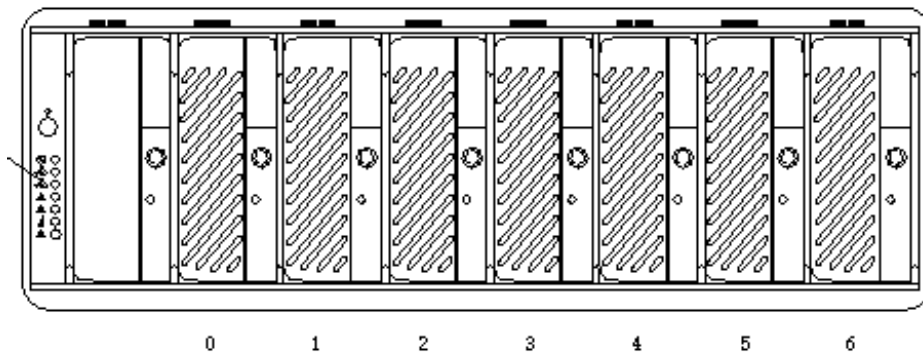
Model 210

- 6 tiroirs de 6 disques differential fast/wide SCSI-2 de 2,1 GB, capacité 75,6 GB max, ou 2,9 GB, capacité 104,4 GB max, ou 9,1 GB, capacité 327,6 GB max
- 2 SOC, 2 hosts

Capacités des matériels

Les périphériques disques

RSM Disk Tray





Capacités des matériels

Les périphériques disques

RSM Disk Tray

Model 214 RSM

- 7 tiroirs RSM de 7 disques differential fast/wide SCSI-2 de 4,2 GB,
- 2 SOC, 2 hosts
- disque HOT PLUG,
- capacité 205.80 GB max

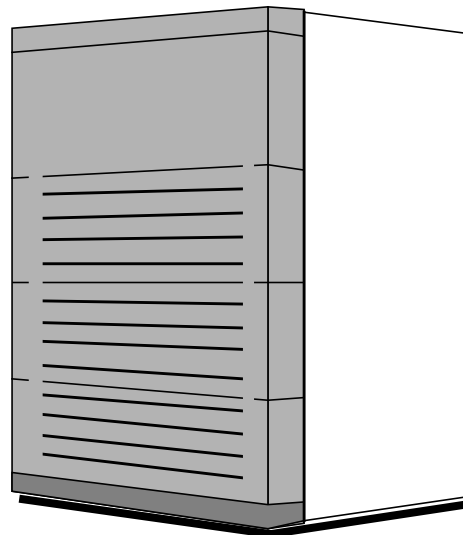
Model 219 RSM

- 7 tiroirs RSM de 7 disques differential fast/wide SCSI-2 de 9,1 GB,
- 2 SOC, 2 hosts
- disque HOT PLUG, capacité 445.90 GB max

Capacités des matériels

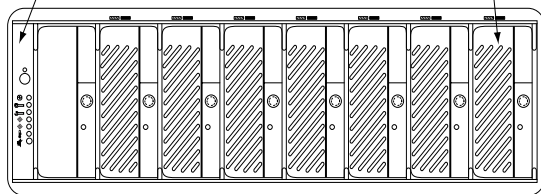
Les périphériques disques

Sun StorEdge A 3000

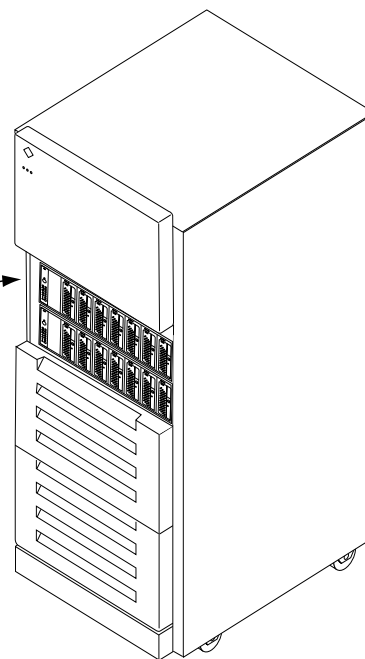


Status Indicators

Hot Plug Drive



SPARCstorage RSM Disk Tray
with Hot Plug Power, Disks, and Cooling



RSM Array 2000

Capacités des matériels

Les périphériques disques

Sun StorEdge A 3000

L'architecture du Sun Enterprise Network Array 3000 reprend l'architecture disque classique de Sun à base de modules RSM, et utilise un contrôleur SONOMA, proposant entre autre du RAID hardware et un doublement de tous les composants matériels. Le Sun Enterprise Network Array 3000 est connecté au host via 2 canaux Ultra SCSI-3 à 40 Mo/sec. Ces deux canaux travaillent en parallèle dans un mode de fonctionnement normal, ils assurent également un fonctionnement redondant, en cas de défaillance d'un des deux contrôleurs, toute l'activité est reportée automatiquement sur le contrôleur restant actif et ce, de façon transparente pour les applications (ceci est fait au niveau du driver).

Le module contrôleur RAID hardware effectue tous les calculs de parité et décharge ainsi le CPU de l'ordinateur hôte de cycles de calcul.

Dans l'armoire Sun Enterprise Network Array 3000, on dispose de 5 plateaux de disques RSM, équipés de 7 disques hot-plug de 4.2 Go ou 9 Go, soit 147 Go ou 318 Go de données. Chaque plateau est équipé de ventilateurs et d'alimentation redondants.

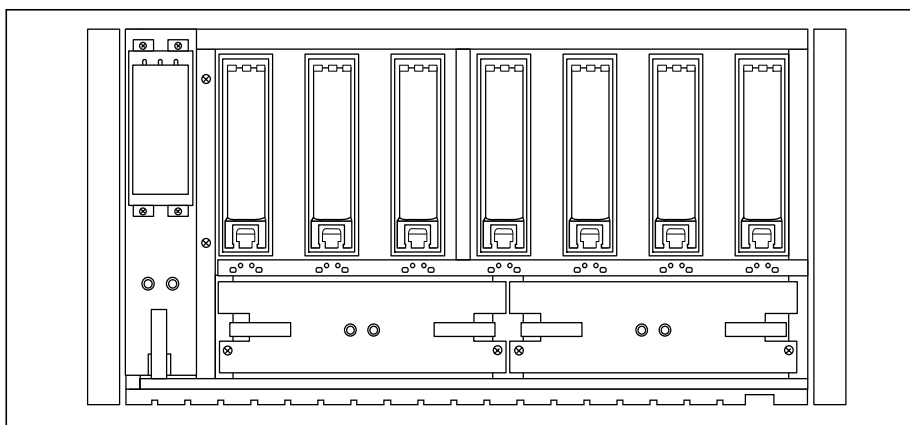
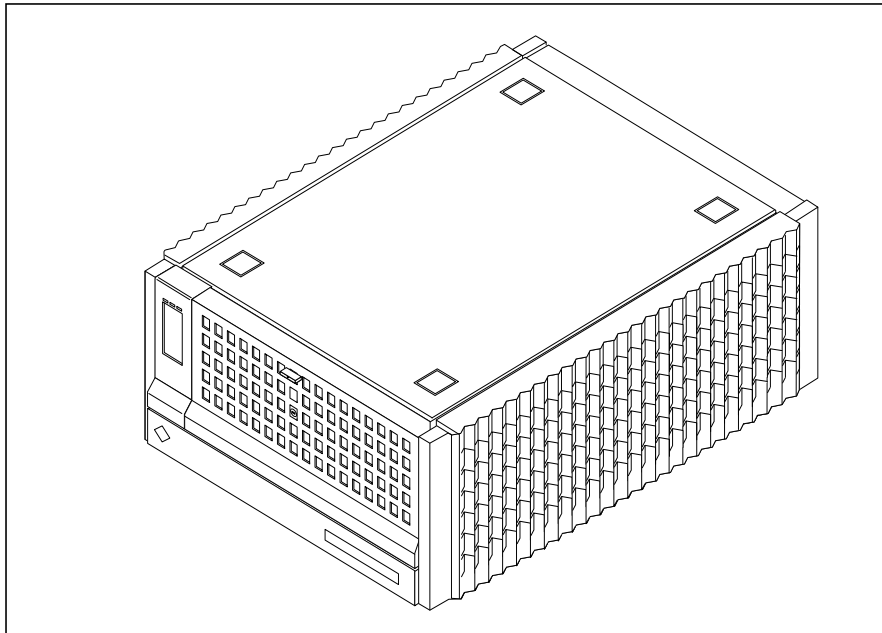
Des composants hot-plug permettent aux administrateurs de remplacer à chaud une unité défaillante alors que le système est en opération. Le système disque Sun Enterprise Network Array 3000 offre :

- 2 contrôleurs RAID hardware hot-plug pour des reprises en cas de panne, automatiques et transparentes,
- des disques RSM hot-plug.

Capacités des matériels

Les périphériques disques

Sun StorEdge A 5000



Capacités des matériels

Les périphériques disques

Sun StorEdge A 5000

- Support de grande capacité disque

Il permet de configurer de 45 Go à plus de 20 To en volume disque.

- Fonctions RAS (Reliability/ Availability/ Serviceability) très complètes

Il offre une redondance N+1 matérielle et la possibilité d'échange à chaud des disques, des blocs d'alimentations, des ventilateurs et des cartes interfaces garantissent la disponibilité continue du réseau de disques.

- Tolérance aux pannes

Une redondance N+1 matérielle peut être configurée pour assurer une transparence complète d'un dysfonctionnement matériel (redondance des blocs d'alimentation, redondance des ventilateurs, redondance des cartes contrôleurs bus, redondance du bus interne, redondance des interfaces FCAL, redondance des liens SCSI optiques),

- Performance et modularité

Chaque disque dispose d'un double attachement à 100 Mo/s full duplex. L'architecture est flexible et ajustable en fonction de la capacité de stockage des serveurs, grâce à l'emploi de la technologie *Fiber Channel Arbitrary Loop* (FCAL).

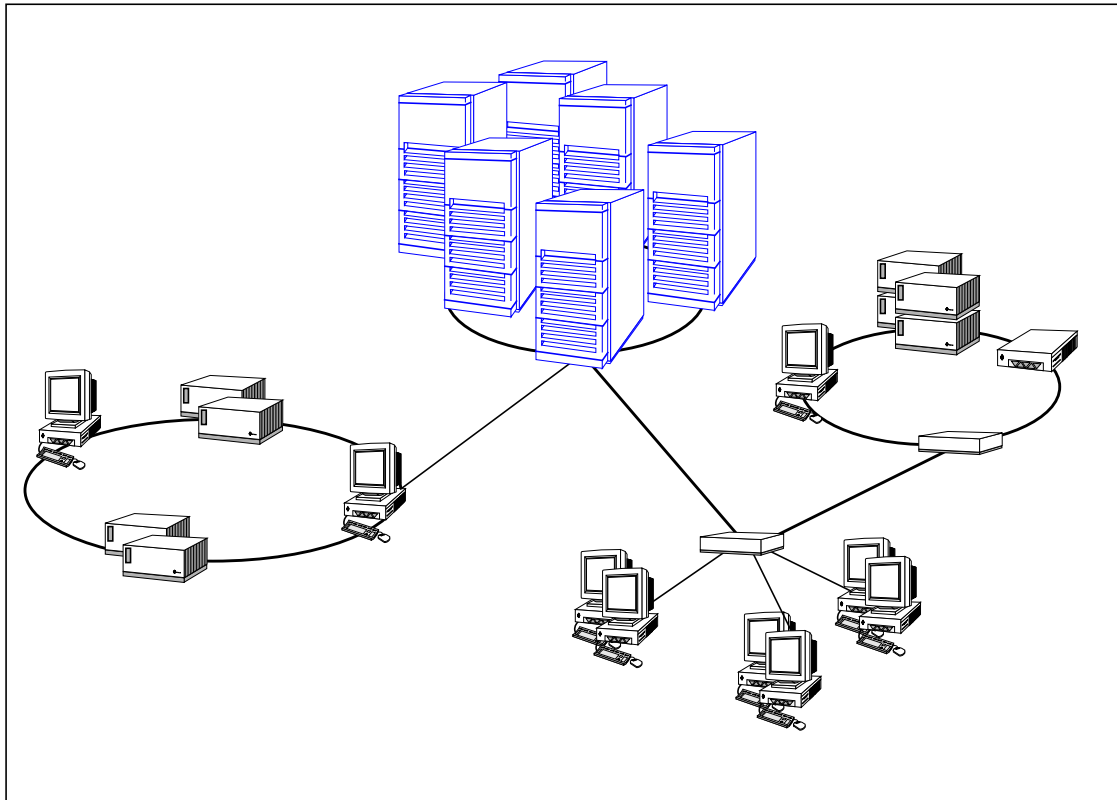
- Administration graphique

La configuration, gestion des volumes et la supervision du réseau de disques s'effectuent au travers d'une interface graphique (Veritas Volume Manager, et ou Solstice DiskSuite).

Capacités des matériels

Les périphériques disques

Sun StorEdge A 5000



Capacités des matériels

Les périphériques disques

Sun StorEdge A 5000

Les caractéristiques techniques de FC-AL sont les suivantes :

- *Vitesse, l'accès au Gigabit/seconde*
Transferts de données à 200 Mo/s en mode full duplex (400 Mo/s prévus)
- *Topologie*
FC-AL permet de construire des infrastructures évoluée de stockage de données similaires aux architectures réseaux actuelles en utilisant des hubs, des commutateurs, la mise en oeuvre des chemins d'accès concurrents et redondants.
- *Extensibilité*
Possibilité d'adresser jusqu'à 127 périphériques FC-AL par contrôleur, distance de déport de connexion jusqu'à 500m (10 km à venir)
- *Standardisation*
FCAL fait partie du standard SCSI-3 ANSI/ISO

Une performance de l'offre Sun Enterprise network Array :

- Disque UltraSCSI SEAGATE 7200 rpm de capacité 9.1 Go en double attachement.
- Convertisseur optique GBIC à 100 Mo/s
- Bande passante de 100 Mo/s par anneaux, 200 Mo/s sur 2 anneaux en équilibre de charge avec un laser à spectre court. (200 Mo/s par anneaux courant Mars 98 avec un laser à spectre large).
- 14, 000 IOPS en anneaux double
- 190 Mo/s en débit soutenu en anneau double



Capacités des logiciels

Les versions de systèmes d'exploitation traités

Les applicatifs liés au système d'exploitation

les RAID : VM, SDS, rm6

les systèmes de fichiers : UFS, VxFs



Capacités des logiciels

les versions d'OS couvertes

Les versions traitées par le support seront le 2.6, 2.5.1 et la 2.5.

Les mécanismes internes peuvent être différents ou les variables traitées différemment. Il est donc important de connaître avec exactitude la version exacte du système d'exploitation mis en oeuvre tant sur le serveur que sur le client.

Les applicatifs liés au système d'exploitation

Certains applicatifs sont liés au système d'exploitation (essentiellement les applicatifs supervisant les accès disques). Ainsi les divers choix de ces logiciels peuvent avoir des conséquences non négligeables sur les performances globales de la plate-forme.

les RAID

Les choix des produits, ainsi que les configurations sont fondamentales pour la supervision et l'amélioration des performances sur le serveur.

les systèmes de fichiers

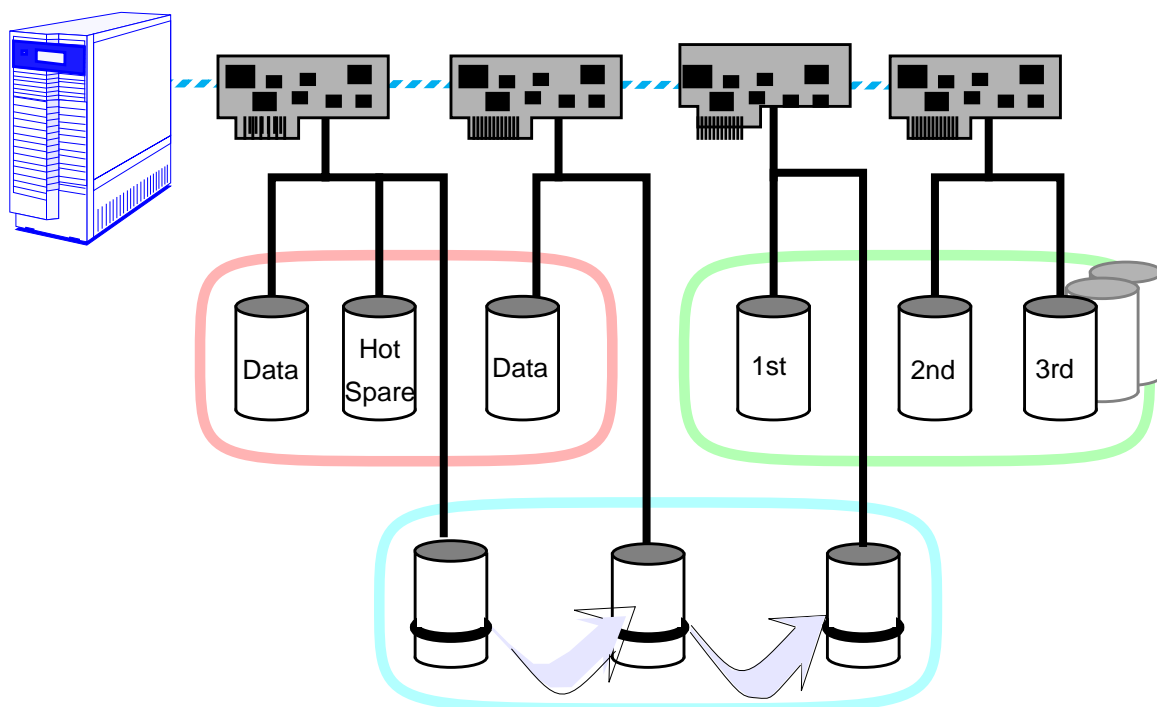
Il en est de même pour la gestion des systèmes de fichiers.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

- les techniques
- les implémentations



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

- les techniques

Plusieurs techniques d'associations des disques physiques sont envisageables. Chacune correspond à un cahier des charges bien précis et améliore les gestions des disques en terme de :

- sécurité,
- performances.

L'administrateur devra veiller à utiliser au mieux les choix qui lui sont proposés pour tirer les meilleures performances de ses matériels.

- les implémentations

Deux types d'implémentations existent :

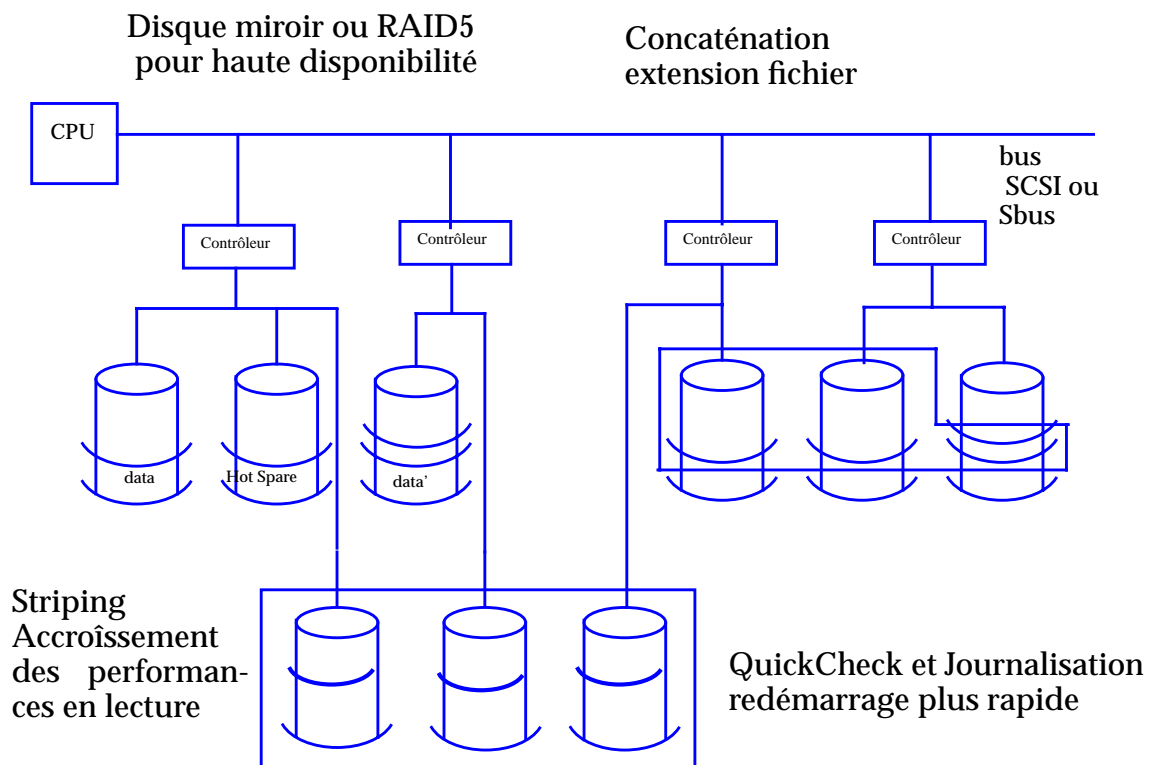
- les raids matériels,
- les raids logiciels.

Chacun correspond à un choix bien précis et ce dernier va avoir un impact non négligeable sur les performances des volumes logiques résultants.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

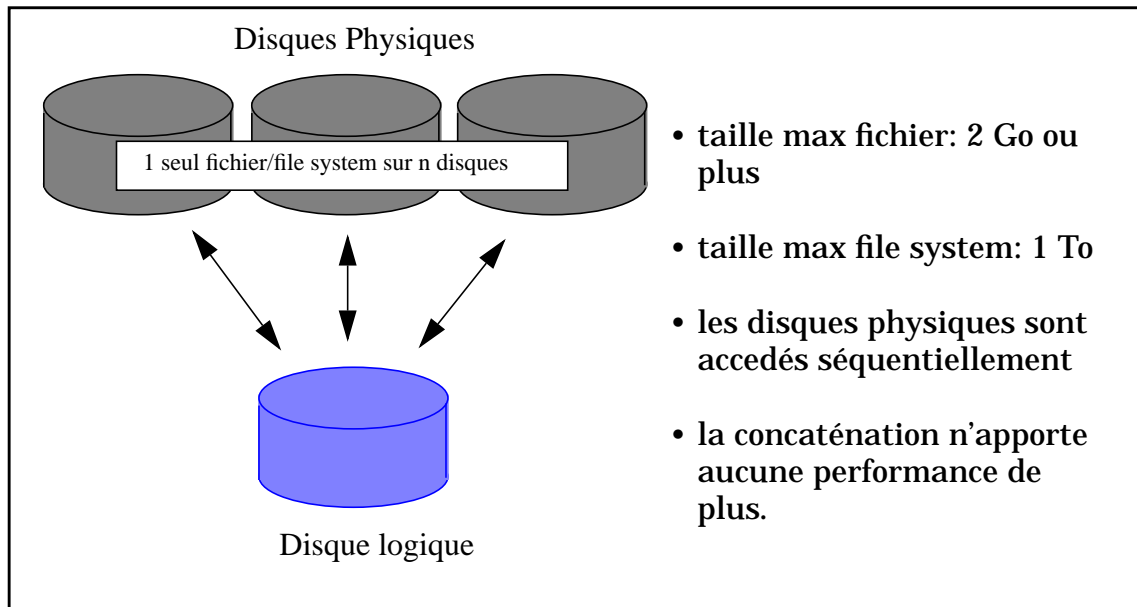
Les logiciels de gestion des raids proposent à la base les choix suivants :

- le Mirroring (RAID 1), très performant pour une haute disponibilité et sécurité des données,
- le Stripping (RAID 0), ou optimisation des écritures séquentielles sur disques, par parallélisation des I/O sur plusieurs disques,
- le RAID 5, ou Stripping avec calcul de parité, qui permet de gagner de la place disque par rapport à une configuration en Mirroring,
- la Journalisation de systèmes de fichiers, permettant le redémarrage rapide du système en cas de panne disque (le `fsck` ne s'effectue plus que sur le journal, de taille très réduite par rapport à celle du système de fichiers).

Capacités des logiciels

Les applicatifs liés au système d'exploitation

Concaténation



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

- Concaténation

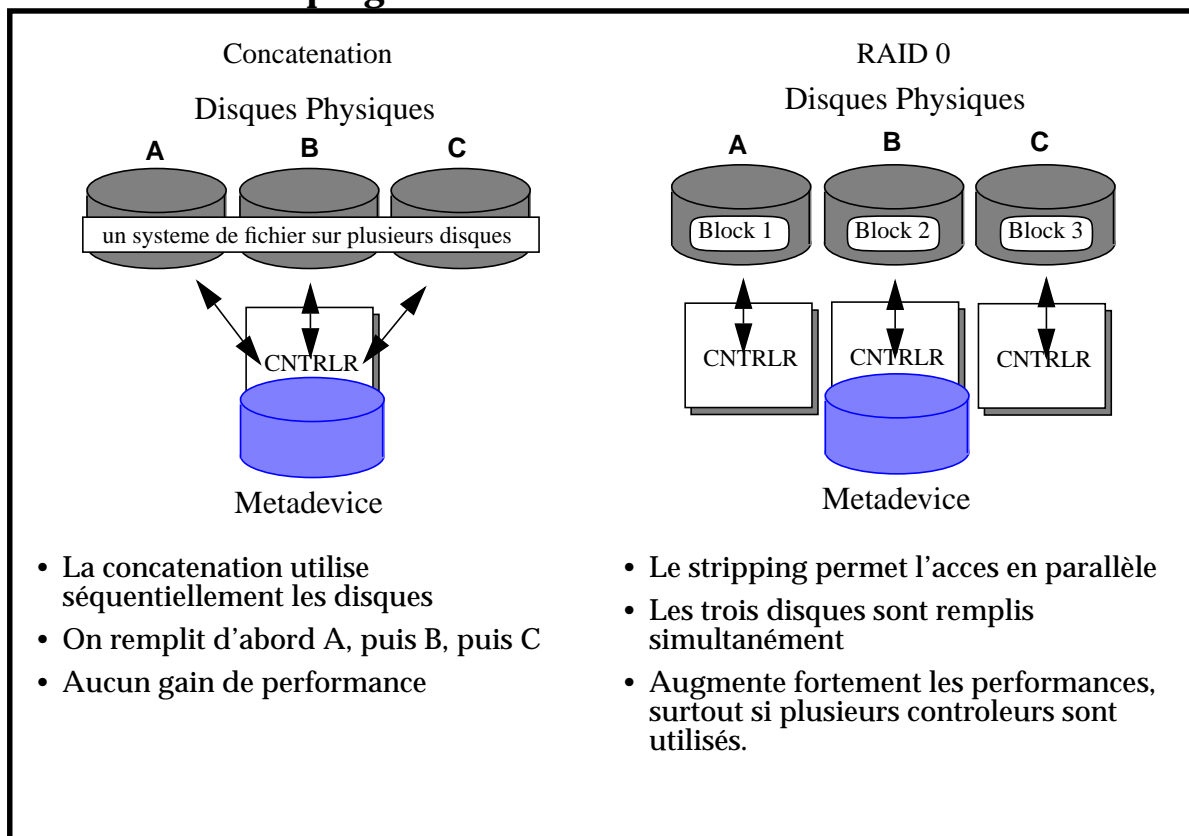
Pour une plus grande flexibilité du système, il est possible d'accroître considérablement la taille maximale d'un système de fichier quelconque, et ce de façon dynamique ou statique, et ceci pour un système de fichiers ou pour un raw device.

L'extension accroît l'importance et l'impact des erreurs disque ou contrôleur : dans le cas où un fichier s'étend sur plusieurs disques, le non fonctionnement de n'importe lequel de ces disques ou contrôleurs, mènerait à la perte de ce fichier. Par conséquent, les utilisateurs ayant besoin d'une grande disponibilité de données avec de grands fichiers, doivent obligatoirement utiliser une fonctionnalité de miroir pour suppléer à l'impact de la concaténation des fichiers.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

Striping



- La concatenation utilise séquentiellement les disques
- On remplit d'abord A, puis B, puis C
- Aucun gain de performance

- Le striping permet l'accès en parallèle
- Les trois disques sont remplis simultanément
- Augmente fortement les performances, surtout si plusieurs contrôleurs sont utilisés.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

■ Striping

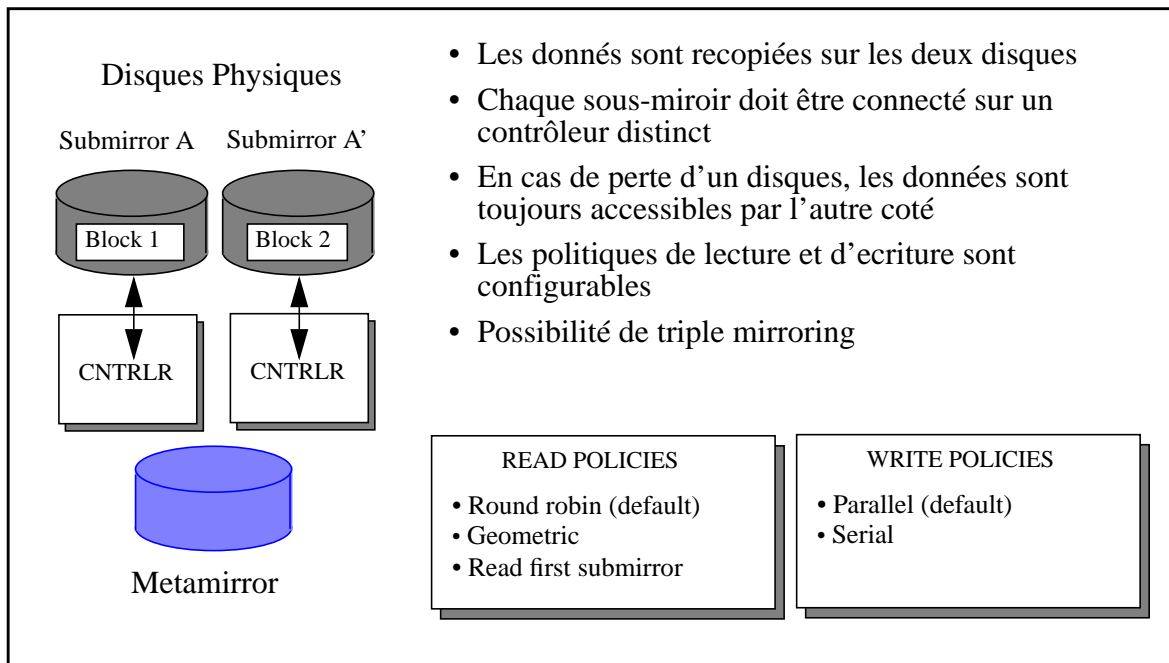
Le "**Disk Striping**" augmente la performance d'une application en accroissant le débit des entrées/sorties. Il répartit la charge des E/S sur plusieurs disques, ce qui par conséquent augmente le débit E/S disponible pour un process unique, et ce plus précisément lors de l'utilisation de grands fichiers.

Le "**Disk Striping**" permet d'améliorer la performance globale d'un système bien configuré.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

Miroir



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

■ Miroir

Pour un stockage de haute disponibilité : une partition (système de fichier ou "raw device") est copiée en miroir sur une seconde partition disque par écriture redondante et simultanée.

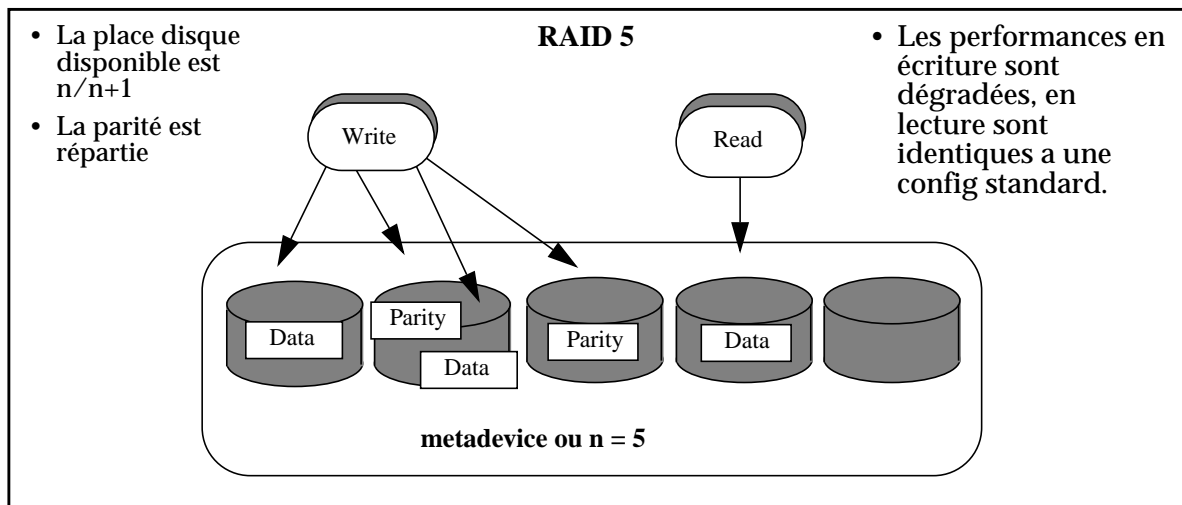
Lors d'une panne disque ou contrôleur, les utilisateurs continuent à travailler en utilisant de façon transparente la copie valide des données. Le "disk Mirroring" permet également une sauvegarde ("*backup*") en ligne des données.

Dans le cas d'une panne disque ou contrôleur, le miroir permet aux applications logicielles de continuer normalement leur exécution, en utilisant la copie redondante des données comme backup instantané. Le recouvrement des données et la reconfiguration système en cas de panne disque s'effectue en ligne.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

Raid 5



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

■ Raid 5

Dans le Mirroring, chaque donnée est recopiée 2 fois, voire trois dans le cas du triple mirroring. Ceci impose de doubler, voire tripler, la place disque nécessaire par rapport aux données brutes.

Le concept du RAID 5 est né de ce constat. Le principe consiste à stripper les données sur n disques, puis à écrire une parité sur un disque $n+1$. Ainsi, la perte d'un disque sur les « n » de données n'entraîne plus la perte de toutes les données, car l'information combinée (parité + données valides sur $n-1$ disques + connaissance du positionnement dans la chaîne du disque perdu) permet à chaque instant au système de reconstruire les données perdues.

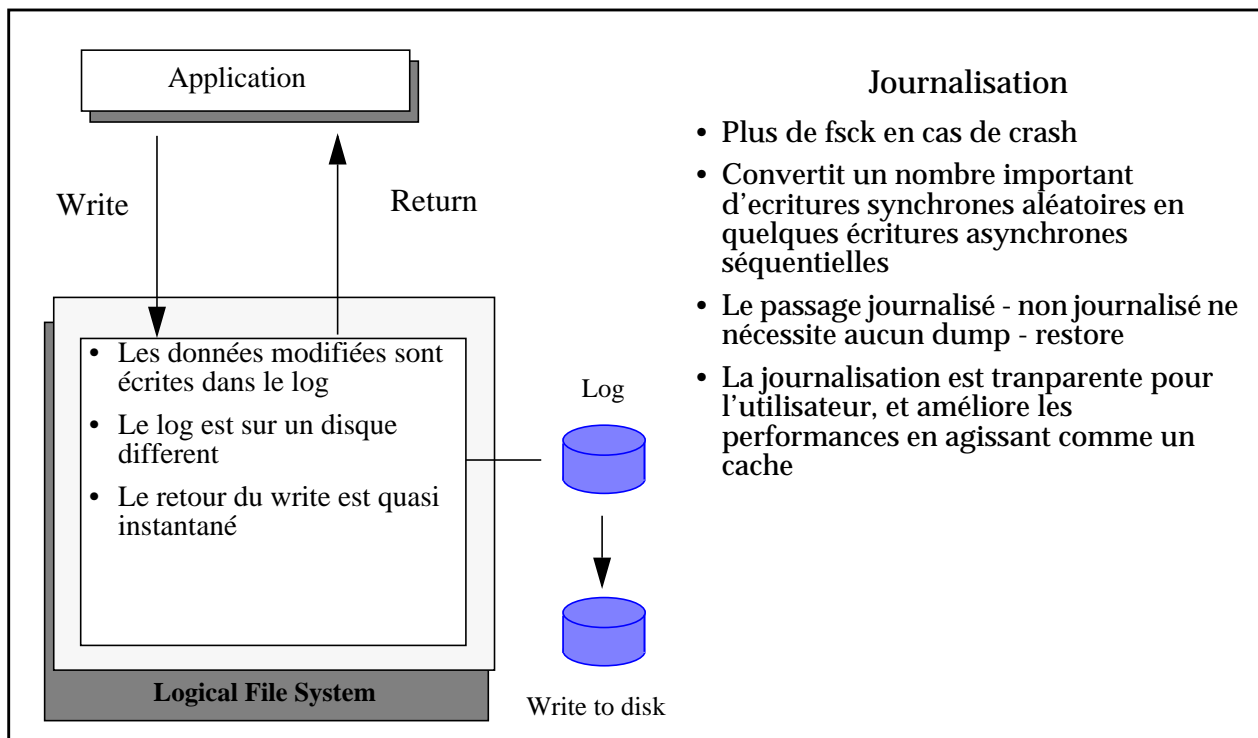
De plus, pour des raisons de performance, la parité a été répartie sur les $n+1$ disques composant la chaîne RAID5 complète, de façon à éviter qu'un seul disque limite la performance de toute la chaîne.

La perte de plus d'un disque entraîne la perte de toutes les données, et les performances du RAID 5 notamment en écriture sont nettement moins bonnes que celles du RAID 0+1, puisque chaque écriture nécessite le recalcul de la parité associée. Par contre, l'avantage indiscutable du RAID 5 est le gain de place disque par rapport au Mirroring pour une configuration sécurisée.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

UFS logging



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

■ UFS Logging

La journalisation de systèmes de fichiers est une technique empruntée aux SGBDs et appliquée à UFS. Elle consiste à journaliser, pour chaque écriture, toutes les modifications atomiques effectuées sur la partition master dans une autre partition beaucoup plus petite appelée partition de logging, et de ne “commiter” l'écriture sur le master qu'en une opération atomique. Ainsi, la partition master est toujours dans un état cohérent, et ne nécessite plus jamais de `fsck` même en cas de “crash & reboot” violent du système, ce qui représente un gain de temps important.

Il est clair que le log est une partie importante du couple, puisque c'est lui qui contient toutes les informations de modification du device. Il est donc fortement conseillé de le mirroring. Par contre, sa taille est par essence très réduite par rapport à celle du master (2% en moyenne).

Enfin, signalons que ce mécanisme entraîne un gain de performance important en écriture synchrone. En effet, le log sert de cache disque, et l'écriture effective (le “commit”) intervient de façon désynchronisée dans deux cas :

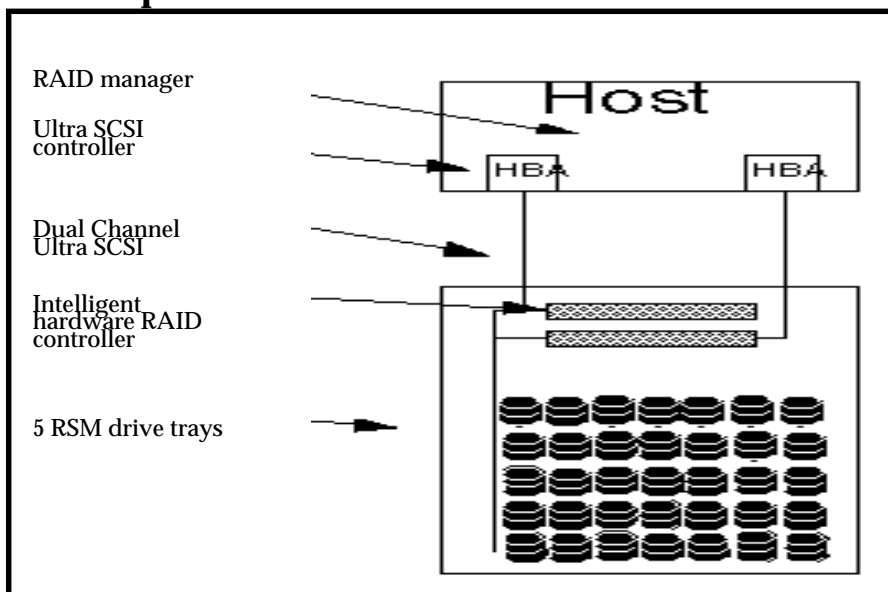
- lorsque le log est plein,
- lorsque le master n'a pas été accédé en écriture depuis plus de 5 secondes (configurable).

Capacités des logiciels

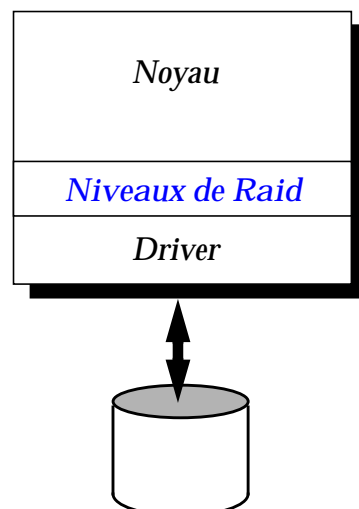
Les applicatifs liés au système d'exploitation

les niveaux de RAID

Implantation matérielle



Implantation logicielle



Capacités des logiciels

Les applicatifs liés au système d'exploitation

les niveaux de RAID

Les niveaux de raids que nous avons décrits précédemment sont implantés logiciellement (un logiciel interne à la machine, en général présent sous forme de module), ou matériellement (intégré dans un contrôleur spécifique). Chaque solution possède des avantages et des inconvénients.

Implantation matérielle

Les performances sont semblables à celles obtenues sur un disque classique (voire meilleures). Ainsi, il n'est pas pénalisant de travailler avec des volumes basés sur du raid 5.

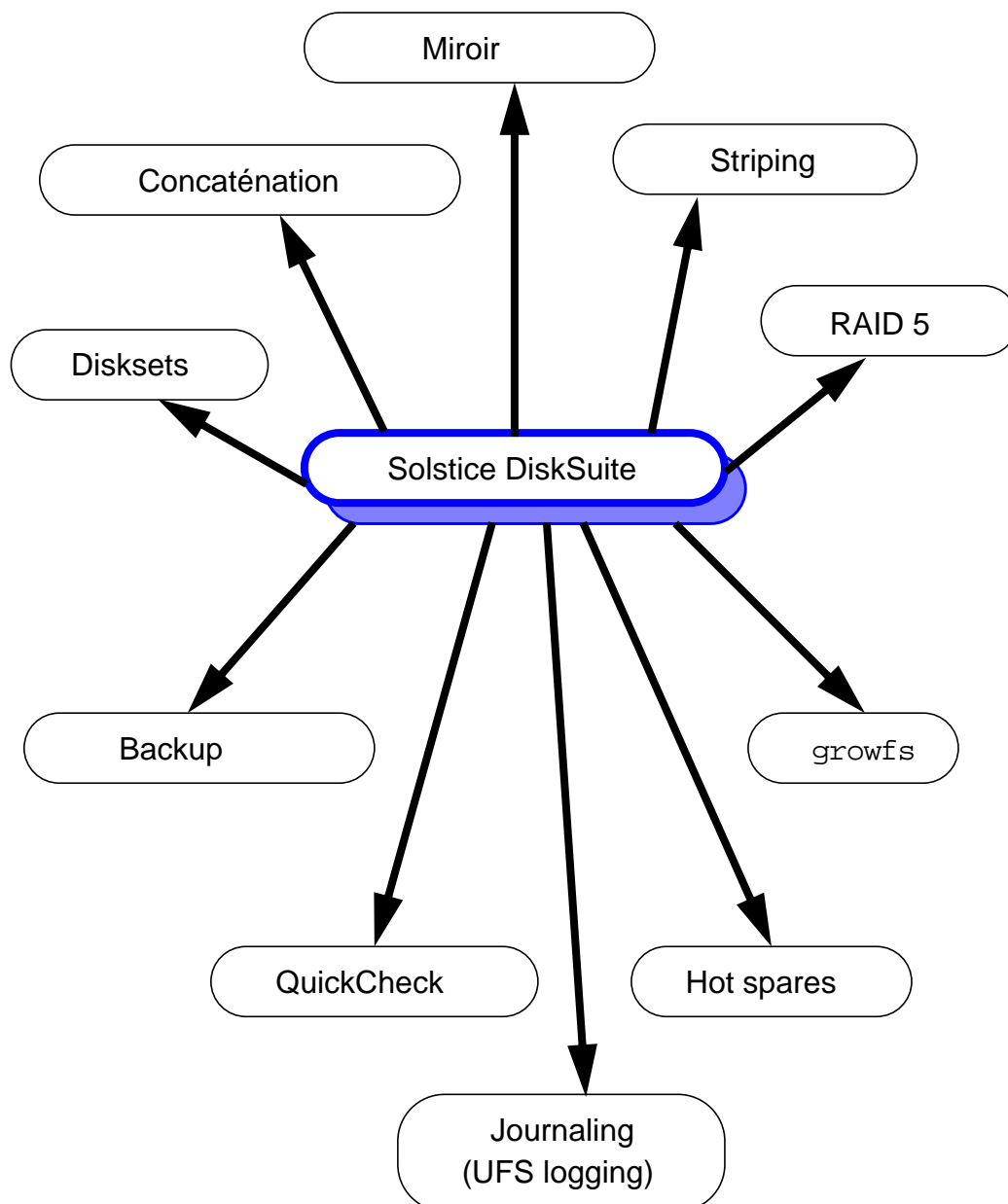
Implantation logicielle

Ici, le travail va être effectué par serveur, il convient donc de choisir judicieusement les niveaux de raid implémentés ainsi que les paramètres du stripping (se référer au dernier module).

Capacités des logiciels

Les applicatifs liés au système d'exploitation

SDS





Capacités des logiciels

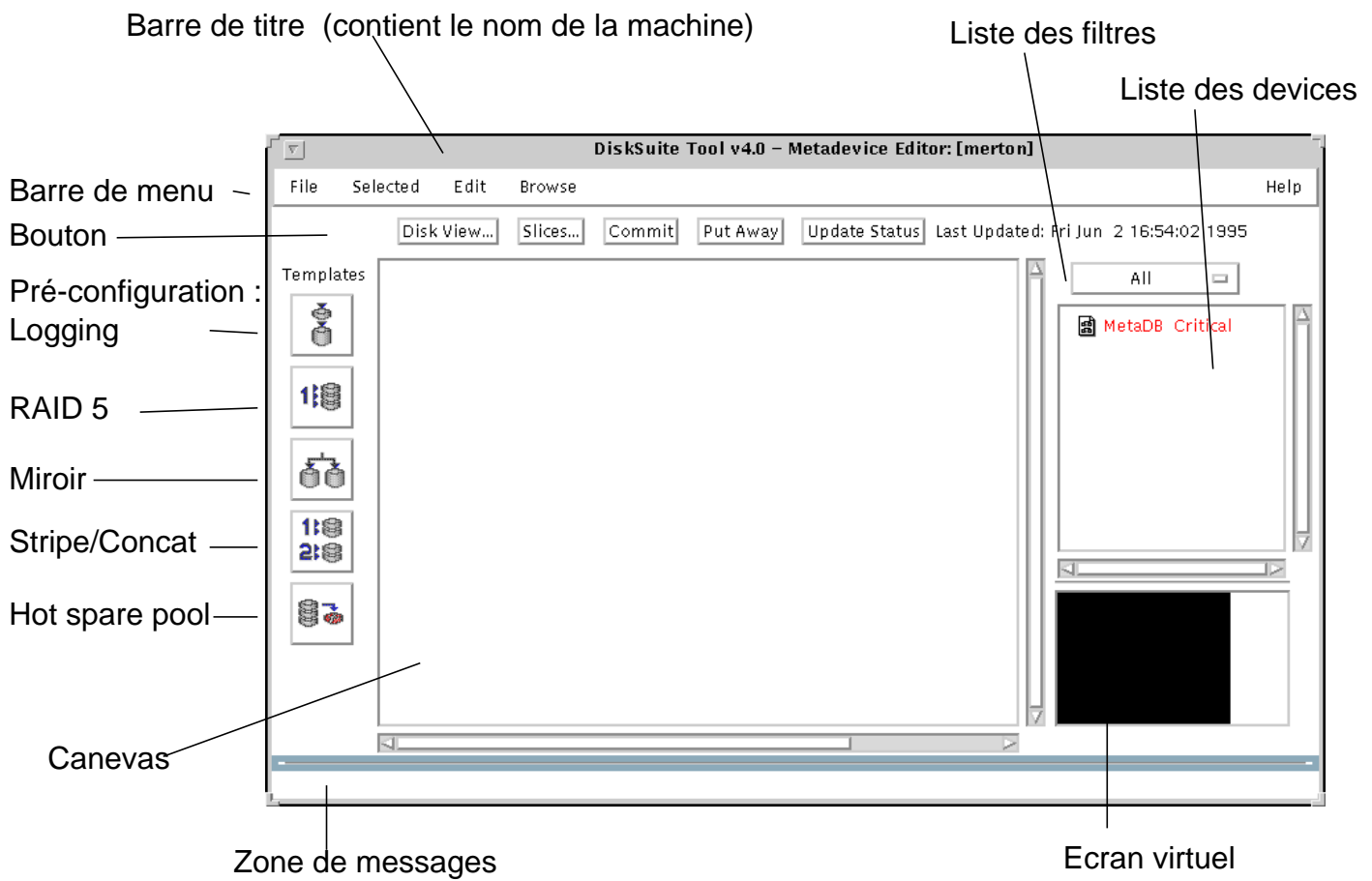
Les applicatifs liés au système d'exploitation : SDS

- **quickcheck** – C'est un programme standard fourni depuis la version SunOS 4.1.2. Il permet un redémarrage rapide du système grâce à une phase de `fsck` exécutée uniquement si nécessaire. Cette fonctionnalité est importante dans un contexte de haute disponibilité, puisqu'elle permet un process de boot accéléré, ou un redémarrage rapide sur un disque de backup.
- **Disk striping** – Cette fonctionnalité permet d'améliorer les performances de lecture et d'écriture en stockant les données sur plusieurs disques.
- **Disk concatenation** – Cette fonctionnalité permet de créer des volumes de taille plus importante qu'une partition ou qu'un disque. Elle est utilisée pour les applications nécessitant des zones de stockage importantes.
- **Diskset** – C'est un ensemble de disques partagés entre deux machines. Cette fonctionnalité permet de faciliter le partage de disques dans un environnement de haute disponibilité.
- **Disk mirroring** – Cette fonctionnalité permet de s'assurer une complète disponibilité des données, même lors de la panne d'un disque. Tout système de fichiers peut être miroré : `root`, `swap` et `/usr`.
- **Hot spare** – C'est un ensemble de partitions qui est mis en service automatiquement lors de la détection d'une panne sur un disque utilisé dans un miroir.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

SDS



Capacités des logiciels

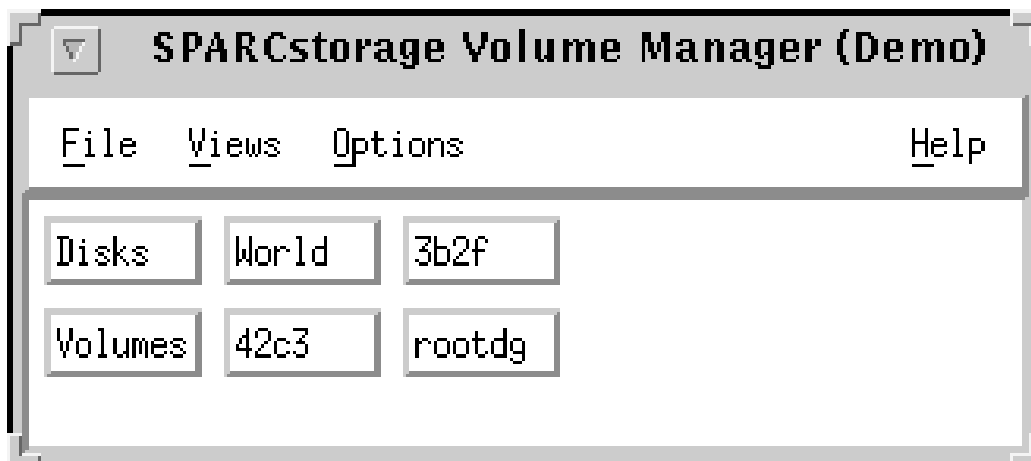
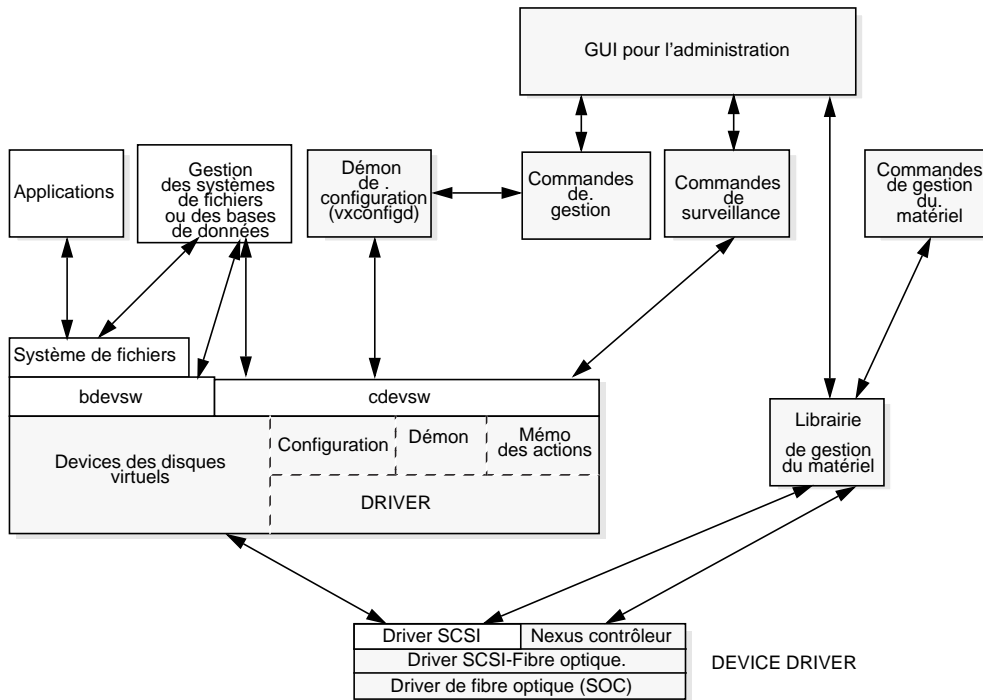
Les applicatifs liés au système d'exploitation : SDS

- **Journaling** – C'est une fonctionnalité qui permet un redémarrage rapide de l'installation. Elle est aussi appelée : UFS logging.
- **RAID** – Redundant arrays of inexpensive disks.
- **growfs** – C'est une commande similaire à `newfs`. Elle permet d'agrandir dynamiquement la taille d'une partition sans perdre les données précédemment stockées. Durant la manipulation, la partition est toujours disponible pour les utilisateurs.

Capacités des logiciels

Les applicatifs liés au système d'exploitation : VM

Volume Manager





Capacités des logiciels

Les applicatifs liés au système d'exploitation : VM

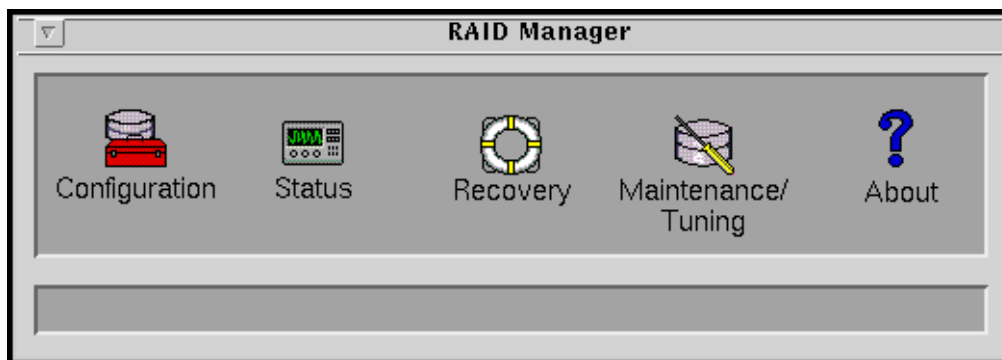
Volume Manager

Ce logiciel propose des fonctionnalités similaires (pour les niveaux de RAID disponibles) à SDS. Il est disponible sur chaque plate-forme SSA.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

Raid Manager



Capacités des logiciels

Les applicatifs liés au système d'exploitation

RAID manager est l'interface graphique utilisateur du Sun Enterprise Network Array 3000. Cette interface a été conçue pour simplifier la façon dont l'utilisateur configure les différents niveaux de RAID, monitore le status de son système disques, et effectue le tuning du SPARCStorage Array. Il offre même une composante assistance lors du remplacement d'un contrôleur, d'un disque ou d'un sous-système défaillant.

RDAC (Redundant Disk Array Controller) est le driver qui gère de façon transparente les chemins d'accès des données sur les deux contrôleurs Ultra SCSI. C'est lui qui prend en charge le re-routage automatique des opérations d'entrées/sorties d'un contrôleur sur l'autre dans le cas de panne d'un des deux Ultra SCSI. RDAC fait partie intégrante de RAID manager.

Performances

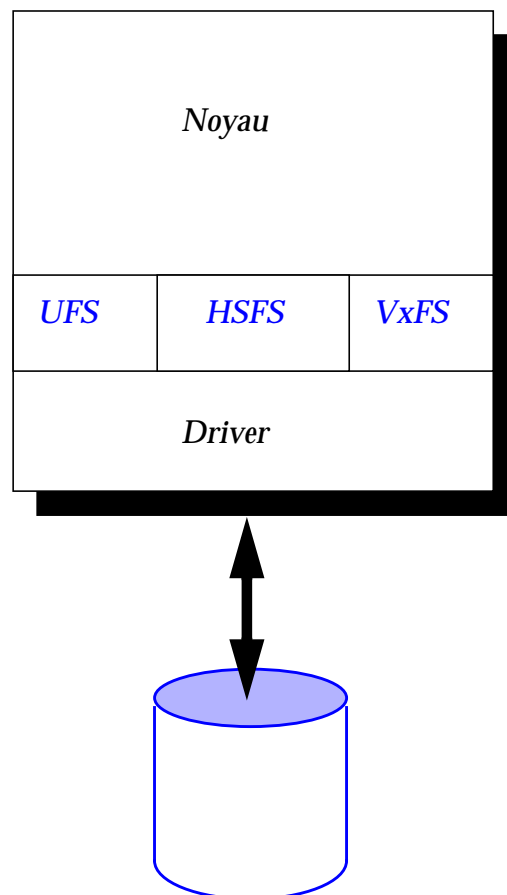
En terme de performance, la bande passante de chaque interface contrôleur/host est de 35 Moctets/sec. En **RAID 5**, les performances mesurées sur chaque interface sont les suivantes :

- random read: 3300 opérations de 4 Ko par seconde (soit un peu plus de 13 Mo/sec)
- random write: 1000 opérations de 4 Ko par seconde (soit 4 Mo/sec)
- soit un débit total de 26 Mo/sec. en lecture aléatoire et de 8 Mo/sec. en écriture aléatoire
- sequential read: 30 Mo/sec avec des blocs de 64 Ko
- sequential write : 10 Mo/sec (100% hits cache) avec des blocs de 64 Ko
- soit un débit total de 60 Mo/sec. en lecture séquentielle et de 20 Mo/sec. en écriture aléatoire.

Capacités des logiciels

Les applicatifs liés au système d'exploitation

Les systèmes de fichiers





Capacités des logiciels

Les applicatifs liés au système d'exploitation

Les systèmes de fichiers

L'administrateur va disposer des systèmes de fichiers natifs au système d'exploitation (Solaris 2.5 et Solaris 2.6). Il peut être amené à utiliser des systèmes de fichiers tierce-partie plus adaptés à des configurations spécifiques (fichiers de tailles importantes, environnement de base de données).



Techniques de surveillance

Isolation des applications

Supprimer les interactions

Algorithme de tuning

Mettre en place une surveillance

Analyser les résultats

Modifications possibles

Reprendre la surveillance

Techniques de surveillance

Isolation des applications

Supprimer les interactions

La politique d'optimisation doit commencer dès la livraison de la machine. Plus l'administrateur possédera de points de repères plus son optimisation sera simplifiée.

Algorithme de tuning

L'algorithme correspond à :

- mettre en place une surveillance pas trop pénalisante pour le système (les commandes et les périodes d'échantillonnage seront choisies judicieusement),
- analyser les résultats .. sur une autre machine, pour ne pas pénaliser le serveur. Cette partie d'analyse doit être faite le plus quotidiennement possible, pour prendre en compte toutes les évolutions possibles,
- en connaissant les mécanismes mis en oeuvre dans l'environnement (voire dans le cadre de travail de l'utilisateur) ... voir les modifications possibles à apporter pour améliorer les performances de l'ensemble,
- reprendre la surveillance ... pour voir si la modification est efficace.



Modifications possibles

Logicielles et matérielles

Élément matériels/ logiciels	Facteur de tuning
Code source	Algorithme, langage, modèle de programmation, compilateur
Exécutable	Variables d'environnement, type de systèmes de fichiers
Base de données	Taille des buffers, indexage
Noyau	Taille des buffers, pagination, configuration
Mémoire	Cache
Disques	Algorithmes des drivers, types de disques
Graphisme	Accélérateurs, système de fenêtrage
CPU	Implémentation du processeur
Multiprocesseurs	Bus, architecture
Réseau	Protocole, hardware



Modifications possibles

Logicielles et matérielles

Le design d'une application ou d'une base de données a des répercussions très importantes sur les performances globales.

Dans le cadre de ce cours, nous allons plutôt nous centrer sur la modification des paramètres liés au système d'exploitation voire aux applications. Il sera aussi important d'utiliser au mieux les outils dont l'administrateur dispose.



Notes

Objectifs

Les sujets couverts par ce chapitre seront les suivants :

- la gestion du noyau,
- la gestion des processus,
- la gestion du swap,
- la gestion des accès disques,
- la gestion des disques,
- la gestion des applicatifs base de données,
- la gestion du réseau,
- une vue rapide sur les problèmes de développement.



Mécanismes internes

Pré-requis nécessaire

Mécanismes mis en oeuvre

Noyau

Gestion des services internes

Gestion des interruptions

Gestion des processus et des threads

Gestion des priorités

Gestion de la zone de swap

Gestion des entrées/sorties

Gestion du système de fichiers

Applicatifs

Gestion des bases de données

Réseau

Les types de transports

Base de données

NFS

WEB

Mécanismes internes

Pré-requis nécessaire

Il est nécessaire de disposer d'un minimum de connaissance sur les mécanismes internes pour pouvoir intervenir sur les paramètres de base de la machine et des applications.

Mécanismes mis en oeuvre

- Noyau

La surveillance s'effectuera sur les dispositifs physiques (gestion des interruptions, gestion des accès disques) et sur les services proposés par la machine (processus, thread, etc.).

- Applicatifs

Le SGBD est un applicatif particulier demandant des ressources non négligeables au niveau du serveur. Une attention particulière lui sera réservée.

- Réseau

Le réseau intervient dans tous les applicatifs clients/serveurs. Il va donc être nécessaire de connaître ses limitations et les actions qui devront être menées pour exploiter au mieux cette ressource.



Gestion du noyau

Noyau

Gestion des services internes

Gestion des interruptions

Gestion des processus et des threads

Gestion des priorités

Gestion de la zone de swap

Gestion des entrées/sorties

Gestion du système de fichiers

Gestion du noyau

Noyau

■ Gestion des services internes

Ces notions sont nécessaires pour comprendre la sortie de la commande `truss`, et les commandes de surveillance des performances du noyau.

■ Gestion des interruptions

Ces notions permettent de surveiller les charges induites par les contrôleurs.

■ Gestion des processus et des threads

Certaines applications proposent deux types d'interfaces, via des processus et via des threads.

■ Gestion des priorités

Il peut être nécessaire de changer la priorité d'une application pour qu'elle puisse utiliser au mieux les ressources du serveur.

■ Gestion de la zone de swap

Il est, à la fois, nécessaire de dimensionner la zone de swap et de pouvoir intervenir sur les paramètres du système afférant à cette gestion.

■ Gestion des entrées/sorties

Le point principal à surveiller est le disque, il est donc nécessaire de surveiller les entrées/sorties dans le noyau et sur les dispositifs physiques.

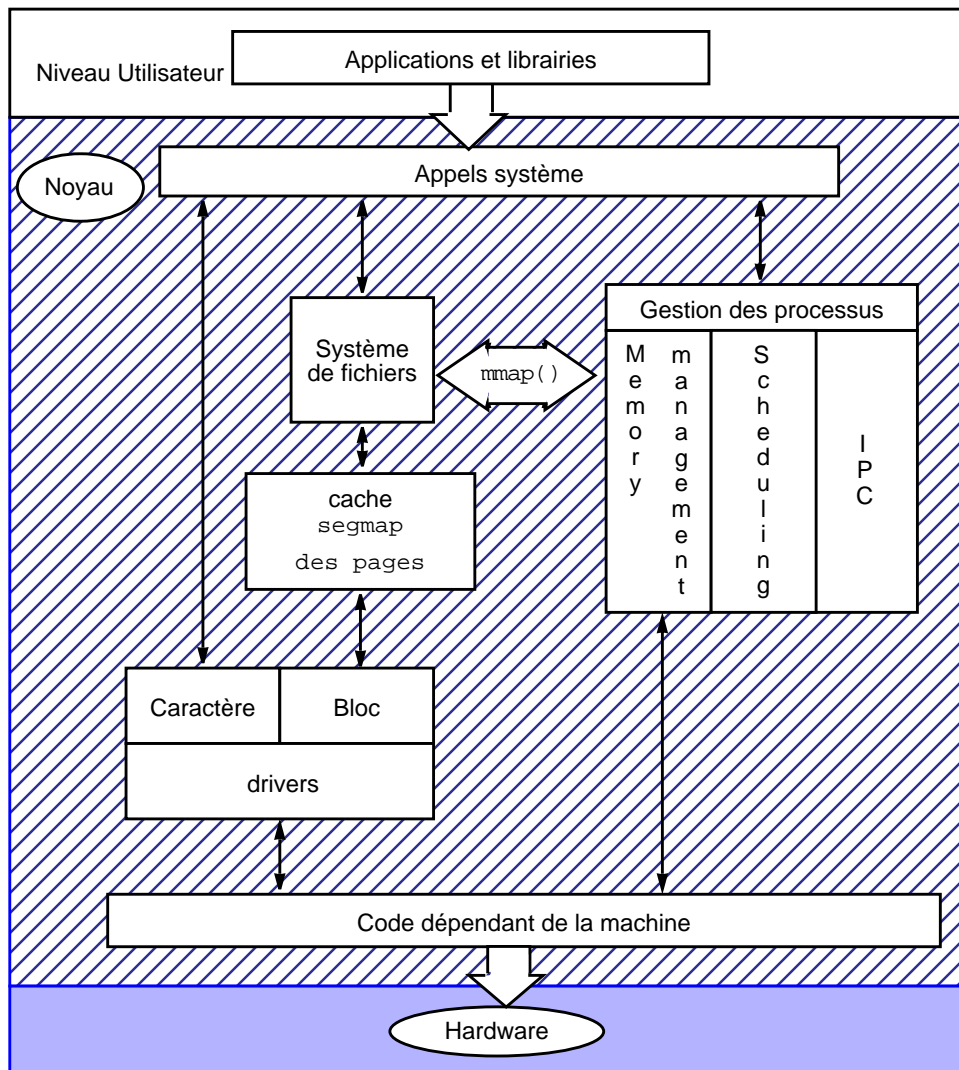
■ Gestion du système de fichiers

Une bonne gestion du système de fichiers permet d'obtenir de bonnes performances tant en local que pour les serveurs NFS.



Gestion du noyau

Fonctionnement interne



Temps d'exécution

Temps SYS, USER et REAL

Gestion du noyau

Fonctionnement interne

Un système d'exploitation est une collection de sous-systèmes qui interagissent les uns sur les autres. Chaque sous-système propose des fonctionnalités que peut exploiter l'utilisateur, via ses applications et/ou via la programmation.

Le noyau est un gestionnaire de ressources (mémoire, entrées/sorties, temps, etc.). Un processus ou un thread demande des services au noyau en fonction des ressources dont il a besoin.

La partie la plus basse du noyau est dépendante du matériel présent dans la machine.

Temps d'exécution

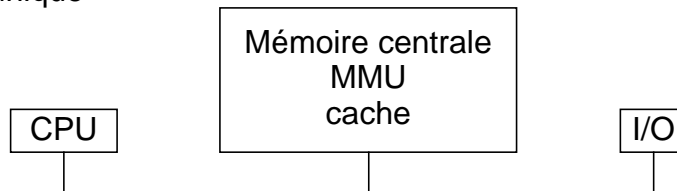
Les statistiques de temps que relèveront les commandes proposent le partage temps en deux variables : le temps `user` et le temps `système`. Le temps `système` est celui passé lors de l'exécution des services demandés au système d'exploitation (temps d'exécution des appels de niveau 2). Le temps `user` est le temps passé dans le code de l'application hormis le temps `système`.



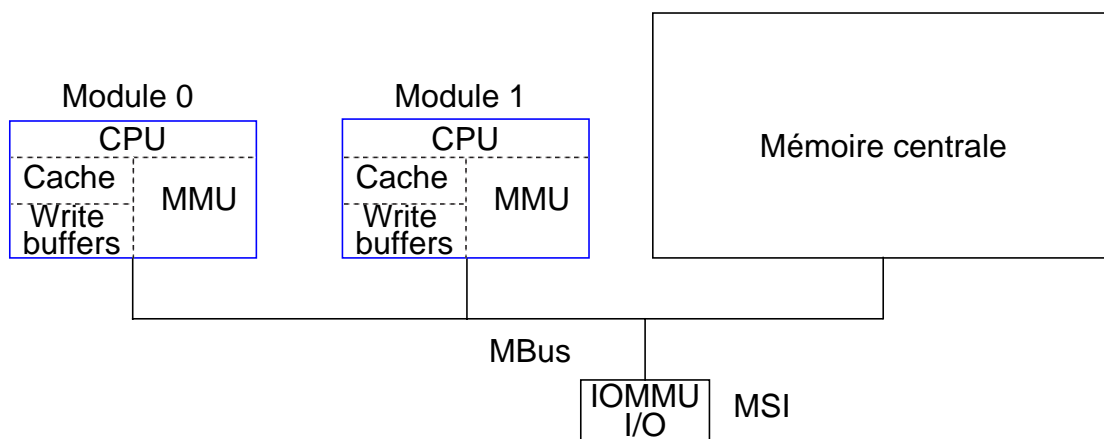
Gestion du noyau

Gestion des processeurs

Processeur unique



Multiprocesseur



Commande

■ mpstat

```
# mpstat
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
0      0  0  0      100  0    2   0   0   0   0   0   1   0  0   0 100
1      0  0  0      3    3    3   0   0   0   0   0   0   0  0   0 100
2      0  0  0      0    0    3   0   0   0   0   0   1   0  0   0 100
3      0  0  0      0    0    3   0   0   0   0   0   2   0  0   0 100
#
```

Gestion du noyau

Gestion des processeurs

Dans un système uniprocasseur, la mémoire et les entrées/sorties sont accédées par une même MMU (memory management unit) qui assure les translations d'adresses.

Dans les systèmes multi-processeurs, chaque système dispose de sa propre MMU et de sa propre mémoire cache. Les entrées/sorties possèdent une MMU qui leur est propre.

Chaque module voit toutes les ressources présentes sur la machine et peut les exploiter selon ses besoins.

Le système d'exploitation gère l'ensemble des processeurs de façon symétrique. Chaque processeur peut utiliser toutes les ressources présentes sur la machine. Il est tout de même possible d'assigner les processeurs à une tâche fixe.

Le système repose donc sur un partage complet des ressources de la machine. Pour assurer une gestion correcte, le système doit donc disposer de verrous assurant un partage cohérent de ses ressources, on parle de MUTEX (mutuelle exclusion). Il est nécessaire de surveiller ces MUTEX pour s'assurer qu'ils ne provoquent pas de point de contention.

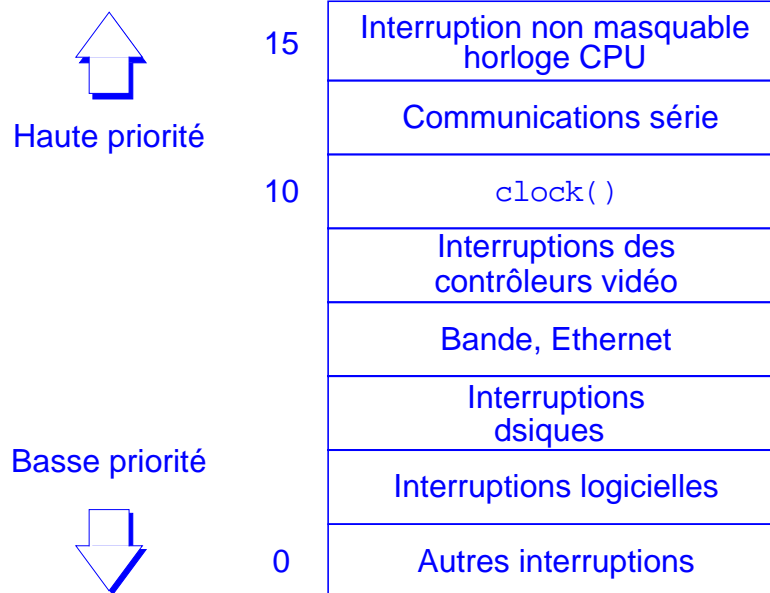
Commande

La commande `mpstat` permet de voir la charge de chaque CPU.



Gestion du noyau

Les interruptions



Commande

■ vmstat -i

```
naxos# vmstat -i
interrupt          total      rate
-----
clock              27923574    100
fdc0                109170      0
-----
Total              28032744    100
naxos#
```

Gestion du noyau

Les interruptions

Les interruptions permettent aux contrôleurs d'indiquer au système qu'un événement doit être traité. Lors de l'arrivée de l'interruption, le système traite un code particulier devant amener un déroutement vers le handler supervisant le matériel.

Commande

La commande `vmstat -i` permet de visualiser les interruptions présentes sur une machine.



Gestion du noyau

Les processus

Définition d'un processus

Implantation mémoire d'un processus

Cycle de vie d'un processus

Commandes

Variables noyau associées

Gestion du noyau

Les processus

Le processus est la première interface que voit l'utilisateur. Il concrétise l'exécution de toute application. Il est nécessaire de connaître certaines de ses caractéristiques pour pouvoir suivre son évolution et comprendre les ressources systèmes qu'il mobilise.

Les caractéristiques que nous allons étudier sont les suivantes :

- qu'est-ce qu'un processus ?
- l'implantation mémoire d'un processus,
- le cycle de vie d'un processus,
- les commandes associées au processus,
- les variables noyau associées aux processus.



Gestion du noyau

Définition d'un processus

<p>u_start Date de départ</p>
<p>u_exdata Informations sur l'exécutable</p>
<p>Arguments de exec()</p>
<p>Ligne de commande</p>
<p>u_cdir - Le répertoire courant</p>
<p>Informations sur les signaux u_signal[MAXSIG], la liste des handlers de signal</p>
<p>u_rlimit[RLIM_NLIMITS] Les limites des ressources</p>
<p>Gestion des fichiers ouverts u_flock - Mutex for next two fields u_nofiles - Number of open files u_flist - Array of uf_entry</p>

```
montreal (sh) # ./pfile 314
314:      dtwm
Current rlimit: 64 file descriptors
 0: S_IFCHR mode:0666 dev:32,24 ino:50 uid:0 gid:3 rdev:13,2
   O_RDONLY
 1: S_IFCHR mode:0666 dev:32,24 ino:50 uid:0 gid:3 rdev:13,2
   O_WRONLY
 2: S_IFCHR mode:0666 dev:32,24 ino:50 uid:0 gid:3 rdev:13,2
   O_WRONLY
 3: S_IFIFO mode:0666 dev:167,0 ino:17 uid:0 gid:0 size:0
   O_RDWR|O_NONBLOCK close-on-exec
 4: S_IFREG mode:0644 dev:32,24 ino:97473 uid:0 gid:0 size:4
   O_WRONLY
   advisory write lock set by process 212
 5: S_IFIFO mode:0666 dev:167,0 ino:18 uid:0 gid:0 size:0
   O_RDWR|O_NONBLOCK close-on-exec
 6: S_IFCHR mode:0000 dev:32,24 ino:61136 uid:0 gid:0 rdev:42,51
```

Gestion du noyau

Définition d'un processus

Un processus correspond à l'exécution d'un code. Il nécessite deux types de zones mémoire pour s'exécuter : une image (zone pour le code) et un `pcb` (fiche d'identité du processus).

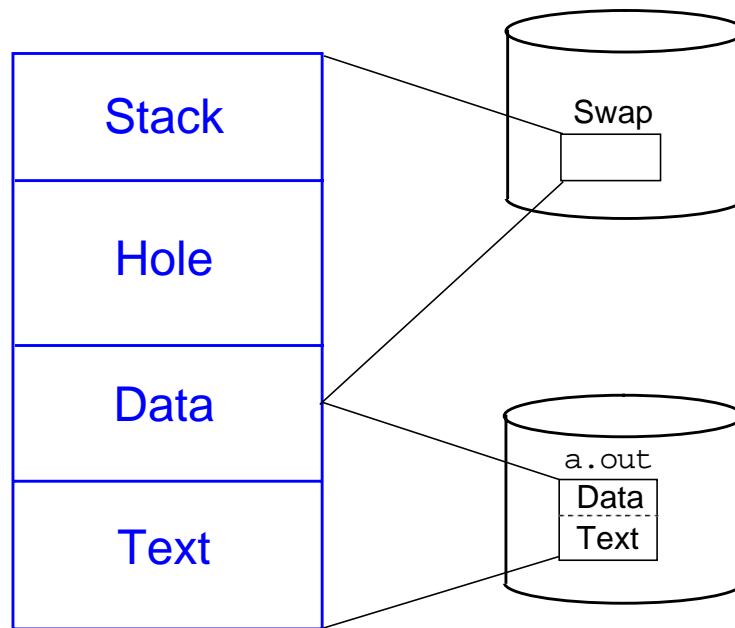
Commandes

Les commandes du répertoire `/usr/proc/bin` permettent d'obtenir des informations sur chaque processus.



Gestion du noyau

Implantation mémoire d'un processus



```
montreal (sh) # ps -edf -o'rss osz addr pmem vsz comm'
RSS  SZ   ADDR %MEM  VSZ  COMMAND
  0   0  f0271ad0 0.0   0    sched
100  103  f5a9f338 0.4   412  /etc/init
  0   0  f5a9ecd8 0.0   0    pageout
  0   0  f5a9e678 0.0   0    fsflush
664  350  f5a9e018 2.2  1400  lpNet
736  434  f5b1e9a0 2.4  1736  /usr/lib/sendmail
756  334  f5b1e340 2.5  1336  /usr/lib/saf/sac
560  421  f5b1d680 1.9  1684  /usr/sbin/rpcbind
484  327  f5b1dce0 1.6  1308  /usr/sbin/in.routed
716  435  f5b1d020 2.4  1740  /usr/sbin/inetd
288  389  f5c35980 1.0  1556  /usr/sbin/keyserv
656  410  f5c35320 2.2  1640  /usr/lib/nfs/statd
684  426  f5c34cc0 2.3  1704  /usr/sbin/kerbd
640  387  f5c34660 2.1  1548  /usr/lib/nfs/lockd
788  628  f5c34000 2.6  2512  /usr/lib/autofs/automountd
708  349  f5c8b988 2.4  1396  /usr/lib/saf/ttymon
696  352  f5c8b328 2.3  1408  /usr/sbin/syslogd
852  398  f5c8acc8 2.8  1592  /usr/sbin/nscd
656  343  f5c8a668 2.2  1372  /usr/sbin/cron
```

Gestion du noyau

Implantation mémoire d'un processus

Le processus demande une allocation de mémoire centrale pour pouvoir s'exécuter.

Les pages allouées à un processus sont soit présentes en mémoire centrale, soit reléguées en zone de swap. Le système d'exploitation conserve en cache, les pages les plus accédées et stocke en zone de swap les pages les moins utilisées.

Page

Des zones mémoires de taille fixe (la taille de la page dépend du matériel, résultat de la commande `pagesize`).

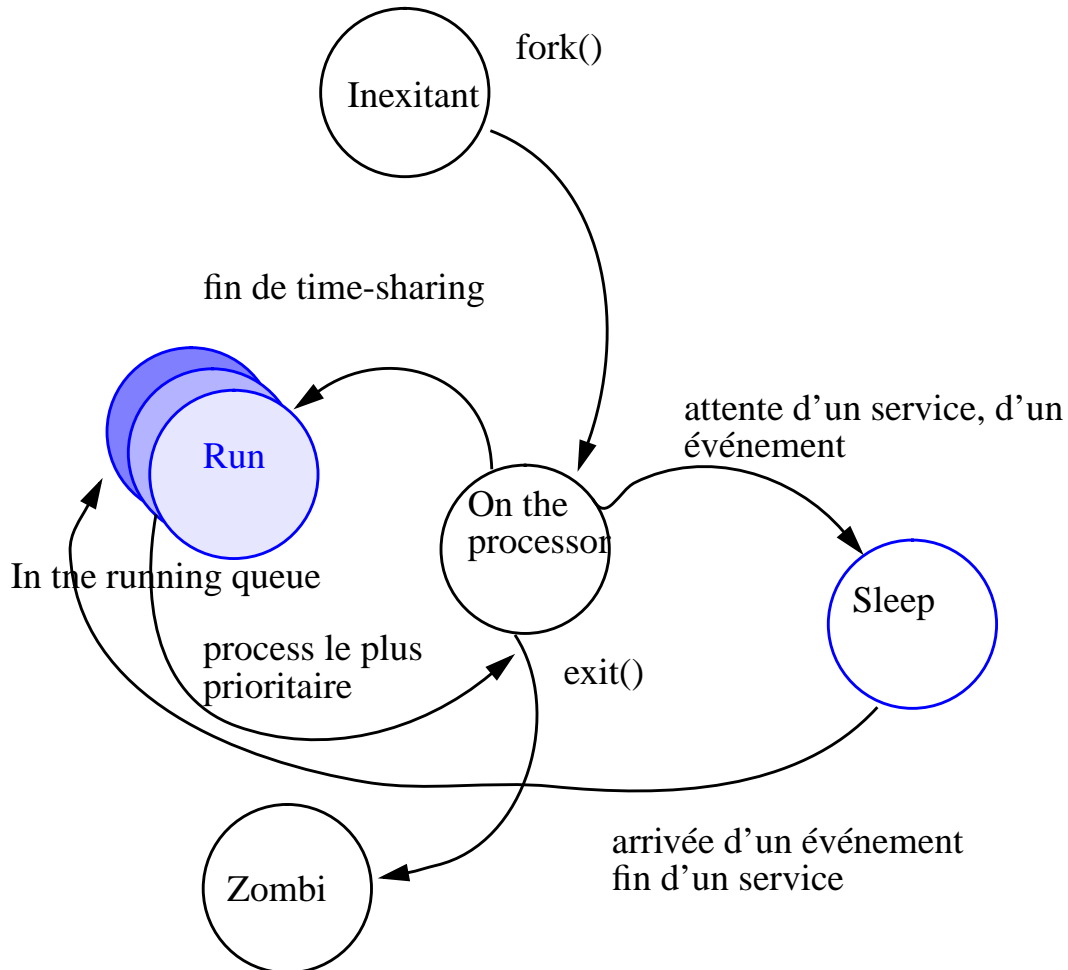
Segment

Un segment est un ensemble contigu de pages.



Gestion du noyau

Cycle de vie d'un processus



```

montreal (sh) # ptree
90  /usr/sbin/in.routed -q
115 /usr/sbin/inetd -s
   306  rpc.ttdbserverd
118 /usr/lib/nfs/statd
120 /usr/lib/nfs/lockd
139 /usr/lib/autofs/automountd
143 /usr/sbin/syslogd
153 /usr/sbin/cron
159 /usr/sbin/nscd
169 /usr/lib/lpsched
   177  lpNet
   183  lpNet
179 /usr/lib/power/powerd
  
```

Gestion du noyau

Cycle de vie d'un processus

Un processus n'existe que lorsqu'une implantation mémoire est définie pour le programme. Pendant son déroulement le processus évolue vers certains états, typiques du système d'exploitation.

Le schéma de la page précédente représente le cycle de vie d'un processus. Il regroupe dans l'état « R », un ensemble de processus en attente du processeur. Le plus prioritaire sera élu et passera en état « O », où il sera réellement en exécution. Ce processus peut suivre plusieurs voies, soit garder le processeur jusqu'à la fin de son time-sharing, soit demander un service au noyau, soit terminer son exécution. Dans le premier cas, il va repasser dans la running queue en fin de tranche de temps. Dans le second cas, le noyau le met dans un état « S », sleeping, en attendant que le service ait abouti. Dans le troisième cas, le noyau le met en état « Z », zombi, et libère son image mais garde son PCB.

L'attente d'un événement s'apparente à la demande d'un service au noyau.

L'environnement multiprocesseurs n'apporte pas de différence fondamentale à ce niveau. Il peut gérer autant de processus en état « O » que la machine dispose de processeurs et qu'il est nécessaire.



Gestion du noyau

Variables associées aux processus

Limites du processus

La commande limit ou ulimit

Scheduling class:	Interactive	
User priority limit:	0	
User priority:	0	
Current nice value:	20	
Interactive mode:	on	
	Current Value	Upper Limit
	-----	-----
CPU Time Limit (ms)	2147483647	2147483647
File Size Limit	2147483647	2147483647
Data Size Limit	2147479552	2147479552
Stack Size Limit	8388608	2147479552
Core File Size Limit	2147483647	2147483647
File Descriptors Limit	64	1024
Maximum Mapped Memory	2147483647	2147483647

Limites des tables noyau

$$\text{maxuprc} = \text{max_nprocs} - 5$$

$$\text{nproc} = \text{nombre de processus actifs}$$

$$\text{max_nprocs} = 16 * \text{maxusers} + 10$$

Gestion du noyau

Variables associées aux processus

■ Limites du processus

Chaque processus peut se voir imposer des limites lors de son exécution. Ces limites sont indiquées aux commandes `limit` ou `ulimit`. Elles permettent de limiter :

- la taille maximum du fichier `core`,
- la taille maximum du heap,
- la taille maximum d'un fichier,
- le nombre maximum de file descriptors,
- la taille maximum de la pile (`stack`),
- le temps CPU maximum,
- la taille maximum utilisée pour le processus.

■ Limites des tables noyau

Les tables noyaux sont aussi limitées pour la gestion des processus (nous parlerons ultérieurement des limites en terme de fichiers).

Cette limite repose sur deux variables `max_nprocs` et `maxuprc`. La première est le nombre maximum de processus pouvant être gérés par le système, la seconde est le nombre maximum de processus pouvant être alloués à un utilisateur.

`nproc` est le nombre courant de processus.

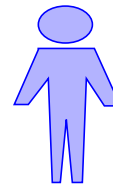


Gestion du noyau

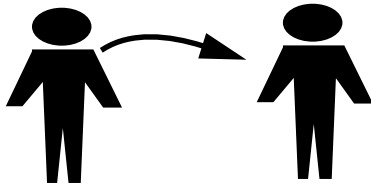
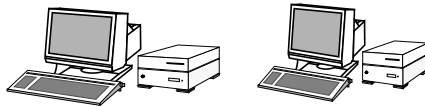
Les threads



Mono-processeur
Mono-processus



Multi-processeurs
Multi-processus ... problème des temps d'attente



Multi-Threads, partage par le même processus de plusieurs CPU

Gestion du noyau

Les threads

Un *thread* représente une unité d'exécution. On peut considérer en première approximation qu'un thread est une *tâche*.

Ainsi, une application sera découpée en une succession de threads.

Le thread n'a pas de réalité au sens UNIX. L'application sera vue comme un seul processus contenant cette succession de Threads. La librairie a pour responsabilité d'ordonner ces threads.

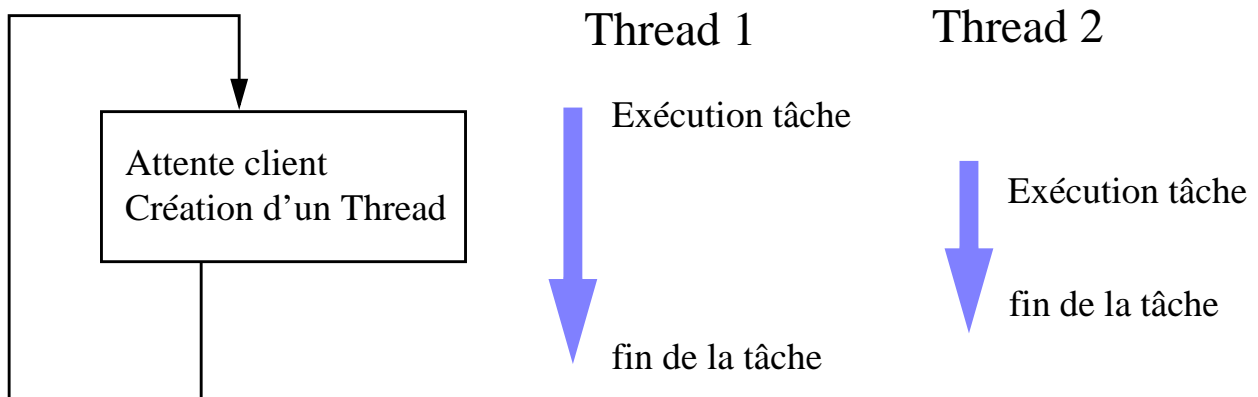
L'avantage de ce type de découpage est que la commutation d'un thread est plus rapide que la commutation d'un processus (commutation, dans ce cas, gérée par le système d'exploitation).

La communication entre threads est aussi très simple, puisqu'ils appartiennent au même processus, ils voient les mêmes variables. La communication est donc implicite.



Gestion du noyau

Client/Serveur concurrent et Threads



Commutation de contexte

```
montreal (sh) # dispadmin -c TS -g | more
# Time Sharing Dispatcher Configuration
RES=1000

# ts_quantum  ts_tqexp  ts_slpret  ts_maxwait  ts_lwait  PRIORITY LEVEL
      200      0      50      0      50      #      0
      200      0      50      0      50      #      1
      200      0      50      0      50      #      2
      200      0      50      0      50      #      3
      200      0      50      0      50      #      4
      200      0      50      0      50      #      5
      200      0      50      0      50      #      6
      200      0      50      0      50      #      7
      200      0      50      0      50      #      8
      200      0      50      0      50      #      9
      160      0      51      0      51      #     10
      160      1      51      0      51      #     11
```

Gestion du noyau

Client/Serveur concurrent et Threads

Lors de l'activation d'une application serveur, nous disposons d'une phase d'attente bloquante sur l'écoute du client. Si un client se présente, le processus serveur doit le prendre en charge. Pour ne pas être bloqué durant tout le traitement de la demande, ce dernier reporte le traitement sur un processus fils ou sur un thread fils.

On parle de service concurrent.

Commutation de contexte

Comme le système est de type « Temps partagé », chaque processus ne dispose que d'un temps fini pour exécuter une partie de son code (cet intervalle de temps est disponible par la commande `dispadmin`).

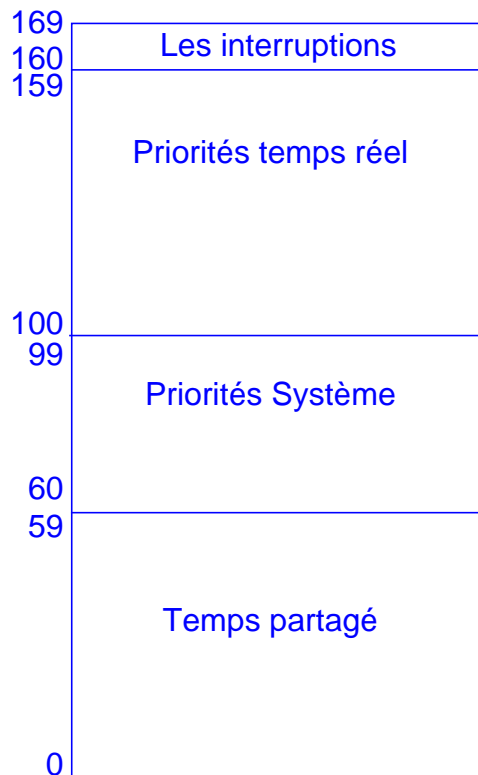
A chaque fin d'intervalle de temps, le système ré-évalue les priorités et élit le processus de plus haute priorité. Pour ce faire, Solaris procède à une commutation de contexte. Cette dernière peut être perturbante pour le système.

Dans le cas d'une application basée sur les threads (comme ces derniers ne sont pas « vus » du système d'exploitation), la commutation de thread n'est pas visible et le processus dispose de tout son quantum de temps.



Gestion du noyau

Les priorités



Commande

```
montreal (sh) # ps -ec | more
  PID  CLS  PRI  TTY      TIME  CMD
    0   SYS  96   ?        0:01  sched
    1   TS   58   ?        0:06  init
    2   SYS  98   ?        0:01  pageout
    3   SYS  60   ?        0:05  fsflush
  184   TS   59   ?        0:00  sendmail
  178   TS   59   ?        0:00  lpNet
  246   IA   59   ?        4:04  Xsun
  247   IA   59   ?        0:00  dtlogin
  248   TS   59  term/a   0:00  ttymon
```

Gestion du noyau

Les priorités

Chaque processus hérite d'une classe de priorité et d'une priorité dans cette classe. Le niveau de priorité détermine le processus qui sera élu pour être exécuté.

Il existe 4 classes de priorités :

- System - SYS
- Timesharing - TS
- Interactive - IA
- Real-time - RT



Gestion du noyau

Les priorités

La priorité temps réel

- la plus haute priorité du système

La priorité IA

- permet d'augmenter les performances des applications interactives
- classe par défaut utilisée dans les environnements graphiques
- augmente la priorité des processus dans la fenêtre active
- diminue la priorité des processus dans la fenêtre non active

Gestion du noyau

Les priorités

Le système dispose donc de 4 priorités pour gérer les processus. Lors de l'activation d'une application le système lui alloue une classe de priorité et une priorité dans cette classe.

Hors de tout environnement graphique, le système alloue la classe « Temps partagé » aux processus utilisateurs. La priorité initiale est la plus haute de la classe et va se modifier en fonction des ressources utilisées par l'application.

Le système utilise pour lui la classe « Système ». Les priorités sont fixes dans cette classe.

■ La priorité temps réel

Elle représente le plus haut niveau des priorités. Elle se situe au dessus de la classe « Système ». Les processus assignés à cette priorité sont non swappable.

Le temps de latence pour le dispatch des processus RT se situe entre 2 et 5 millisecondes en fonction des architectures.

■ La priorité IA

Cette classe est typique des environnements graphiques. Elle place au dessus de la classe « Temps partagé » tout processus qui s'exécute dans une fenêtre graphique.



Gestion du noyau

Gestion du swap

Définition de la mémoire virtuelle

Gestion de la mémoire virtuelle

Gestion de la zone de swap

Mécanisme de pagination

Mécanisme de swapping

Les commandes de visualisation

Gestion du noyau

Gestion du swap

Maintenant, nous allons étudier la gestion des mécanismes de swapping sur Solaris 2.x. Les sujets suivants seront étudiés :

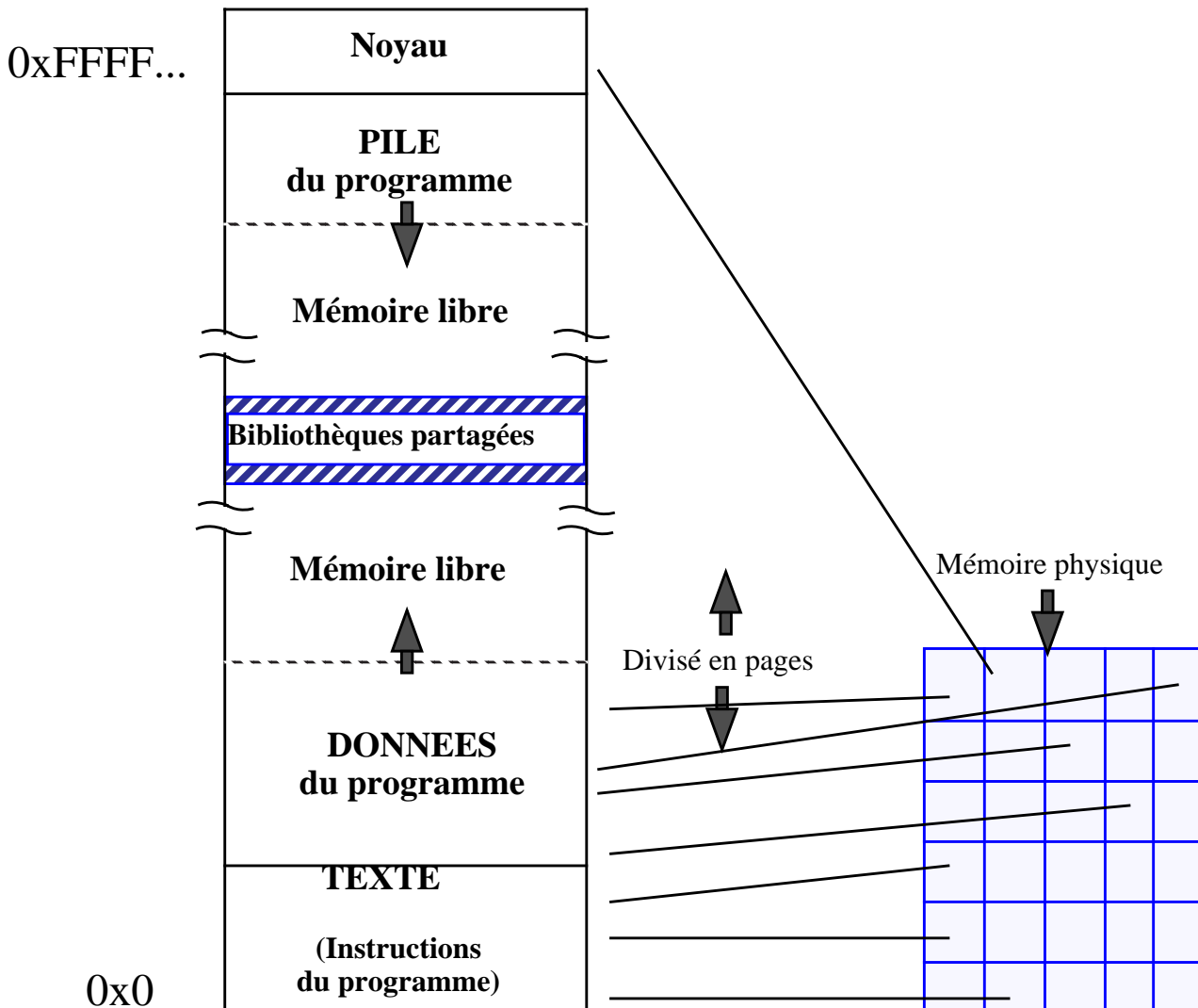
- définition de la mémoire virtuelle,
- gestion de la mémoire virtuelle,
- gestion de la zone de swap,
- mécanisme de pagination,
- mécanisme de swapping.



Gestion du noyau

Gestion du swap

Définition de la mémoire virtuelle



Gestion de la mémoire virtuelle

Gestion du noyau

Gestion du swap

Définition de la mémoire virtuelle

Le système d'exploitation gère une zone de mémoire (dite virtuelle, c'est la somme de la RAM et de la zone de swap) pour l'ensemble de processus et des mécanismes internes qui sont sous son contrôle.

Un processus doit disposer de mémoire pour s'exécuter. Pour cela, le système lui alloue une somme de pages en RAM.

La mémoire totale dont dispose le système d'exploitation est la quantité de RAM présente majoré de la taille de la zone de swap.

Gestion de la mémoire virtuelle

Le système utilise la mémoire virtuelle pour y stocker :

- le noyau,
- des buffers,
- les processus.



Gestion du noyau

Gestion du swap

Gestion de la zone de swap

```
montreal (sh) # prtconf -v | more
System Configuration: Sun Microsystems sun4m
Memory size: 32 Megabytes
System Peripherals (Software Nodes):
...
montreal (sh) # swap -l
swapfile          dev  swaplo blocks   free
/dev/dsk/c0t3d0s1 32,25      8 246232 183104
montreal (sh) #

montreal (sh) #swap -s
total: 39004k bytes allocated + 6996k reserved = 46000k used, 96816k available
montreal (sh) #
```

- Taille de la zone physique de swap :
246232 blocks : 120 M octets
- Taille de la zone de swap utile pour Solaris :
142 M octets
- Taille utilisée pour les processus système : 10 M octets

Gestion du noyau

Gestion du swap

Gestion de la zone de swap

Le système peut nécessiter de la zone de swap pour exécuter certaines applications (en fonction de la quantité de RAM dont dispose la machine).

La commande `swap` permet de visualiser l'espace disponible pour la zone de swap.

En Solaris 2.x, le champ `reservable swap` représente :

- `Reservable swap space = disk space + physical memory - kernel locked pages - 3.5-Mbyte buffer.`

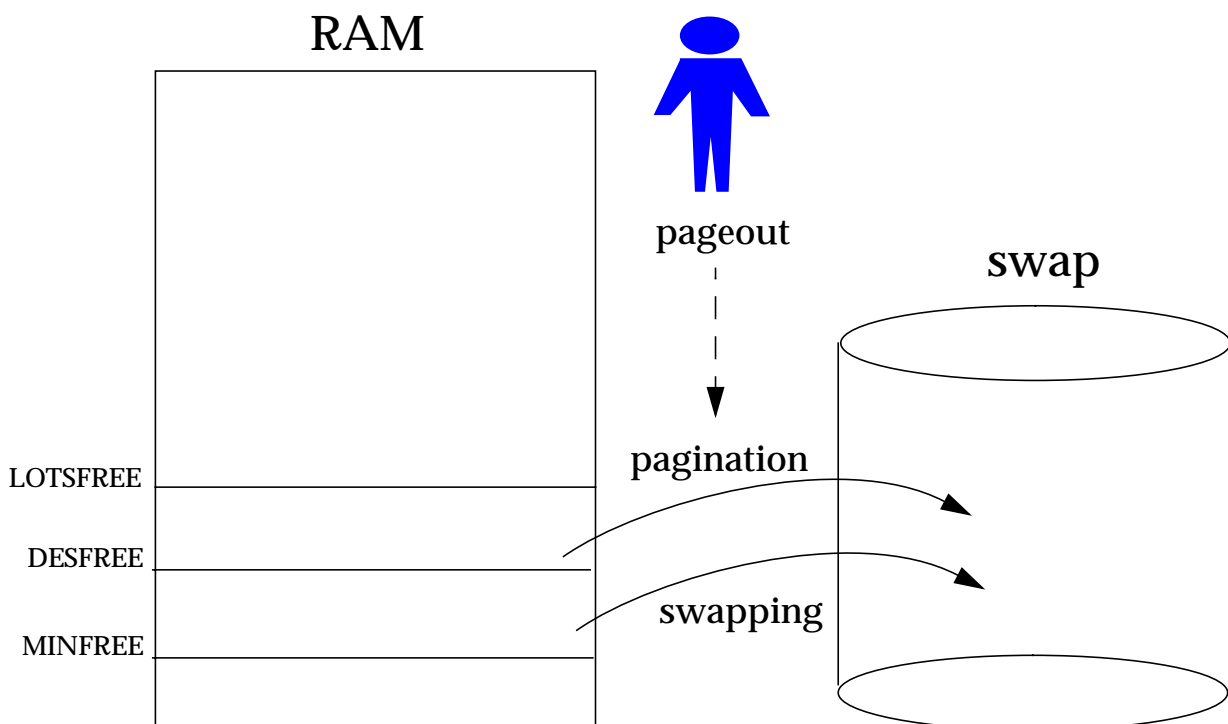
Ainsi, la taille de la zone de swap inclut une partie de la mémoire physique (RAM). Le système ne se réserve une page de swap que lorsqu'une page doit réellement être déportée en mémoire secondaire (et si la page contient des données valides).



Gestion du noyau

Gestion du swap

Mécanisme de pagination



```
montreal (sh) # ps -ecf | grep pageout
  root      2      0  SYS  98   Apr 12  ?           0:01 pageout
  root    9053  8940   IA   48  09:30:46 pts/10   0:00 grep pageout
montreal (sh) #
```

Gestion du noyau

Gestion du swap

Pour libérer la mémoire centrale de la machine, le noyau va intervenir dans un mécanisme de pagination et de swapping.

la pagination consiste à libérer de la RAM page à page via un processus. Le swapping correspond à libérer toutes les pages d'un processus.

Mécanisme de pagination

La pagination repose sur deux mécanismes de base :

- page-in : des pages doivent être présentes en mémoire centrale,
- page-out : les pages peuvent être écrites sur le disque.

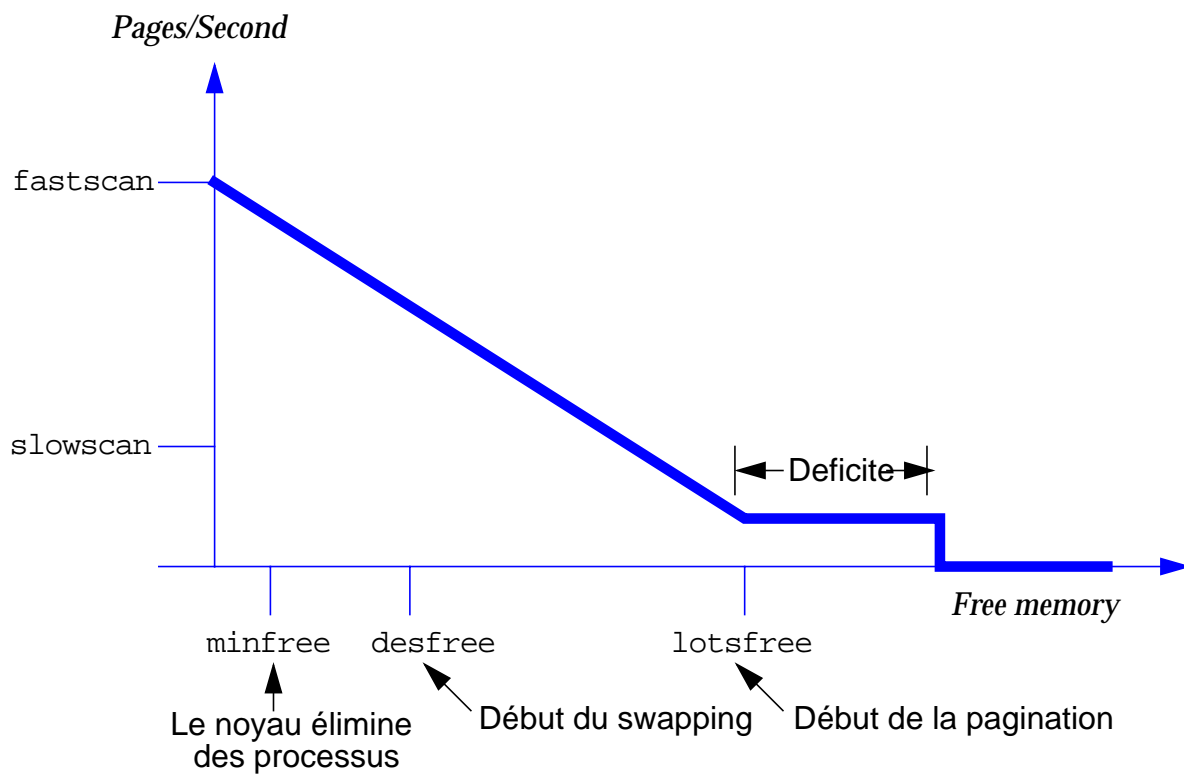


Gestion du noyau

Gestion du swap

Mécanisme de pagination

Enclenchement du mécanisme de pagination



Gestion du noyau

Gestion du swap

Mécanisme de pagination

■ Enclenchement du mécanisme de pagination

Lors de la détection de l'atteinte d'un seuil d'occupation de la mémoire centrale, le système entre en pagination.

Tant que le système dispose de plus de `LOSTFREE` pages libres, aucune action n'est prise. Ce taux d'occupation est calculé toutes les 4 secondes. Lorsque le système atteint cette limite, il enclenche un algorithme de scrutation (voir la page suivante). Cet algorithme consiste à vérifier le nombre de pages libres et à élire celles qui vont devoir être libérées. On parle du mécanisme de pagination. Cette libération de place mémoire aura lieu à une vitesse de `maxpgio` par seconde (nombre de page par seconde libérée vers la mémoire secondaire).

Si les processus continuent à demander des pages mémoires, le système scrute de plus en plus vite les pages en mémoire. Cette vitesse croît linéairement jusqu'à la valeur de `fastscan`.

Si la limite `DESFREE` est atteinte, le système entre dans un mécanisme de swapping. Il va libérer toutes les pages liées à un processus vers la zone de swap.

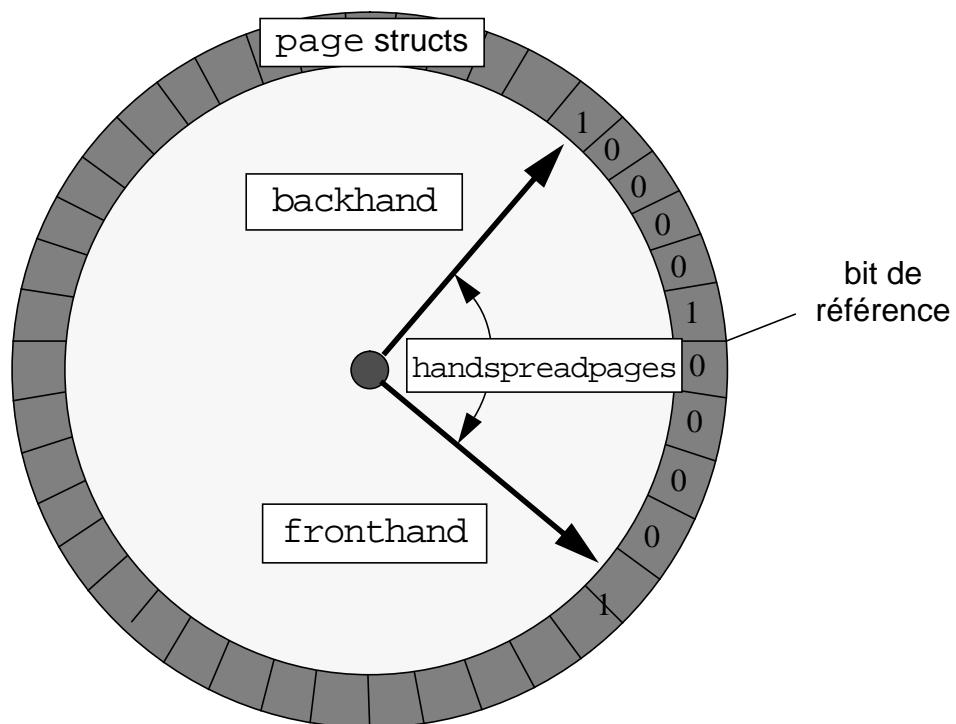
Si la limite `MINFREE` est atteinte, le système ne fournit plus de page mémoire et sort en erreur tout processus nécessitant de la RAM.



Gestion du noyau

Gestion du swap

Mécanisme de pagination



Gestion du noyau

Gestion du swap

Mécanisme de pagination

Le système scrute les pages présentes en mémoire. Il recherche les zones mémoire les moins accédées. Pour cela, il scrute toutes les pages et positionne à « 1 » un flag présent dans le descripteur de page. Ce dernier sera remis automatiquement à « 0 » si la page est accédée par un processus.

La scrutation a lieu à la vitesse de `slowscan pages` par seconde.

Si la page n'a pas été accédée par un processus, elle est mise dans la liste des pages disponibles (on parle de la `freelist`).

Si le processus initial accède à cette page, le système détecte une « erreur » de type `reclaim page fault`, les données sont immédiatement mises à la disposition du processus et aucune entrée/sortie n'a eu lieu. Si un autre processus a écrasé le contenu de la page, le système fournit une nouvelle page au processus initial (des entrées/sorties devront avoir lieu pour recharger cette page), le système parle de `major page fault`.



Gestion du noyau

Gestion du swap

Mécanisme de swapping

- Le processus de swapping

```
montreal (sh) # ps -ecf | grep sched
    0  SYS  96 ?          0:00 sched
   184  TS  58 ?          0:00 lpsched
montreal (sh) #
```

- Le hardswap

- Le soft swap

Gestion du noyau

Gestion du swap

Mécanisme de swapping

- Le processus de swapping

Le swap est un mécanisme de scrutation interne au noyau Unix est validé toutes les secondes. Il se concrétise par le processus `sched` dans la commande `ps`. Son but est de transférer des informations de et vers la mémoire centrale. Si le système nécessite ce type de transfert, il met ce processus en tête de la `dispatch queue`.

Deux types de swapping sont utilisés par le système d'exploitation, on parle de `HARDSWAP` et `SOFTSWAP`.

- Le hardswap

Le système doit libérer de la mémoire primaire (la limite `MINFREE` est atteinte). Tout processus dans un état « S » depuis plus de 2 secondes sera mis en zone de swap (on parle de `swap out`).

- Le softswap

Le système libère de la place en prévision d'une utilisation intensive de la RAM. Tout processus dans un état « S » depuis plus de « `maxslp` » secondes sera mis en zone de swap (on parle de `swap out`).



Gestion du noyau

Gestion du swap

Les variables

- lotsfree
- desfree
- minfree
- slowscan
- fastscan
- maxpgio
- maxslp

Gestion du noyau

Gestion du swap

Les variables

Ces variables peuvent être modifiées dans le fichier `/etc/system`.

Variables	Valeurs	sun4c	sun4u	sun4d
maxpgio	40			
maxslp	20			
pagesize		4096	8192	4096
desfree	1/64 RAM	en page		
minfree	128 k	en page		
lotsfree	1/32 RAM	en page		
slowscan	100			
fastscan	nombre pages/2			



Gestion des accès disques

Types d'accès

Raw device et système de fichiers

Mécanismes internes

Les types de buffers

Description physique des disques

Les limitations physiques des disques

Description des types de systèmes de fichiers

Les types de fichiers

Gestion des accès disques

Types d'accès

- Raw device et système de fichiers

Il est possible d'envisager deux types d'échanges avec le système d'exploitation. Le premier dit «raw device» utilise le support magnétique sans y créer de structure d'accueil, le second est basé sur la création d'une structure d'accueil compréhensible par Unix.

Mécanismes internes

- Les types de buffers

Le noyau met à la disposition des applications des buffers internes en fonction des types d'objets gérés.

Description physique des disques

- Les limitations physiques des disques

Il est nécessaire de connaître les types de disques présents sur une machine pour pouvoir optimiser au mieux ce support qui est le plus lent de la chaîne.

Description des types de systèmes de fichiers

- Les types de fichiers

L'administrateur peut être amené à choisir entre plusieurs organisations pour stocker les objets Unix.



Gestion des accès disques

Types d'accès

Raw device et système de fichiers

- Choix de l'appliatif
- Avantages et inconvénients

Gestion des accès disques

Types d'accès

Raw device et système de fichiers

De façon native, Unix propose deux gestions des supports magnétiques. On parle de raw device ou de systèmes de fichiers.

■ Raw device

Dans ce cas, le support magnétique ne dispose pas de structure d'accueil, et l'application gère une zone de stockage brute.

■ Système de fichiers

Unix construit une structure d'accueil pour les objets de type fichiers, répertoires, etc.

■ Les avantages et inconvénients

Le système de fichiers est plus simple à administrer, toutes les commandes Unix sont à la disposition de l'administrateur. Les mécanismes de bufferisation de tous les objets sont mis en oeuvre lors des accès. Il est donc recommander pour les gestions des zones utilisateurs.

Le raw device ne met en oeuvre que les buffers liés aux entrées/sorties. Il mobilise donc beaucoup moins de ressource au niveau de la mémoire centrale. Il est particulièrement adapté aux applicatifs de type base de données qui disposent de leur propre gestion interne des zones de stockage.

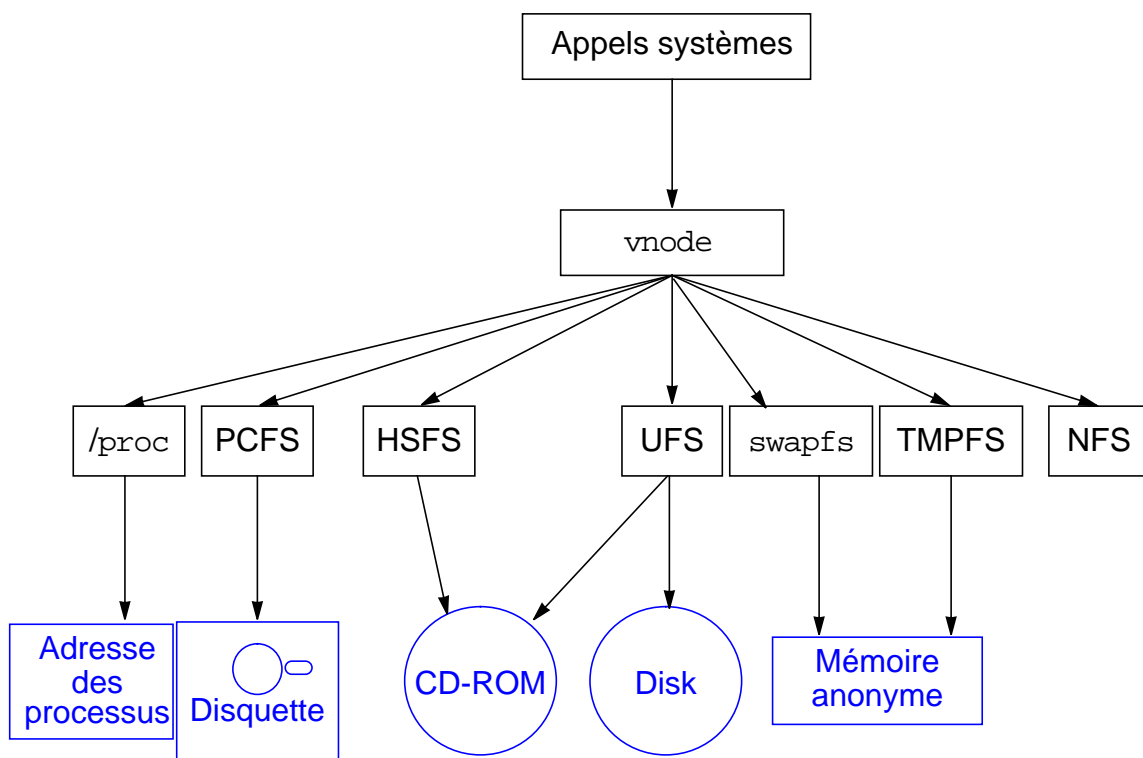


Gestion des accès disques

Mécanismes internes

Bufferisation

- Tables des inodes
- Répertoires
- Buffers des entrées/sorties
- Caches des systèmes de fichiers



Gestion des accès disques

Mécanismes internes

■ Bufferisation

Unix propose une interface standard quel que soit le support de mémorisation (disposant d'une organisation) utilisé. Cette interface simplifie la programmation des applications qui gèrent de la même manière tout accès à tout support physique.

Cette interface permet aussi à Unix de gérer des buffers en fonction des types d'objets gérés. Il va exister 3 grands types de buffers, les buffers de la table des inodes, les buffers des répertoires et les buffers des entrées/sorties.

■ Tables des inodes

Pour ne pas accéder de façon incessante à la table des inodes présente sur les disques, Unix mémorise une partie des inodes en zone cache.

La taille de cette table est `ufs_ninode`.

■ Répertoires

Les contenus des répertoires sont aussi cachés dans une zone cache appelée le « DNLC » (directory name lookup table). La taille de cette table est `ncsize`. Les entrées ne sont mémorisées que pour les fichiers dont les noms font moins de 30 caractères.

■ Buffers des entrées/sorties

Unix utilise toute la mémoire disponible pour zone cache des informations, cette dernière dépend donc de la taille RAM restant à la disposition du système d'exploitation.

■ Cache des systèmes de fichiers

Le nombre maximum de place utilisé par les buffers physiques est de 2% par défaut. Il peut être positionné par la variable `bufhwm`.



Gestion des accès disques

Mécanismes internes

- Tables des inodes

```
tadoussac# adb -k /dev/ksyms /dev/mem
physmem          eb8
ufs_ninode/D
ufs_ninode:
ufs_ninode:      583
maxusers/D
maxusers:
maxusers:        29

tadoussac# netstat -k
inode_cache:
size 1345 maxsize 583 hits 16994 misses 72530 mallocs 1368 frees 0 maxsize reached
1345
puts at frontlist 65881 puts at backlist 8331 queues to free 0 scans 430630

vancouver # adb -k /dev/ksyms /dev/mem
physmem          1e42
ufs_ninode/D
ufs_ninode:
ufs_ninode:      2400
maxusers/D
maxusers:
maxusers:        30

vancouver# netstat -k
inode_cache:
size 2801 maxsize 2400 hits 671 misses 5081 kmem allocs 2991 kmem frees 137
maxsize reached 2856 puts at frontlist 2231 puts at backlist 655
queues to free 0 scans 325603 thread idles 2100 lookup idles 0 vget idles 0
cache allocs 5081 cache frees 2280 pushes at close 0
```

Gestion des accès disques

Mécanismes internes

■ Tables des inodes

Cette table est importante au niveau des serveurs NFS où l'accès aux systèmes de fichiers est permanent.

La valeur de `ufs_ninode` est fonction de `maxusers` :

Valeur de <code>ufs_ninode</code>	2.5	2.6
<code>ufs_ninode</code>	<code>max_nprocs</code> + 16 + <code>maxusers</code> + 64	68 * <code>maxusers</code> + 360



Gestion des accès disques

Mécanismes internes

Bufferisation

- Répertoires

structure dirent

d_ino
d_reclen
d_namlen
d_name[MAXNAMLEN+1]

```
tadoussac# adb -k /dev/ksyms /dev/mem
physmem          eb8
ncsize /D
ncsize:
ncsize:          583
tadoussac#
tadoussac# vmstat -s
 1179234 cpu context switches
 5432157 device interrupts
   82756 traps
 4179720 system calls
 288645 total name lookups (cache hits 68%)
   2402 toolong
  28450 user   cpu
  14822 system cpu
 2002514 idle  cpu
   47921 wait  cpu
tadoussac#
```

Gestion des accès disques

Mécanismes internes

Bufferisation

■ Répertoires

Les répertoires sont des fichiers disposant d'une structure interne particulière. Ils sont divisés en bloc de 512 octets et contiennent les références aux fichiers qu'ils contiennent. Leur structure interne repose sur un tableau décrit à la page précédente.

Sur le disque, ils peuvent changer de taille en fonction du nombre de fichiers qui y sont stockés. Lors de la destruction de fichiers, le système libère la place inoccupée du tableau, lors de la prochaine allocation de place (création d'un nouvel objet).

Pour ne pas avoir à scruter en permanence le contenu des répertoires, le système mémorise un contenu succinct de ces derniers en mémoire.

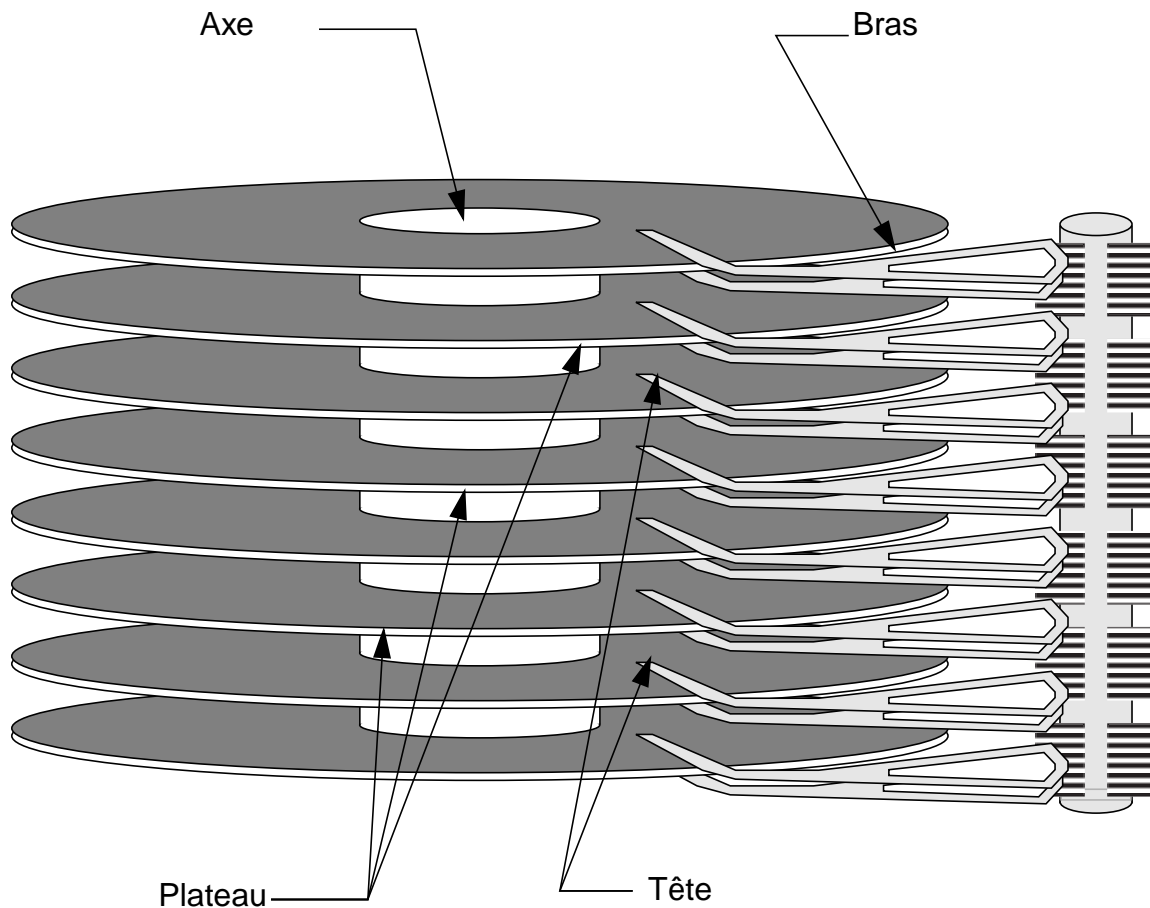
La grandeur de la zone cache est fournie par la variable `ncsize`.

L'occupation de cette zone est fournie par la commande `vmstat -s`.

Par défaut, la valeur de `ncsize` est la même que la valeur de `ufs_ninode`.

Gestion des accès disques

Description physique des disques



Taille	Bande passante	Temps	Temps SCSI
2 ko	118 ko/s	17 ms	0,1 ms
8 ko	432 ko/s	19 ms	0,4 ms
64 ko	2025 ko/s	31 ms	3 ms
1 Mo	2834 ko/s	262 ms	40 ms

Gestion des accès disques

Description physique des disques

Le disque est formé d'un ensemble de plateaux découpé en pistes, elles-mêmes découpées en secteur.

Un certain nombre de paramètres sont importants pour prendre en compte le temps d'accès au support physique :

- la vitesse de rotation du disque,
- le déplacement de la tête (seek time),
- le temps de transfert de l'information sur le disque.

Nous ne prenons pas en compte, à ce niveau, le temps pris par le système d'exploitation pour donner l'ordre de transfert, ni le temps de transfert vers le périphérique (se référer au premier chapitre).

Par exemple :

Nous disposons d'un disque sur bus SCSI. Ce disque tourne à la vitesse de 5400 t/mn. Son positionnement de tête se déroule en 11 ms, et le temps de transfert est de 4168 ko/s (voir le fichier `format.dat`).

Le temps de transfert de l'information est donc :

temps de positionnement de la tête +
temps de positionnement sur le secteur (1/2 tour en moyenne) +
temps de transfert de l'information.



Gestion des accès disques

Les types d'accès

- Les accès en lecture/ les accès en écriture
- Les accès synchrones/ les accès asynchrones
- Les accès séquentiels/ les accès random
- Les synchronisations

Gestion des accès disques

Les types d'accès

■ Les accès en lecture/ les accès en écriture

Les disques disposent de plus en plus de zones caches, directement intégrées dans le disque ou dans le périphérique (voir les périphériques de type SSA et AS 5000). Ces zones caches servent lors des accès en écriture et non lors des accès pour la lecture. L'accounting permet d'indiquer le nombre de lecture effectuée par une application et le nombre de blocs transféré sur le disque. Ainsi, nous voyons les types de transferts principalement utilisés par application.

■ Les accès synchrones/ les accès asynchrones

Certaines applications gèrent les accès de façon synchrone via des appels systèmes de type `read/write`. D'autres utilisent les accès asynchrones via les appels `aioread, aiowrite`. Ce qui est principalement le cas dans les environnements base de données.

■ Les accès séquentiels/ les accès random

Dans le cas d'accès NFS (cas des répertoires d'accueil de développeurs C, par exemple), les accès sont de type aléatoire. Dans le cas de traitement de fichiers séquentiels (cas des images), les accès sont de type asynchrone.

■ Les synchronisations

La synchronisation peut avoir lieu par l'application (par l'utilisation d'appels systèmes de type `fsync()`, pour par la demande d'un checkpoint, dans une base de données), ou via le processus `fsflush` qui synchronise les buffers. Ces accès sont alors synchrones.

Il est important de connaître les types d'accès principalement utilisés au niveau d'un serveur, des choix de tailles de stripping et de type de systèmes de fichiers (ou d'options de montage) vont en résulter.



Gestion des accès disques

Les types d'accès

- Le processus fsflush

```
tadoussac# adb -k /dev/ksyms /dev/mem
physmem          eb8
autoup/D
autoup:
autoup:          30
tune_t_fsflushr/D
tune_t_fsflushr:
tune_t_fsflushr: 5
```

Gestion des accès disques

Les types d'accès

- Le processus `fsflush`

Ce processus synchronise les accès aux objets toutes les « `tune_t_fsflushr` » secondes. Il dispose d'un intervalle complet de « `autoup` » secondes pour synchroniser tous les accès aux systèmes de fichiers.

Sur des systèmes disposant de beaucoup de systèmes de fichiers, ce processus peut être perturbant, il est possible de modifier les 2 paramètres précédemment nommés.



Gestion des accès disques

Description des types de systèmes de fichiers

- UFS 2.5
- UFS 2.6
- VXFS

Gestion des accès disques

Description des types de systèmes de fichiers

L'administrateur dispose de trois systèmes de fichiers différents. Il s'agit de :

- UFS 2.5
- UFS 2.6
- VXFS

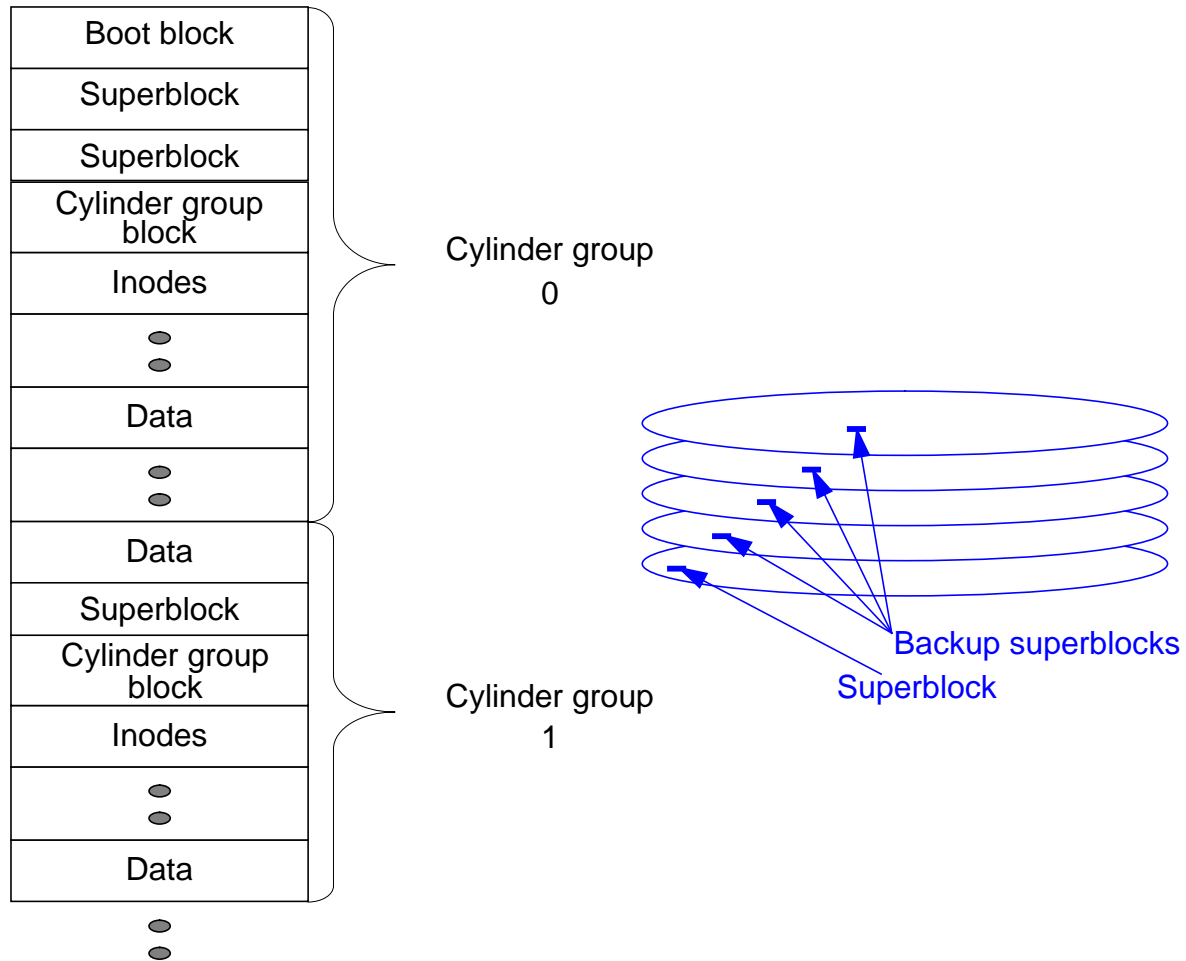
Chacun possède des organisations qui lui sont propres et qui correspondent à des utilisations différentes.



Gestion des accès disques

Les systèmes de fichiers natifs

UFS 2.5 et UFS 2.6



```
tadoussac# fstyp -v /dev/rdisk/c0t0d0s0 | more
/dev/rdisk/c0t0d0s0: Invalid argument
ufs
magic      11954      format  dynamic time      Sun Apr 19 17:55:55 1998
sblkno    16         cblkno  24         iblkno   32         dblkno504
sbsize    2048      cgsize  2048      cgoffset 40      cgmask    0xffffffff0
ncg       51         size   409752   blocks   384847
bsize    8192     shift  13        mask     0xffffe000
fsize    1024      shift  10        mask     0xfffffc00
```


Gestion des accès disques

Les systèmes de fichiers natifs

Les systèmes de fichiers UFS sont dits Fast File System. Ils découpent le disque en groupe de cylindres, et gère chacun de ces regroupements.

Il est possible d'obtenir les caractéristiques d'un système de fichiers via la commande `fstyp`.

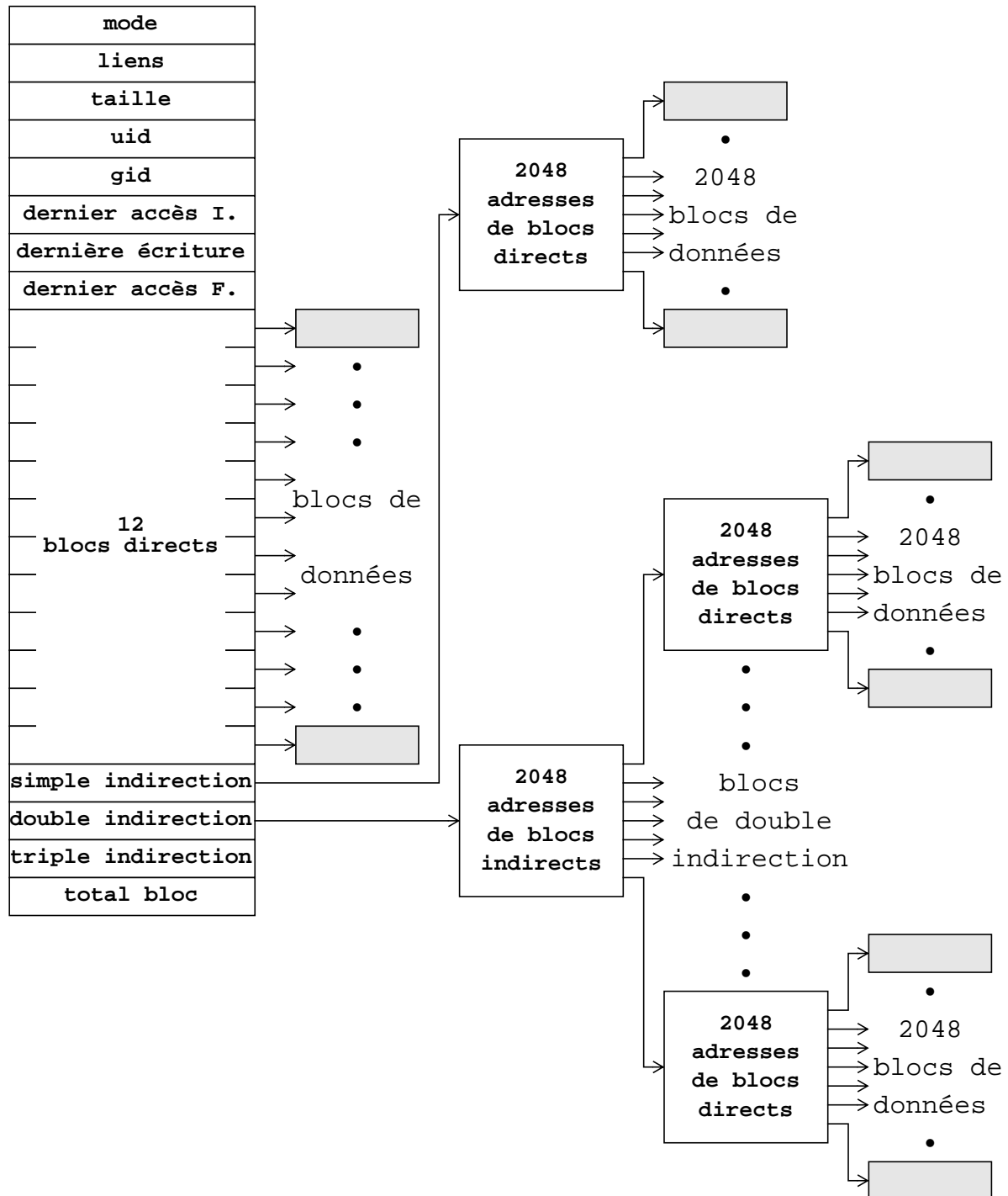
Un flag présent en en-tête permet de s'abstenir d'effectuer la phase de vérification de cohérence lors du redémarrage de la machine. Cette phase pouvant être longue et donc perturbante pour des machines ayant à re-démarrer le plus rapidement possible.

Si le système de fichiers a été démonté normalement, le `flag` est positionné à `FSCLEAN`, et le programme `fsck` n'est pas utile. Quand un système de fichiers est monté, le `flag` est positionné à `FSACTIVE`. Les interruptions, comme `STOP-A`, amènent donc une activation du programme `fsck`. Les valeurs des marques (`flags`) sont :

FSACTIVE	Le système de fichiers peut être incohérent ; il doit être vérifié par <code>fsck</code> et ne peut être monté dans cet état en <code>read/write</code> .
FSCLEAN	Le système de fichiers est cohérent et a été démonté proprement.
FSSTABLE	Le système de fichiers n'a pas changé depuis le dernier <code>sync</code> ou <code>fsflush</code> . La commande <code>fsck</code> ne détecte pas d'incohérence mais en cas d'arrêt brutal du système, l'utilisateur peut perdre quelques données.
FSBAD	Le système de fichiers <code>root</code> a été monté alors qu'il n'était ni <code>FSCLEAN</code> ni <code>FSSTABLE</code> . Il est monté en <code>read-only</code> .

Gestion des accès disques

Les systèmes de fichiers natifs



Gestion des accès disques

Les systèmes de fichiers natifs

Les systèmes de fichiers 2.5 et 2.6 possèdent les caractéristiques suivantes :

- ils gèrent des blocs de données de 8 K octets,
- ils laissent 10% d'espace libre pour ne pas avoir à gérer des structures trop pleines et trop désorganisées,
- ils se réservent une inode par 2 k octets,
- ils gèrent des groupes de cylindres de 16 cylindres contigus,
- ils ne disposent pas de log (journaling),
- leurs accès sont bufferisés par le système. Ainsi, lors d'une écriture par une application, le système d'exploitation charge ses buffers internes en fonction du contenu du disque, puis recopie ses buffers dans la zone utilisateur. Cette double recopie peut être évitée par les applications, si elles utilisent les appels de type `mmap`, ou par l'administrateur si ce dernier utilise l'option `forecedirectio` lors de la commande de montage.



Gestion des accès disques

VXFS

- Allocation par extent
- extents attributs
- fast file system recovery
- online administration
- online backup
- il existe une API, mais le comportement doit être identique pour les applications l'utilisant ou non
- augmente les performances des I/O synchrones
- supporte des fichiers supérieurs à 2 Go
- supporte les quotas
- supporte les ACL
- permet une allocation dynamique des inodes

Gestion des accès disques

VXFS

Ce système de fichiers est propriétaire et est un produit de la société Veritas.

■ Allocation par extent

Par défaut, la taille des blocs est de 2 K pour les systèmes de fichiers de moins de 8 G, 4 K pour les FS de moins de 32 G, et 8 K pour les plus grands. Ce bloc est le bloc logique traité par défaut.

Il gère les accès par des inodes de type UFS, sauf que dans les 10 premières adresses on trouve l'adresse de l'extent et sa longueur. Puis, il gère deux autres indirections. La commande `vxtunefs` permet de visualiser le type du système de fichiers.

■ Extents attributs

VXFS gère les fichiers par groupe d'extents. On peut choisir la politique de l'allocation des extents via des commandes de type `setext` et `getext`.

■ Online administration

■ défragmentation on line.

il faut utiliser la commande `fsadm` qui supprime les places laissées libres dans les répertoires, range de façon contiguë les petits fichiers, consolide les blocs libres. il est conseillé de mettre cette commande en `crontab`.

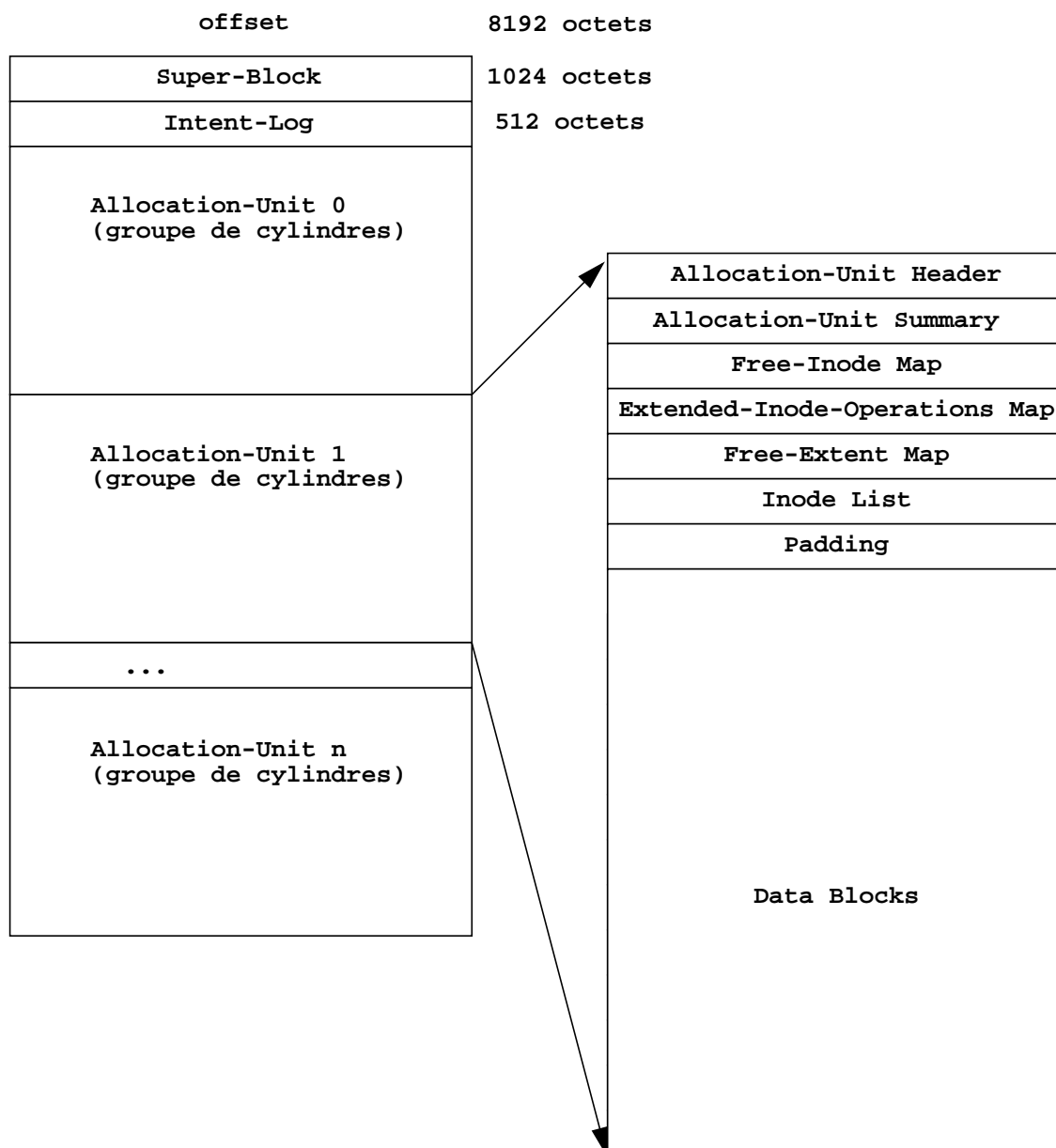
- changement de la taille des systèmes de fichiers
- online backup

Il s'appuie sur une technique de snapshot.



Gestion des accès disques

VXFS



Gestion des accès disques

VXFS

Le *Super-Block* contient :

- le type de système de fichiers,
- les dates de création et de modification,
- des informations sur le label,
- des informations sur la taille et le partitionnement,
- le total des ressources utilisables,
- le numéro de version du partitionnement de disque du système de fichiers.

Il a toujours la même taille de 1024 octets, et se trouve toujours à 8192 octets du début du système de fichiers.

Il en existe des copies dans l'*Allocation Unit Header*. Ces copies peuvent être réutilisées via *fsck*.

L'*Intent Log* est une zone temporaire utilisé notamment afin de garantir l'intégrité du système de fichiers lui-même.

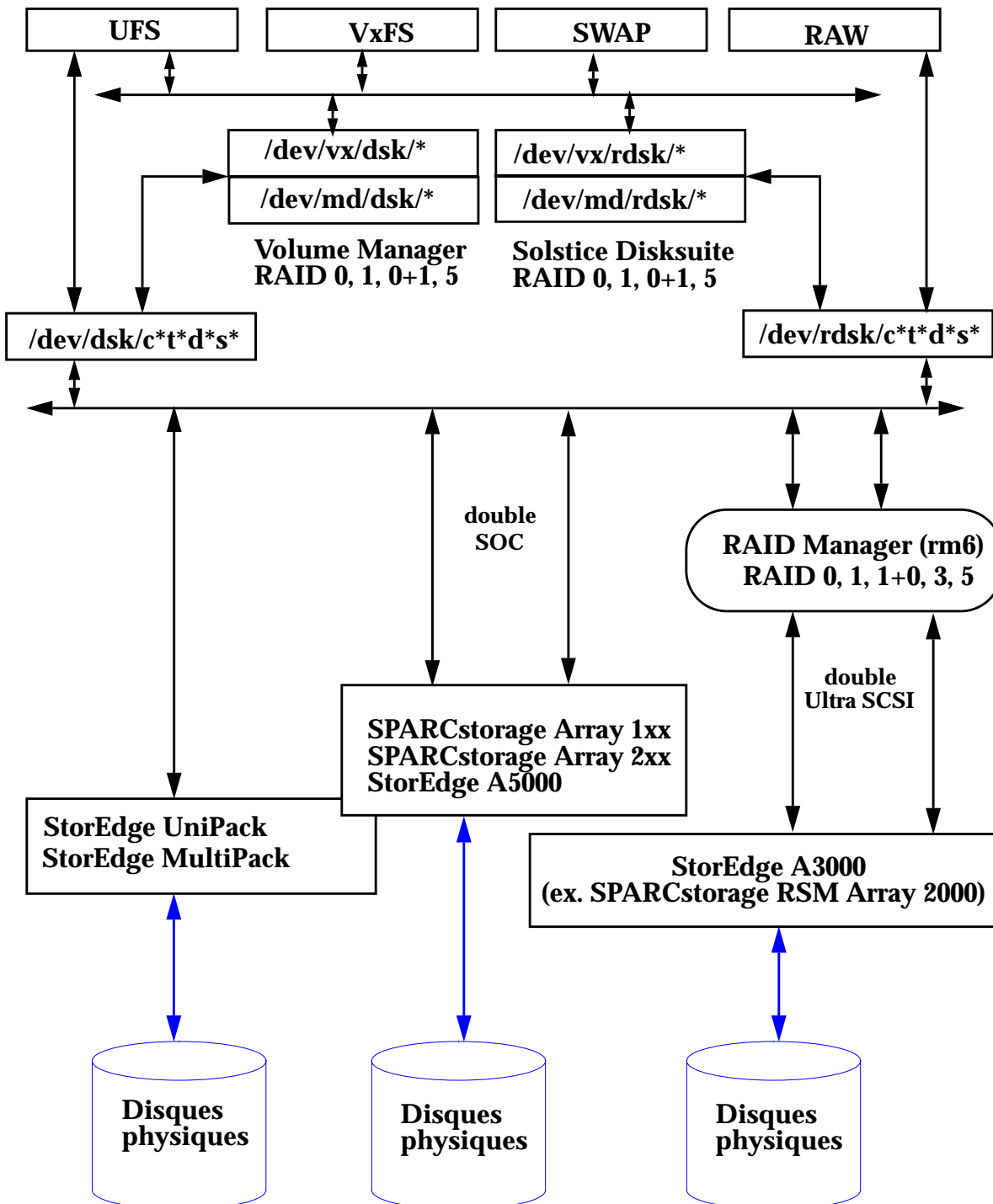
Sa taille par défaut est de 512 blocs. Si le système de fichiers est inférieur à 4Mo, la taille de cette zone est automatiquement réduite par la commande *mkfs*. L'*Intent Log* contient les enregistrements des intentions du système à mettre à jour la structure du système de fichiers. La mise à jour de cette dernière est appelée *transaction*. Elle est elle-même divisée en sous-fonctions pour chaque structures de données.

L'unité d'allocation de VxFS correspond au groupe de cylindres d'UFS.



Gestion des accès disques

Interaction entre les produits UNIX/APPLICATIONS



Gestion des accès disques

Interaction entre les produits

L'administrateur dispose de l'ensemble de produits suivants pour gérer son équipement. Il est nécessaire de disposer d'une vue globale de l'ensemble pour obtenir une plate-forme cohérente et performante.



Applicatifs base de données

Gestion des bases de données

Les bases de données

- Oracle, Sybase, Informix, Ingres

Les 3 types d'interventions

- Design
- Codage des applications
- Administration système et de la base

Applicatifs base de données

Gestion des bases de données

Les bases de données

- Oracle, Sybase, Informix, Ingres

Chaque logiciel propose le moteur de la base, des applications développées, des outils d'interrogations, des générateurs d'applications et des outils propres au tuning de la base de données.

Les 3 types d'interventions

Le développement d'une plate-forme portant une application basée sur un SGBD suit les étapes :

- design de la base (choix de l'architecture, implantation sur une ou plusieurs machines, type de tables). Ce choix est responsable de 60 % des performances,
- codage des applications : il est du ressort du développeur de fournir un code prenant en compte les choix d'architectures effectués précédemment. Ici, la connaissance des mécanismes internes d'optimisation des moteurs de SGBD est important,
- administration de la base de données. Nous regroupons ici les tâches d'administration pure de la base et celles du système. Ces dernières étant intimement liées.

Nous resterons le plus général possible pour pouvoir s'adapter à chaque type de moteur présent sur les serveurs.



Applicatifs base de données

Place de la base dans le système

Les mécanismes internes Unix nécessaires

Fonctionnement d'une base de données

Applicatifs base de données

Place de la base dans le système

Dans un premier temps, nous allons positionner ce logiciel par rapport au système d'exploitation.

Les mécanismes internes Unix nécessaires

Les noyaux Unix sont sous-dimensionnés pour pouvoir prendre en compte ce type d'application, il est donc nécessaire de reprogrammer le système pour qu'il puisse être utilisé.

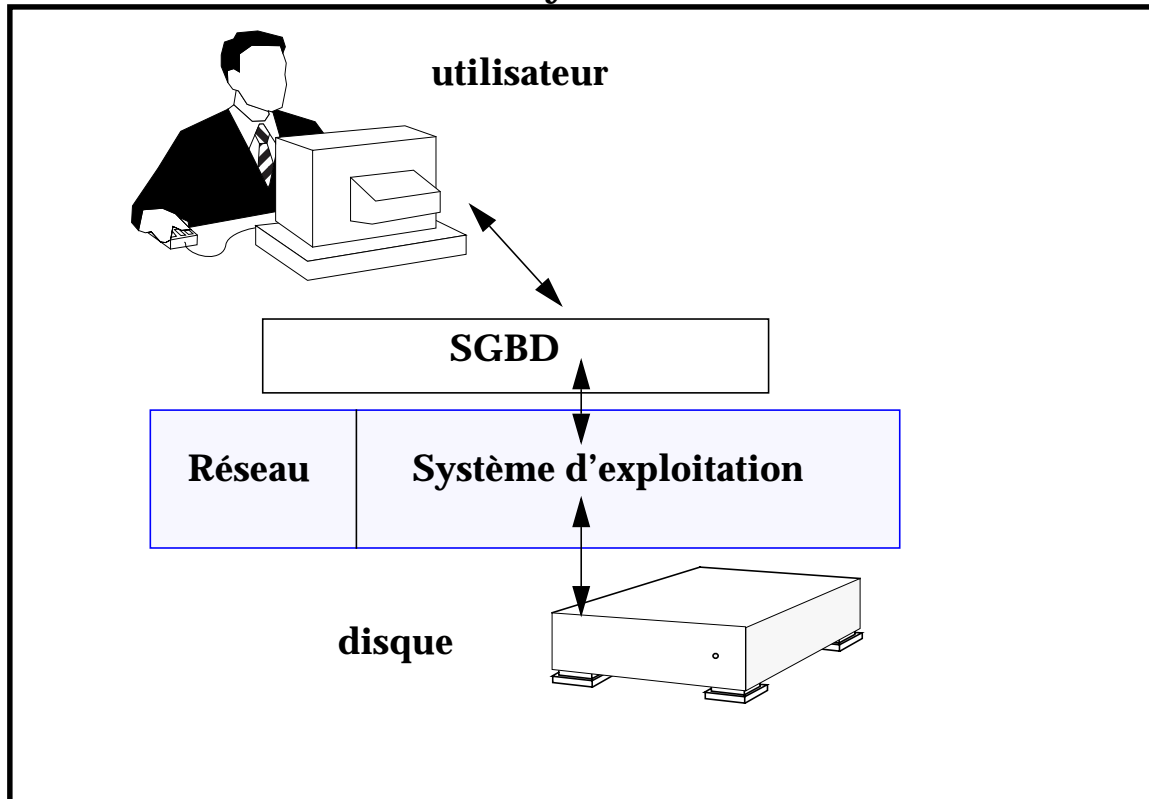
Fonctionnement d'une base de données

Il est nécessaire de connaître des rudiments du fonctionnement d'une base de données pour pouvoir proposer les meilleures ressources à ce logiciel.



Applicatifs base de données

Place de la base dans le système



Utilisation des ressources disques

Applicatifs base de données

Place de la base dans le système

Le SGBD apparaît comme un logiciel applicatif porté par un système d'exploitation.

Comme toute application, son installation et sa gestion sont intimement liées au système d'exploitation de la machine cible.

Le SGBD s'appuie sur les *mécanismes internes* du noyau du système d'exploitation pour assurer la gestion de la base de données.

Utilisation des ressources disques

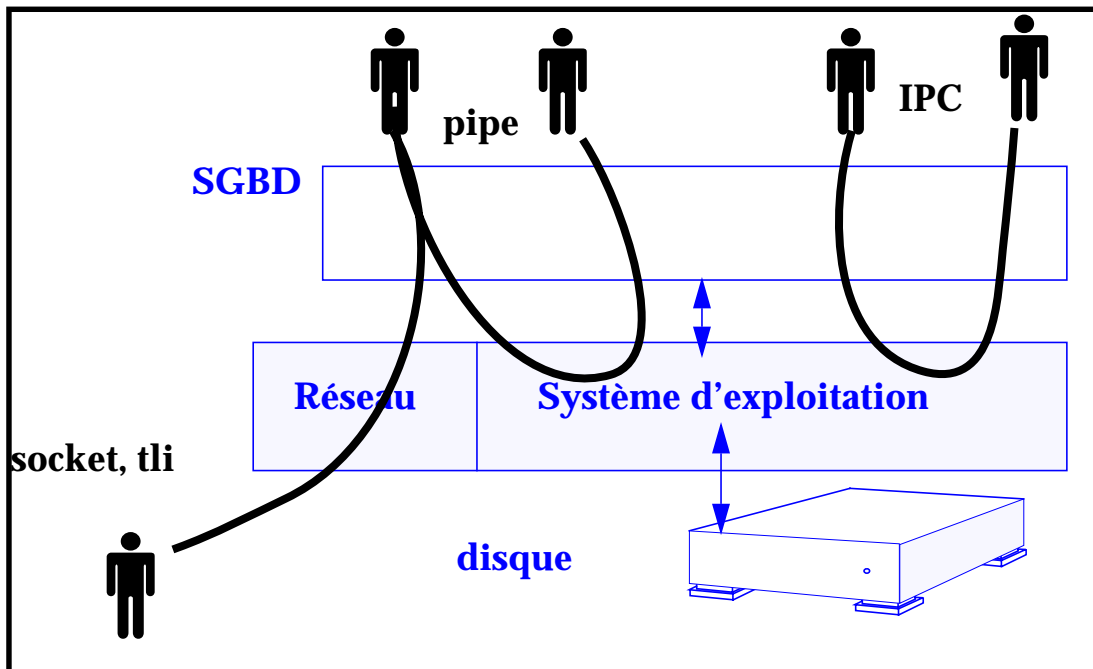
Le logiciel demande deux zone disques distinctes :

- pour stocker le logiciel,
- pour stocker les données.



Applicatifs base de données

Mécanismes internes Unix



Applicatifs base de données

Mécanismes internes Unix

Le SGBD est un logiciel multi-tâches (ou multi-threadé) qui utilise des services internes au système d'exploitation de la machine cible. Il utilise les mécanismes internes suivants :

- les processus,
- les threads,
- les pipes anonymes,
- les IPC,
- les sockets et tli.

Les performances sont liées aux ressources suivantes :

- les commutations de contextes,
- threads,
- zone de swap,
- buffers liés aux entrées/sorties,
- espace disque en Raw Device,
- transferts réseau.



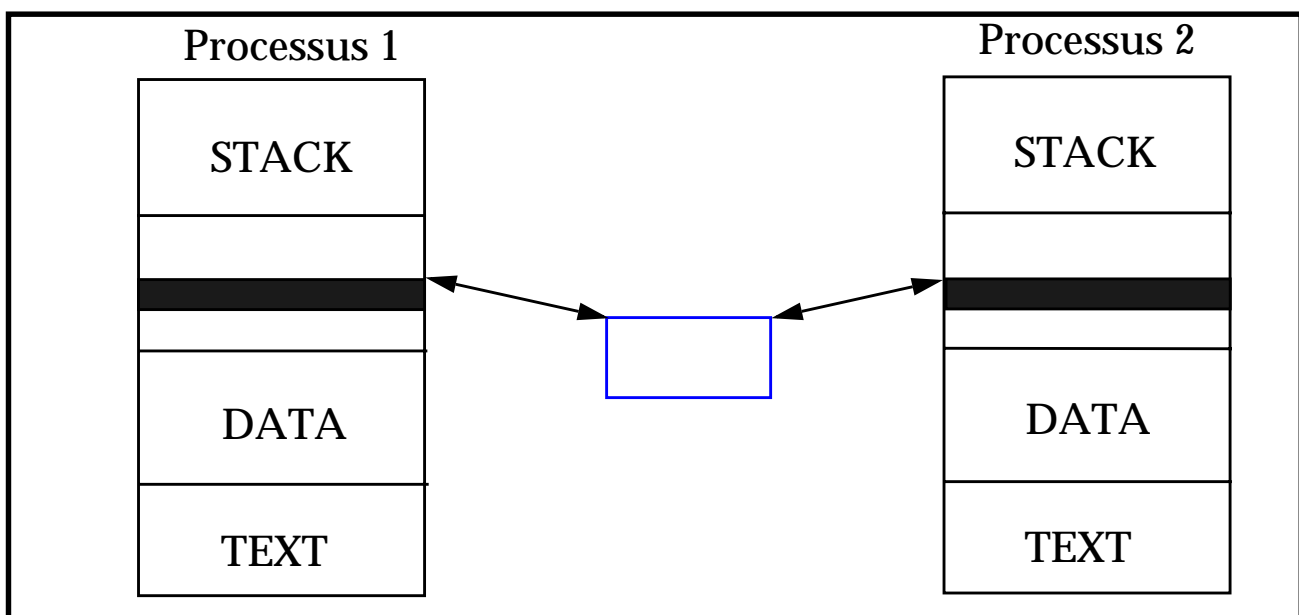
Applicatifs base de données

Mécanismes internes Unix

Inter Processus Communication

- Trois formes de mécanisme de communication inter-processus :
 - Mémoire partagée : communication possible entre processus connaissant la zone partagée,
 - Sémaphores : synchronisation dans l'utilisation des ressources partagées,
 - File de messages : envoi et réception sélective de messages entre processus.
- Mécanismes alloués lors d'une utilisation

Mémoire partagée



Applicatifs base de données

Mécanismes internes Unix

Inter Processus Communication

Définition

Les IPC (*Inter Processus Communication*) proposent un mécanisme de communication entre processus d'une même unité centrale. Ces mécanismes sont largement utilisés dans de nombreux serveurs et en particulier, les serveurs de bases de données.

UNIX propose trois formes différentes de mécanisme de communication inter-processus :

- *les messages* : les processus communiquent à l'aide d'enregistrement de données formatées,
- *la mémoire partagée* : les processus se partagent une partie de leur espace mémoire virtuel (même zone mémoire),
- *les sémaphores* : méthode de synchronisation des processus.

Mémoire partagée

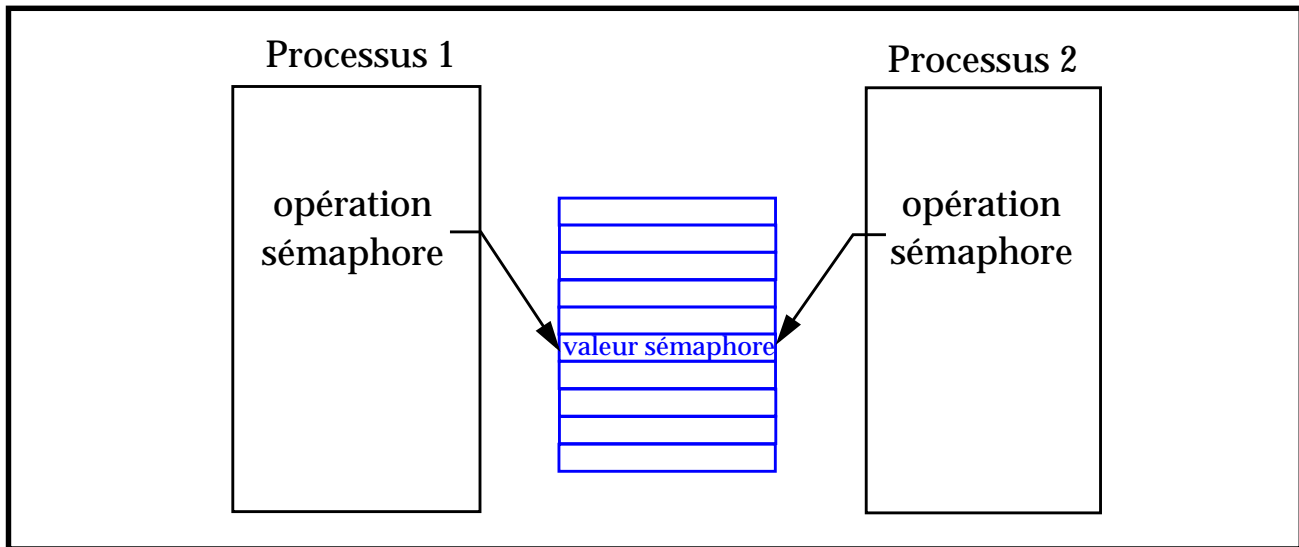
La zone mémoire partagée peut être mappée dans l'image de plusieurs processus qui communiquent via cette dernière. Elle n'est pas protégée contre l'accès concurrent. Il est du ressort du programmeur d'utiliser un sémaphore pour en protéger l'accès.



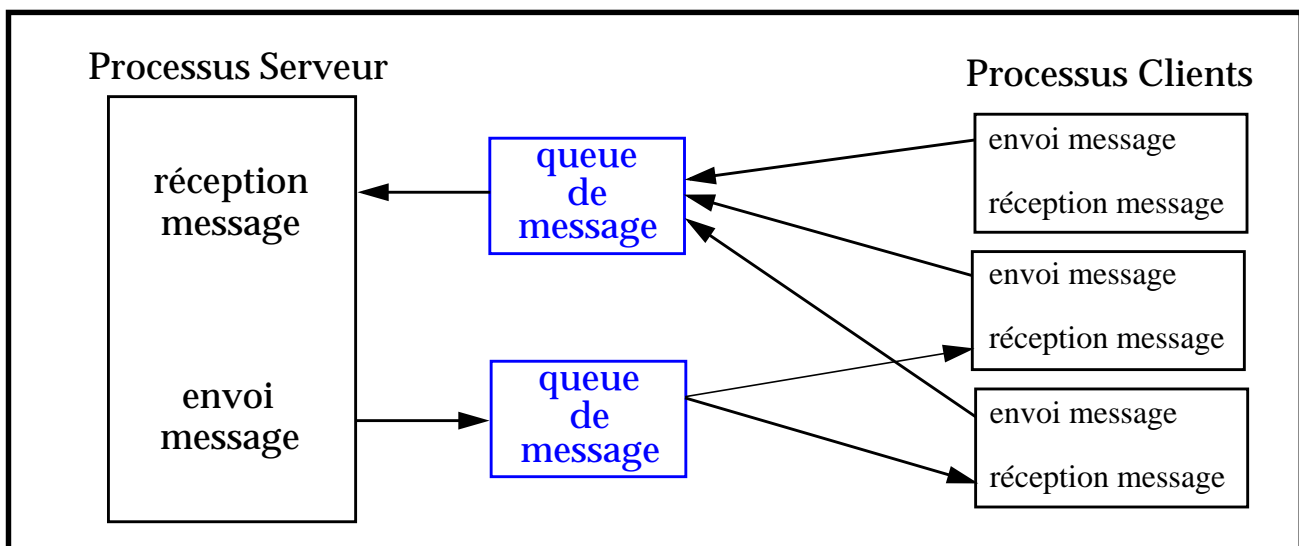
Applicatifs base de données

Mécanismes internes Unix

Sémaphores



File d'attente de messages



Applicatifs base de données

Mécanismes internes Unix

Sémaphores

Le groupe de sémaphores autorise tous types d'actions pour gérer une synchronisation, une protection contre un accès concurrent, une gestion de buffers de taille fixe.

File d'attente de messages

La file d'attente de messages propose un système de messagerie. Les processus postent des messages « typés » (ayant une marque de reconnaissance -une sorte d'adresse-). D'autres processus (ou les mêmes) vont recevoir ces messages. Ils peuvent choisir de recevoir tout « type » de messages ou de n'être sensible qu'à un certain « type » de message. L'utilisation de ces « type » permet à l'application de n'avoir à s'allouer qu'une file unique de messages et non pas plusieurs.

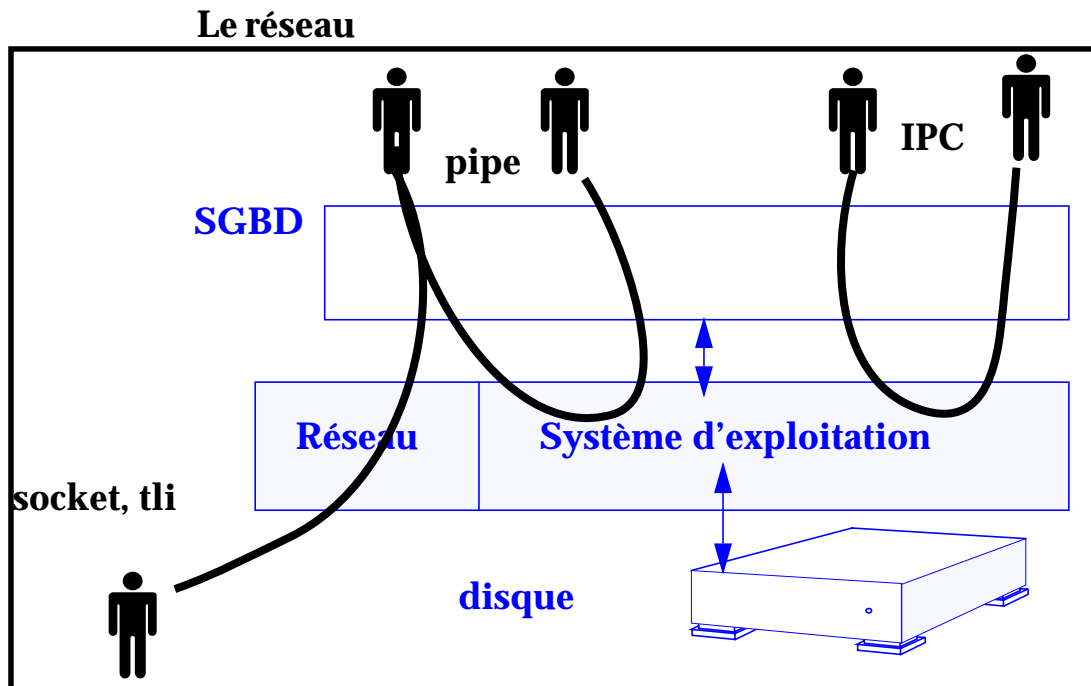
Le système propose un ensemble d'appels assurant la gestion de ces objets. Ces appels sont *généraux* et n'implémentent pas d'algorithme particulier. Il est du ressort du programmeur de les utiliser comme le nécessite l'application.

Interfaces de programmation des IPC

Les principales opérations possibles sont *xxxget* (créer un objet ou obtenir son identificateur), *xxxop* (travailler sur l'outil) et *xxxctl* (contrôler ou commander une fonctionnalité de l'outil). Il existe d'autres appels typiques de certains outils comme lors de l'utilisation de la file de messages. Pour gérer correctement ces ressources, l'opération *xxxop* est « *indivisible* ». Lors de son exécution, soit le processus l'utilisant la déroule sans être interrompu (protection contre le time-sharing), soit il se trouve bloqué si l'action est impossible.

Applicatifs base de données

Mécanismes internes Unix



```

tadoussac -> vancouver    TELNET C port=32805 1
vancouver -> tadoussac     TELNET R port=32805 1
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805 s
vancouver -> tadoussac     TELNET R port=32805 s
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805
vancouver -> tadoussac     TELNET R port=32805
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805
vancouver -> tadoussac     TELNET R port=32805
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805
vancouver -> tadoussac     TELNET R port=32805
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805
vancouver -> tadoussac     TELNET R port=32805 -
tadoussac -> vancouver    TELNET C port=32805 -
vancouver -> tadoussac     TELNET R port=32805 -
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805 a
vancouver -> tadoussac     TELNET R port=32805 a
tadoussac -> vancouver    TELNET C port=32805
tadoussac -> vancouver    TELNET C port=32805 1
vancouver -> tadoussac     TELNET R port=32805 1

```

Applicatifs base de données

Mécanismes internes Unix

Le réseau

Chaque interrogation cliente passe par une connexion sur le serveur. Les protocoles d'interrogation liés aux bases de données sont basés sur TCP. Il est donc nécessaire de surveiller ce type de ressource.

Il est indubitablement préférable d'effectuer une interrogation via le protocole spécifique au SGBD plutôt que par des protocoles non adaptés de type `telnet`, qui nécessitent une trame par caractère entré (sans compter les trames d'accusé de réception).



Applicatifs base de données

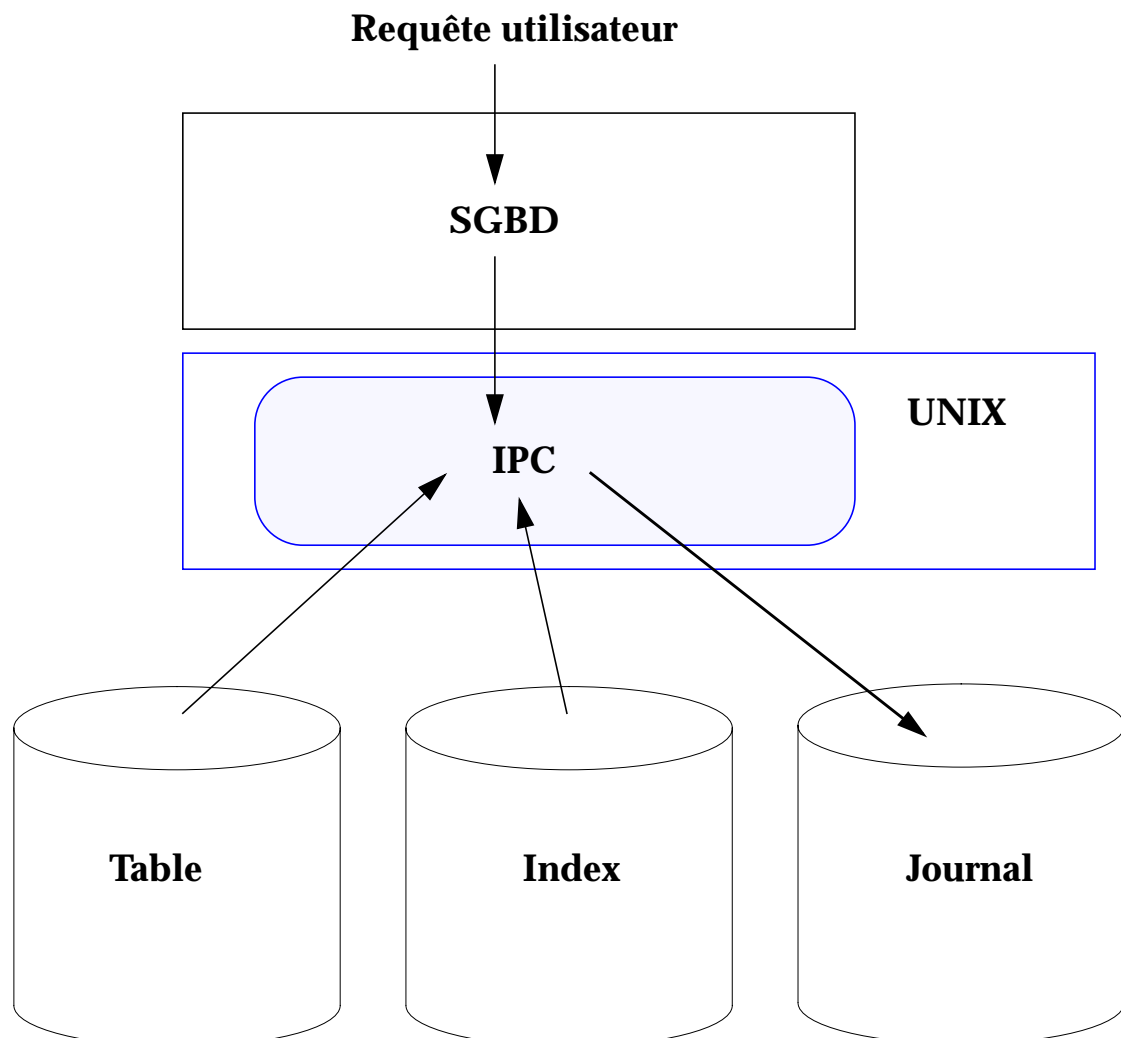
Mécanismes internes à la base de données

Transaction

Tables

Index

Journal



Applicatifs base de données

Mécanismes internes à la base de données

Transaction

La transaction est l'ordre de base géré par le SGBD, il correspond à un ordre de lecture, écriture ou modification demandé par un utilisateur. cette transaction met en oeuvre des accès à une ou plusieurs tables.

Tables

La table est l'objet de base du SGBD, il y stocke les informations qu'il gère. Il est possible de faire un parallèle entre la table pour le SGDB et le fichier pour Unix. Comme dans le cas du fichier, il est possible que cette table soit fragmentée. Il est du ressort de l'administrateur de la base de ré-organiser ces structures.

Index

Pour accéder plus rapidement à une information, il est possible d'adjoindre (et fortement conseillé !) à une table un (ou des) index. Ils proposent des méthodes rapides d'accès aux informations. Ces derniers sont accédés en parallèle avec les tables.

Journal

Le SGBD ne peut pas accéder de façon incessante aux disques. Ainsi, il stocke le plus d'informations possibles dans la RAM du système d'exploitation. Si le système vient à s'arrêter les informations sont alors perdues. Pour ne pas induire de problème d'intégrité, le SGBD gère un journal des transactions où ces dernières sont stockées avant d'être validées sur le disque. Ce journal est accédé de façon permanente.



Gestion du réseau

Les bandes passantes

Les types de transports

Les charges des divers applicatifs

Gestion du réseau

Les bandes passantes

Comme dans le cas des supports magnétiques, il est important de connaître les capacités de transfert de ce matériel.

Les types de transports

L'administrateur doit superviser deux types de transports, un transport connecté et un transport non connecté. Chacun dispose de ses caractéristiques et possède ses propres domaines d'utilisations et ses propres limites.

Les charges des divers applicatifs

Chaque applicatif nécessite une utilisation particulière du support physique.



Gestion du réseau

Les bandes passantes

Type de contrôleurs

- Ethernet
 - Lance Ethernet SBus controller
 - Quad Lance Ethernet SBus controller

- Fast Ethernet
 - Quad Fast Ethernet SBus controller
 - SunFastEthernet2.0 SBus controller

- FDDI
 - FDDI/S SAS Fiber SBus controller (simple)
 - FDDI/S DAS Fiber SBus controller (double)

Gestion du réseau

Les bandes passantes

Type de contrôleurs

■ Ethernet

Chaque machine dispose de une ou plusieurs interfaces Ethernet. La vitesse de l'information sur le support est de 10 Mbits/s. Par le protocole CSMA/CD et par les protocoles applicatifs, nous perdons environ 40 % de la bande passante théorique.

■ Fast Ethernet

Les matériels les plus récents disposent de contrôleurs Ethernet travaillant soit à 10Mbits/s soit à 100Mbits/s. la règle précédente s'applique encore pour la bande passante.

■ FDDI

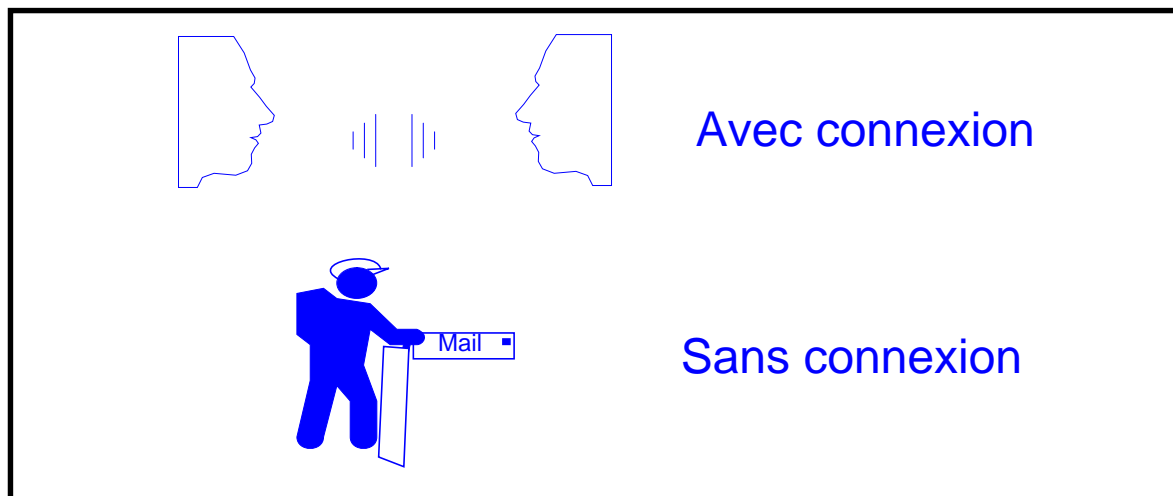
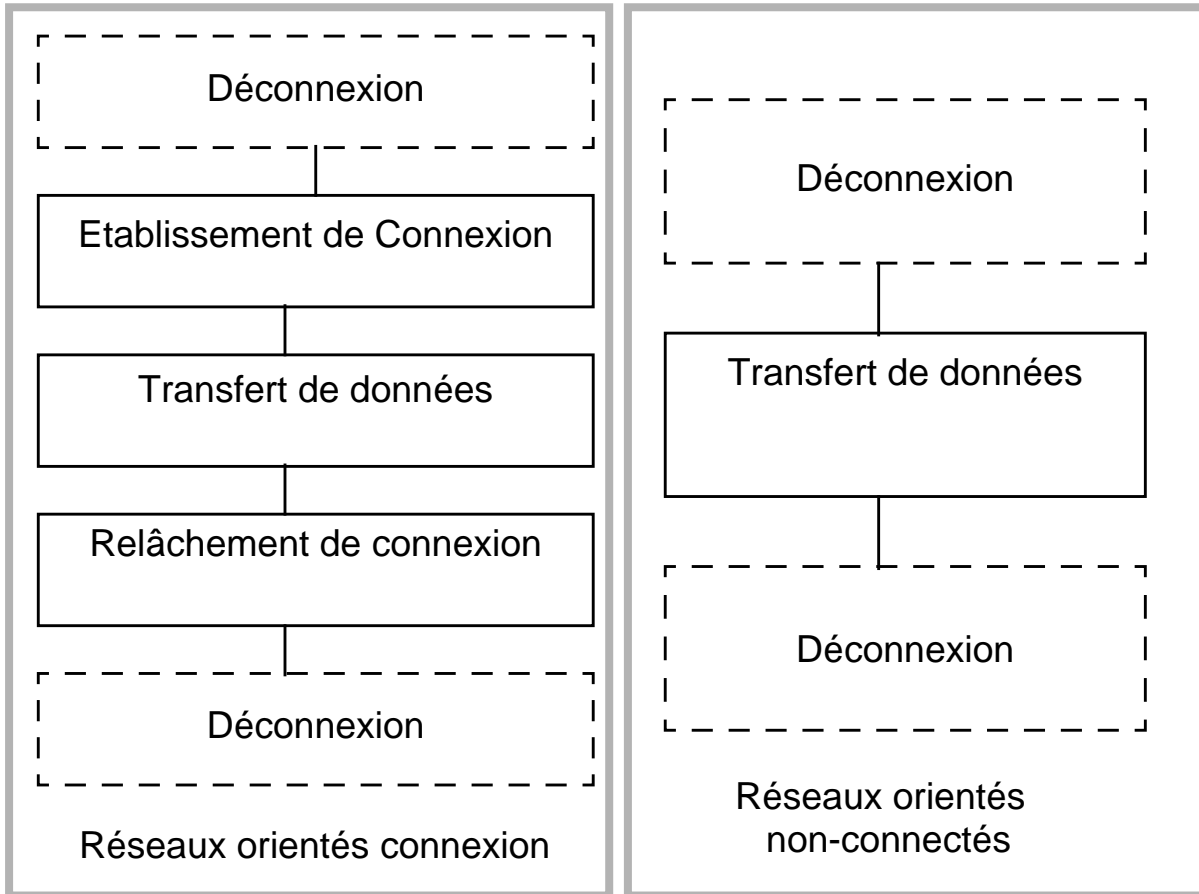
L'interface FDDI n'est pas présente en natif sur les machines. Elle propose une topologie double anneau et une vitesse de l'information sur le support de 100Mbits/s. Le protocole liaison est de type jeton.

Type de réseau	Vitesse	Bande Passante
Ethernet	10 Mbits/s	600 ko/s
Fast-Ethernet	100 Mbits/s	6 Mo/s
FDDI	100 Mbits/s	10 Mo/s



Gestion du réseau

Les types de transports



Gestion du réseau

Les types de transports

Nous disposons d'un transport connecté et d'un transport non connecté.

■ Cas de UDP

Ce transport est non connecté. Les échanges ont lieu via des paquets décorrélés entre-eux, aucun accusé de réception n'est géré par la couche transport. L'application place sous son contrôle ce type de mécanisme en programmant des timers et en utilisant des mécanismes de relance. Il est nécessaire pour les envois en multi-cast.

Une trame UDP est encapsulée dans une trame Ethernet (1024 octets au maximum).

■ Cas de TCP

Ce protocole est connecté. Il assure l'ordonnancement des paquets, et la bonne distribution de ces derniers.

Il est le protocole de base utilisé par les principales applications clients/serveurs (interrogation de base de données, http, ftp, etc.).

Ce protocole est plus contraignant en terme d'utilisation du réseau et de la machine, ses paramètres sont nombreux.



Gestion du réseau

TCP

Nombre de connexions

Taille des transferts

Temps de déconnexion

```
vancouver (root-sh) # ndd /dev/tcp
name to get/set ? ?
? (read only)
tcp_close_wait_interval (read and write)
tcp_conn_req_max_q (read and write)
tcp_conn_req_max_q0 (read and write)
tcp_conn_req_min (read and write)
tcp_conn_grace_period (read and write)
tcp_cwnd_max (read and write)
tcp_debug (read and write)
tcp_smallest_nonpriv_port (read and write)
tcp_ip_abort_cinterval (read and write)
tcp_ip_abort_linterval (read and write)
tcp_ip_abort_interval (read and write)
tcp_ip_notify_cinterval (read and write)
tcp_ip_notify_interval (read and write)
tcp_ip_ttl (read and write)
tcp_keepalive_interval (read and write)
tcp_deferred_ack_interval (read and write)
tcp_snd_lowat_fraction (read and write)
tcp_sth_rcv_hiwat (read and write)
tcp_sth_rcv_lowat (read and write)
tcp_dupack_fast_retransmit (read and write)
tcp_ignore_path_mtu (read and write)
tcp_rcv_push_wait (read and write)
tcp_xmit_hiwat (read and write)
tcp_xmit_lowat (read and write)
tcp_rcv_hiwat (read and write)
tcp_rcv_hiwat_minmss (read and write)
tcp_fin_wait_2_flush_interval (read and write)
tcp_co_min (read and write)
tcp_max_buf (read and write)
tcp_zero_win_probesize (read and write)
tcp_strong_iss (read and write)
tcp_slow_start_initial (read and write)
tcp_extra_priv_ports (read only)
....
```

Gestion du réseau

TCP

Nombre de connexions

Chaque connexion TCP est coûteuse à gérer et à entretenir. Il est donc nécessaire de surveiller chaque connexion.

Lors d'un transfert TCP, l'algorithme d'échange est le suivant :

- demander une connexion
- attendre que le serveur nous accorde cette connexion
- accuser réception de la connexion
- transférer des informations
- se déconnecter.

Chaque partie de l'algorithme est programmable, tant pour obtenir de meilleurs performances que pour obtenir une qualité de service correcte au niveau d'un serveur.



Gestion du réseau

TCP

Nombre de connexions

Taille des transferts

Temps de déconnexion

```
tadoussac# netstat -s
```

UDP

```
udpInDatagrams      = 410      udpInErrors         = 0
udpOutDatagrams     = 405
```

TCP

```
tcpRtoAlgorithm     = 4        tcpRtoMin           = 200
tcpRtoMax           = 60000   tcpMaxConn          = -1
tcpActiveOpens     = 50      tcpPassiveOpens   = 34
tcpAttemptFails     = 1        tcpEstabResets      = 0
tcpCurrEstab        = 4        tcpOutSegs          = 80382
tcpOutDataSegs      = 36717   tcpOutDataBytes   = 27837144
tcpRetransSegs      = 75      tcpRetransBytes   = 134
tcpOutAck            = 43664   tcpOutAckDelayed    = 6153
tcpOutUrg           = 0        tcpOutWinUpdate     = 0
tcpOutWinProbe      = 22      tcpOutControl        = 166
tcpOutRsts          = 1        tcpOutFastRetrans   = 0
tcpInSegs            =115279
tcpInAckSegs        = 18708   tcpInAckBytes        =27837223
tcpInDupAck          = 125     tcpInAckUnsent       = 0
tcpInInorderSegs    =107442   tcpInInorderBytes    =133438107
tcpInUnorderSegs    = 0        tcpInUnorderBytes = 0
tcpInDupSegs        = 0        tcpInDupBytes        = 0
tcpInPartDupSegs    = 0        tcpInPartDupBytes    = 0
tcpInPastWinSegs    = 0        tcpInPastWinBytes    = 0
tcpInWinProbe        = 0        tcpInWinUpdate       = 22
tcpInClosed          = 0        tcpRttNoUpdate       = 1
tcpRttUpdate        = 5794   tcpTimRetrans        = 232
tcpTimRetransDrop    = 0        tcpTimKeepalive      = 7
tcpTimKeepaliveProbe= 0        tcpTimKeepaliveDrop  = 0
tcpListenDrop       = 0
```

Gestion du réseau

TCP

Nombre de connexions

Le nombre de connexions parties d'un serveur (il se comporte comme un client) est donné le champ `tcpActiveOpens`, le nombre de connexions parvenues à un serveur (il se comporte comme un serveur) est donné par la champ `tcpPassiveOpens`.

Une connexion arrivant, elle est mise en file d'attente pour être traitée. Le nombre de connexions rejetées (car la file d'attente était pleine) est fournie par le champ `tcpListenDrop`.

Les connexions correspondant à des SYN ATTACK sont repérées par les champs `tcpHalfOpenDrop` et `tcpListenDropQ0`. Ces champs sont disponibles en natif sur la 2.6 et sur la 2.5.1 avec les patch 103582-12, et 103630.

La file d'attente des connexions est fournie par le paramètre `tcp_conn_req_max_q` (commande `ndd`).

Taille des transferts

La commande `netstat -s` permet de visualiser le nombre de retransmissions TCP et el nombre de réordonnancements ayant dû avoir lieu.

Il est possible de change la taille des buffers de transfert associés à TCP, via les variables `tcp_xmit_hiwat` et `tcp_recv_hiwat`.

Temps de déconnexion

Lors de la demande de déconnexion par le client, le serveur entretient encore la connexion pendant un temps donné par la variable `tcp_close_wait_interval`.



Gestion du réseau

Les charges des divers applicatifs

Utilisation du réseau

Temps de transfert

Type de trame

Utilisation de la machine

Concurrence

Gestion du réseau

Les charges des divers applicatifs

Utilisation du réseau

Le premier paramètre à prendre en compte est le type de transport utilisé pour une application. Le second paramètre est le temps de transfert utilisé. la charge induite par des serveurs de sauvegarde, de jumpstart ou d'impression n'est pas comparable à celle d'un serveur NFS.

Il est aussi nécessaire de prendre en compte la taille des trames qui seront échangées.

Utilisation de la machine

Le réseau n'est pas la seule ressource à surveiller. Si l'application dispose d'une notion de concurrence (processus ou thread), il est important de surveiller les ressources mémoires liées au serveur.



Application NFS

Le protocole

Cas du client

Cas du serveur

Application NFS

Le protocole

NFS est fourni selon deux protocoles différents depuis Solaris 2.5. L'administrateur doit pouvoir choisir le protocole à utiliser en fonction de son environnement.

Cas du client

Le client peut changer les paramètres du protocole sur son ordre de montage.

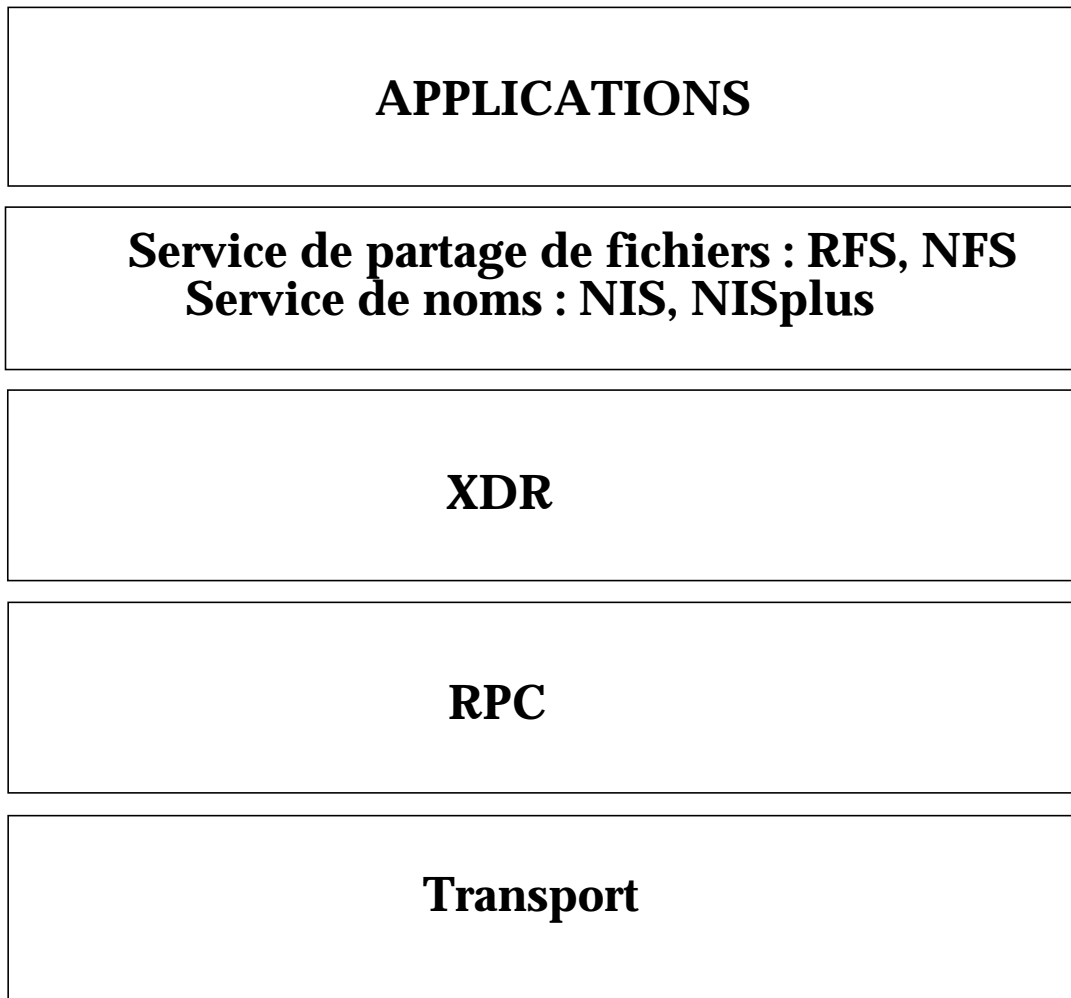
Cas du serveur

Il peut être nécessaire de modifier certains paramètres du serveur pour s'adapter aux besoins des clients.



Application NFS

Le protocole



Service de temps extension RPC
Service de sécurité extension RPC

Parallélisme possible avec RPC

Application NFS

Le protocole

NFS permet de partager un espace disque entre plusieurs machines. Il est proposé en deux versions : la version 2 et la version 3 (disponible depuis Solaris 2.5).

NFS version 2

Ce protocole était, au départ, proposé uniquement sur UDP. Le client effectue une requête lui permettant de récupérer les informations qu'il a besoin pour créer son arborescence (en fonction de son système d'exploitation), puis il peut disposer des objets mis à sa disposition.

NFS version 3

Cette version est une nouvelle implémentation du protocole. Elle permet de :

- travailler sur TCP ou UDP,
- augmenter les potentialités de cache des clients,
- travailler de façon asynchrone pour les écritures,
- fournir systématiquement l'inode d'un fichier, lors d'une modification de ce dernier.



Application NFS

Le protocole

```
tadoussac# snoop rpc nfs

vancouver (root-sh) # mount tadoussac:/b /b
vancouver -> tadoussac      NFS C NULL3
tadoussac -> vancouver      NFS R NULL3
vancouver -> tadoussac      NFS C FSINFO3 FH=00AA
tadoussac -> vancouver      NFS R FSINFO3 OK
vancouver -> tadoussac      NFS C GETATTR3 FH=00AA
tadoussac -> vancouver      NFS R GETATTR3 OK

vancouver (root-sh) # cd /b
vancouver -> tadoussac      NFS C ACCESS3 FH=00AA (lookup)
tadoussac -> vancouver      NFS R ACCESS3 OK (lookup)

vancouver -> tadoussac      NFS C ACCESS3 FH=00AA (read)
tadoussac -> vancouver      NFS R ACCESS3 OK (read)
vancouver -> tadoussac      NFS C REaddirPLUS3 FH=00AA Cookie=0 for 1048/8192
tadoussac -> vancouver      NFS R REaddirPLUS3 OK 10+ entries (incomplete)

vancouver (root-sh) # ls
vancouver -> tadoussac      NFS C GETATTR3 FH=00AA
tadoussac -> vancouver      NFS R GETATTR3 OK
vancouver -> tadoussac      NFS C GETATTR3 FH=00AA
tadoussac -> vancouver      NFS R GETATTR3 OK

vancouver (root-sh) # ls -al (1 fichier dans le repertoire)
vancouver -> tadoussac      NFS C ACCESS3 FH=3C6C (lookup)
tadoussac -> vancouver      NFS R ACCESS3 OK (lookup)

vancouver (root-sh) # cd LI*
vancouver -> tadoussac      NFS C ACCESS3 FH=3C6C (read)
tadoussac -> vancouver      NFS R ACCESS3 OK (read)

vancouver (root-sh) # ls -al
vancouver -> tadoussac      NFS C REaddirPLUS3 FH=3C6C Cookie=0 for 1048/8192
tadoussac -> vancouver      NFS R REaddirPLUS3 OK 3 entries (No more)

vancouver (root-sh) # cat fic_lic
vancouver -> tadoussac      NFS C GETATTR3 FH=57A8
tadoussac -> vancouver      NFS R GETATTR3 OK
vancouver -> tadoussac      NFS C ACCESS3 FH=57A8 (read)
tadoussac -> vancouver      NFS R ACCESS3 OK (read)
vancouver -> tadoussac      NFS C READ3 FH=57A8 at 0 for 4096
tadoussac -> vancouver      NFS R READ3 OK (3501 bytes) EOF
```

Application NFS

Le protocole

Le suivi via la commande snoop, nous permet d'avoir une première vision du protocole NFS.

Les machines vont échanger des trames de deux types :

- recherche d'informations de type inode : `readdir`, `access3`, `getattr3`,
- recherche d'informations contenues dans un fichier : `read`, `write`.

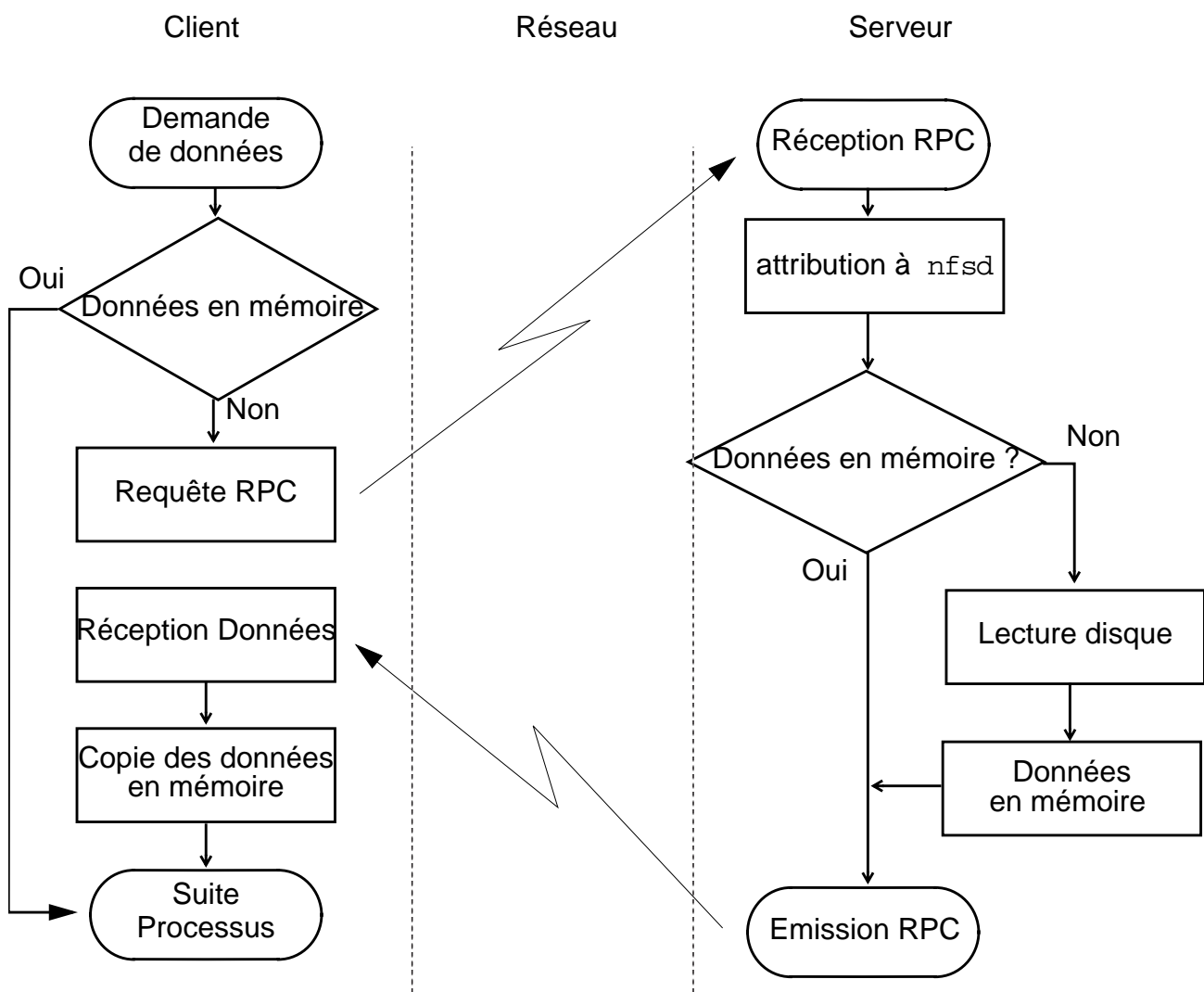
Un serveur sera dit « serveur d'attributs » si la plus grande partie des trames qu'il échange sont du premier type. Un serveur sera dit « serveur de data », si la plus grande partie des trames qu'il échange sont de second type.



Application NFS

Le protocole

Requête de lecture



Application NFS

Le protocole

Requête de lecture

Les étapes suivantes décrivent une transaction de lecture NFS :

- Le processus Client demande des données lors de son exécution. Cette requête est bloquante jusqu'à la terminaison de la transaction.
- Le Client vérifie si les données sont en mémoire ou cachées localement. Si les données sont disponibles, le processus les utilise et continue son exécution.
- Si les données n'ont pas été rapatriées localement en mémoire, le client peut les obtenir du serveur. La requête vers le serveur NFS se fait à travers des *Remote Procedure Call* (RPC).
- Le message RPC ainsi constitué est alors envoyé sur le réseau.
- Le message RPC est reçu par le serveur qui valide un processus `nfsd` pour servir la requête. Il convient de rappeler que le processus `nfsd` est un démon qui s'exécute sur le serveur et accepte les appels RPC des Clients NFS.
- Le serveur vérifie si les données sont déjà stockées en mémoire ; si les données sont disponibles, le démon `nfsd` encapsule les données dans un message RPC à destination du Client. Le processus Client reçoit le message, recopie les données en mémoire et le processus Client continue.
- Tant que les données seront en mémoire sur le Client une requête ultérieure pour ces données sera réalisée localement.
- Si les données ne sont pas disponibles en mémoire, le serveur initialise une opération de lecture sur disque. Les données sont alors copiées en mémoire puis encapsulées dans un message RPC comme précédemment.

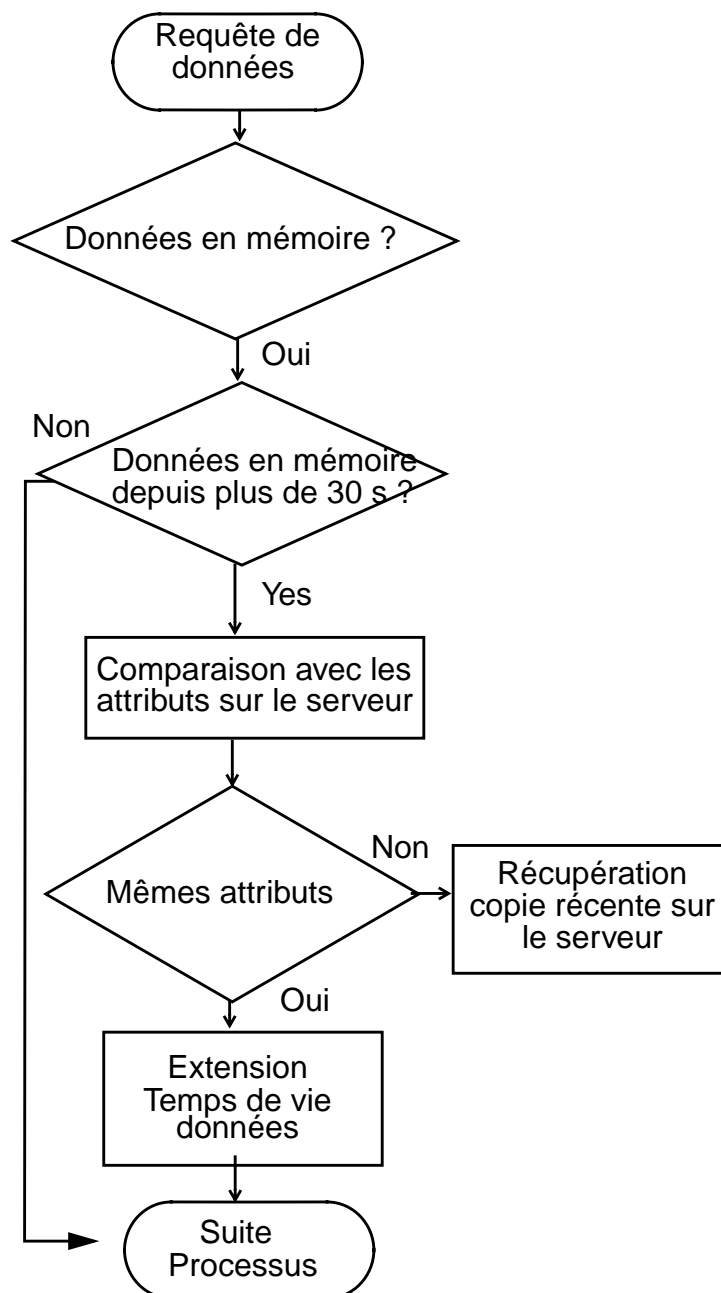


Application NFS

Le protocole

Vérification du contenu du cache

Client



Application NFS

Le protocole

Vérification du contenu du cache

Avant qu'un Client émette un message RPC de lecture NFS, il vérifie si les données sont déjà stockées en mémoire depuis une requête antérieure. Si les données sont disponibles, le processus les utilise et continue son exécution. Mais, il convient de vérifier si les données ont changé sur le serveur depuis la copie initiale depuis le serveur sur le Client.

Le Client vérifie la consistance du *cache* pour valider l'intégrité des données recopiées localement.

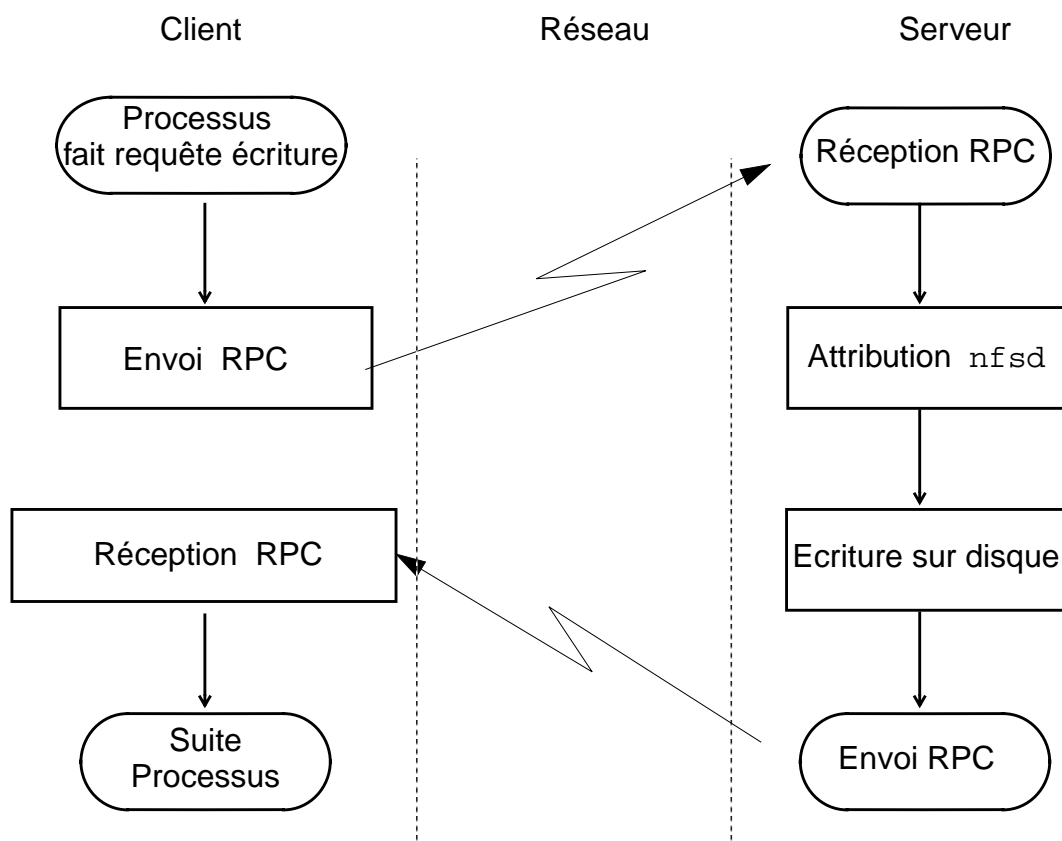
- Le Client trouve une copie des données sollicitées en mémoire. Il vérifie si la recopie locale a été faite depuis un temps supérieur à `actimeo` qui est une option spécifiée dans la commande `mount` (par défaut cette valeur vaut 30 secondes).
- Si la durée de vie des données dans le *cache* n'excède pas `actimeo`, le processus Client utilise les données et continue son exécution.
- Si les données sont stockées localement depuis un temps supérieur à `actimeo`, le Client génère une requête NFS `getattr` pour comparer le temps de dernière modification entre le *cache* et les données du serveur.
- Si les attributs du *cache* ne sont différents de ceux du serveur, le Client NFS étend la durée de vie des données de 30 secondes supplémentaires et le processus Client continue son exécution.
- Si les attributs des données cachées localement sont différents de celles du serveur de données, les données sont invalidées et le Client génère une requête de lecture NFS pour récupérer une copie récente.



Application NFS

Le protocole

Requête d'écriture



Application NFS

Le protocole

Requête d'écriture

NFS implémente les écritures asynchrones sur le serveur.

Les étapes suivantes décrivent une transaction d'écriture NFS :

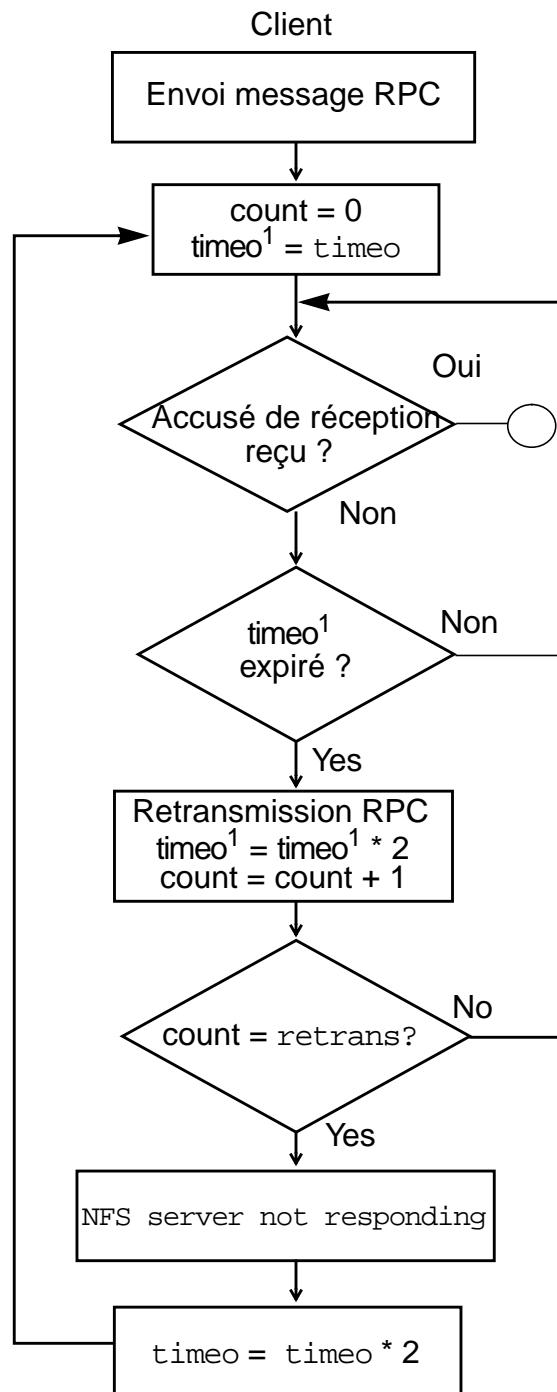
- Le processus Client génère une requête d'écriture durant son exécution. Le processus ne pourra continuer son exécution qu'à la complétude de la transaction en cours.
- Le Client encapsule les données dans un message RPC pour le serveur NFS.
- Le message RPC est ensuite envoyé à travers le réseau.
- Le serveur reçoit les messages RPC et attribut un démon `nfsd` pour le service de la requête.
- Le démon `nfsd` programme une opération sur disque pour lire les données.
- Le serveur envoie un message d'accusé de réception RPC au Client validant la complétude de la transaction d'écriture.
- Le Client reçoit le message et le processus continue son exécution.



Application NFS

Le protocole

Retransmission



Application NFS

Le protocole

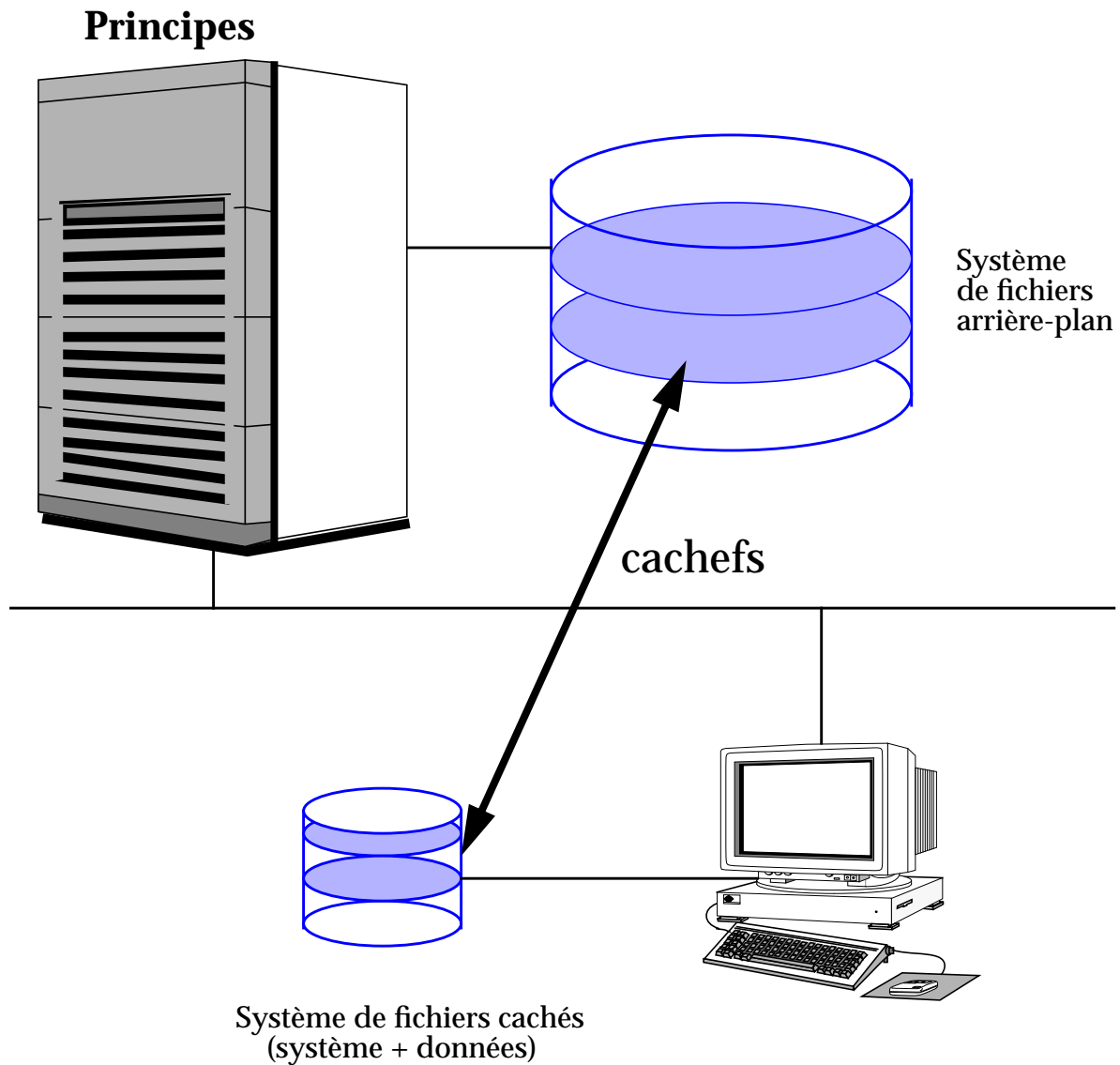
Retransmission

Les retransmissions NFS sont générées par un Client qui n'a pas reçu de réponse d'un serveur dans un temps donné.

- Après l'envoi d'un message au serveur, le Client attend une réponse dans une période de temps donné (`timeo` qui est un paramètre pouvant être positionné par l'option correspondante de la commande `mount`)
- Si aucun accusé de réception n'est reçu avant l'expiration de la période de *time-out* `timeo`, le Client renvoie la requête en doublant le temps de *time-out*, et incrémente le compteur de retransmission.
Le paramètre `timeo` initial est aussi appelé *minor time-out*.
- Si le nombre de retransmissions est égal à la valeur du paramètre `retrans` (qui peut aussi être spécifié par l'option correspondante de la commande `mount`), le serveur génère un message d'erreur "NFS server not responding".
- Le paramètre `retrans` est aussi appelé *major time-out*.
- Le Client NFS double à chaque retransmission la période d'attente `timeo`.
- Le Client continue ses retransmissions jusqu'à réception d'un accusé de réception.



Cachefs



Intérêt en matière d'applications réseau

- Cacher des systèmes de fichiers distants (NFS)

Cachefs

Principes

Un cache système de fichier utilise un disque local pour stocker temporairement des données fréquemment utilisées à partir d'un système distant, d'un CD-ROM...

Un cache de système de fichier peut utiliser tout ou une partie d'un disque pour stocker des données.

Un utilisateur n'a pas à savoir si les données sont sur le cache ou non.

Le système de fichier original est appelé *back file system* (système de fichiers en arrière plan) et les fichiers stockés dessus s'appellent des *back files* (fichiers en arrière plan).

Le cache d'un système de fichier réside sur le disque local et les fichiers qui s'y trouvent sont les fichiers cachés.

Le *directory cache* est un répertoire sur le disque local où se trouve le cache du système de fichier.

Objectifs

- Réduire le trafic réseau, sur des serveurs NFS,
- Améliorer les performances des systèmes de fichiers sur certains médias (CD-ROM...),
- Utilisation comme cache des arborescences système sur des machines avec peu d'espace disque (Solstice AutoClient).



Cachefs

Ressources

- Système de fichiers UFS pour le cache
- Système de fichiers en avant plan, dédié ou pas
- Ressources cachées pour les accès en lecture seulement.

Configuration en deux étapes

- 1/ `cfsadmin`
- 2/ `mount -F cachefs`

Cachefs

Ressources

Cachefs nécessite un système de fichiers UFS pour le cache.

Vous pouvez utiliser un système de fichier déjà existant comme cache ou vous pouvez en créer un nouveau. Dédier un système de fichier au *cachefs* vous donnera un plus grand contrôle sur l'espace alloué au cache.

Configuration

Il y a deux étapes pour configurer un cache de système de fichier :

1/ Vous devez créer un cache avec la commande `cfsadmin`.

2/ Vous devez monter le file système que vous désirez cacher en utilisant la commande `mount -F cachefs`.

Vous utilisez `cfsadmin` pour exécuter différentes tâches :

- Créer un cache,
- Créer et modifier les paramètres du cache,
- Afficher le informations du cache,
- Supprimer le cache.



Cachefs

Administration de cachefs

Principales commandes

- Créer un cache (paramètres par défaut) :
option `-c` de `cfsadmin`

```
# cfsadmin -c <répertoire_cache>  
# cfsadmin -c /local/mycache  
# mount -F cachefs...
```

- Supprimer un cache :
option `-d` de `cfsadmin`

```
# umount ...  
# cfsadmin -d <cache_id> <répertoire_cache>  
# cfsadmin -d /dev/dsk/c0t1d0s7 /local/mycache  
# fsck -F cachefs /local/mycache  
# cfsadmin -d all /local/mycache
```


Cachefs

Administration de cachefs

Principales commandes

- Créer un cache (paramètre par défaut)

L'exemple de la page précédente crée un cache ainsi qu'un *cache directory* `/local/mycache`, vous devez être sûr que le *cache directory* n'existe pas déjà. Cet exemple utilise les valeurs par défaut des paramètres du cache.

- Supprimer un cache

Avant de supprimer un cache d'un système de fichier, vous devez démonter tous les systèmes de fichiers cachés du *cache directory*.

L'identification du cache est une information obtenue par `cfsadmin -l`.

Après avoir supprimé un ou plusieurs systèmes de fichiers cachés, vous devez lancer la commande `fsck_cacheofs` pour corriger les comptes ressources du cache.

Vous pouvez supprimer tous les systèmes de fichiers dans un cache particulier en utilisant l'argument `all` avec l'option `-d` de `cfsadmin`.



Cachefs

Administration de cachefs

Paramètres du cache

■ Les paramètres du cache

Paramètres pour l'allocation de l'espace	Paramètres pour l'allocation des fichiers
maxblocks	maxfiles
minblocks	minfiles
threshblocks	threshfiles

■ Les valeurs par défaut des paramètres

Paramètres du cache	Valeur par défaut
maxblocks	90 %
minblocks	0 %
threshblocks	85 %
maxfiles	90 %
minfiles	0 %
threshfiles	85 %
maxfilesize	3 MB

Cachefs

Administration de cachefs

Paramètres du cache

Les valeurs par défaut sont pour un cache qui utilise tout le système de fichiers en avant plan. Pour limiter un cache pour seulement une portion du système de fichiers en avant plan, vous devez changer les paramètres.

■ Paramètres

maxblocks : pourcentage maximum de blocs, que le cachefs peut atteindre.

maxfiles : pourcentage maximum d'inodes que le cachefs peut atteindre.

■ Remarque

Ces paramètres ne garantissent pas que les ressources soient disponibles pour le *cachefs*.

Minblocks : ne garantit pas la disponibilité d'un minimum de ressource. Les paramètres *minblocks* et *threshblocks* travaillent ensemble. *Cachefs* doit attendre au minimum le pourcentage de blocs spécifié par *minblocks*, si le pourcentage de blocs disponibles du système de fichier en avant plan est supérieur à *threshblocks*.

Les paramètres *threshfiles* et *threshblocks* agissent sur le système de fichiers en avant plan, en entier, et non sur les systèmes de fichiers que vous avez cachés avec le système de fichiers en avant plan.

■ Remarque

Si vous utilisez la totalité du *front file system*, il devient inutile de changer les paramètres.



Cachefs

Options de montage de cachefs

- Montage d'un système de fichiers caché
 - mount
 - /etc/vfstab
 - autofs
- Exemples

```
# mount -F cachefs \  
-o backfstype=<type>,cachedir=<répertoire_cache>  
<back_filesystem> <mount_point>  
# mount -F cachefs \  
-o backfstype=nfs,cachedir=/local/cache1 \  
merlin:/docs /docs
```

Cachefs

Options de montage de cachefs

- Quelques options utilisés avec l'option "-o" :
 - **acdirmax = n, acdirmin = n, acregmax = n, acregmin = n** : utilisés pour la mise à jour périodique. Le défaut pour chaque paramètre est de 30 secondes.
 - **actimeo = n** : positionne tous les paramètres de vérification périodique à n secondes.
 - **backfstype** : spécifie le type de système de fichiers en arrière plan.
 - **backpath** : spécifie le point de montage du système de fichiers en arrière plan. A n'utiliser seulement si le système de fichiers est déjà monté.
 - **cachedir** : spécifie le nom du *cache directory*.
 - **cacheid** : vous permet d'assigner une chaîne de caractères pour identifier chaque système de fichiers caché. Si vous ne spécifiez pas de cacheid, cachefs en génère un. Vous en avez besoin lorsque vous désirez supprimer un système de fichiers caché.
 - **noconst** : invalide la mise à jour périodique. Utilisez noconst quand le système de fichiers en arrière plan et le cache du système de fichiers sont en read-only.
 - **rw/ro** : permet au système de fichiers caché d'être exclusivement lu, ou modifiable.



Application NFS

Le client

```
# nfsstat -m
/nfs from ita2:/nfs
  Flags:
vers=2,proto=udp,auth=unix,soft,intr,dynamic,acl,rsize=8192,wsiz=8192,retrans=
5
Lookups:  srtt=7 (17ms), dev=3 (15ms), cur=2 (40ms)
Reads:    srtt=14 (35ms), dev=3 (15ms), cur=3 (60ms)
Writes:   srtt=39 (97ms), dev=8 (40ms), cur=8 (160ms)
All:      srtt=12 (30ms), dev=7 (35ms), cur=5 (100ms)

#
```

Application NFS

Le client

Le client dispose de plusieurs techniques pour changer le comportement de NFS :

- modification des options de montage :
 - soft, hard
 - rsize, wsize,
 - proto, vers,
 - actimeo
- modification du montage :
 - montage dans le fichier `/etc/vfstab`
 - automount
- modification de comportement :
 - cachefs.



Application NFS

Le serveur

```

ita2 (sh) # nfsstat -s

Server rpc:
Connection oriented:
calls      badcalls  nullrecv  badlen    xdrCALL   dupchecks dupreqs
62314      0          0         0         0         33638     0
Connectionless:
calls      badcalls  nullrecv  badlen    xdrCALL   dupchecks dupreqs
0          0         0         0         0         0         0

Server nfs:
calls      badcalls
62314      0
Version 2: (0 calls)
null      getattr   setattr   root      lookup    readlink  read
0 0%     0 0%     0 0%     0 0%     0 0%     0 0%     0 0%
wrcache  write     create    remove    rename    link      symlink
0 0%     0 0%     0 0%     0 0%     0 0%     0 0%     0 0%
mkdir    rmdir     readdir   statfs
0 0%     0 0%     0 0%     0 0%
Version 3: (62268 calls)
null      getattr   setattr   lookup    access    readlink  read
1 0%     719 1%   13 0%    117 0%   205 0%   0 0%    27331 43%
write     create    mkdir     symlink   mknod    remove    rmdir
33536 53%  41 0%   21 0%    0 0%     0 0%     1 0%    0 0%
rename    link      readdir   readdir+  fsstat   fsinfo    pathconf
0 0%     0 0%    4 0%     26 0%   0 0%     1 0%    0 0%
commit
252 0%

Server nfs_acl:
Version 2: (0 calls)
null      getacl    setacl    getattr   access
0 0%     0 0%     0 0%     0 0%     0 0%
Version 3: (46 calls)
null      getacl    setacl
0 0%     46 100%  0 0%

```

Application NFS

Le serveur

Le serveur est avant tout un serveur d'espace disque, il est donc important qu'il ait de bonnes performances sur ses disques partagés.

Le travail d'optimisation portera ensuite sur le protocole TCP ou UDP et enfin sur NFS (nombre de `nfsd`, etc.).



Application HTTP

Protocole

Méthodes

- GET
- POST
- HEAD
- PUT

Application HTTP

Protocole

HTTP (Hypertext Transfer Protocol) est un protocole applicatif pour échanger des informations multimédia entre des sites.

Il permet des échanges d'objets (informations et méthodes) entre les machines.

La représentation des données est banalisée et s'adapte à tout type de site.

Le protocole est basé sur un échange de type requête/réponse. La requête venant du client est analysée par le serveur. Il y correspond une méthode qui sera effectuée sur le serveur et qui conditionnera le traitement de cette requête.

HTTP utilise traditionnellement le port 80 sur une connexion TCP.

Le standard HTTP définit un certain nombre de règles de communication entre un browser web et un serveur web. Celles-ci constituent les « méthodes » de communication HTTP.

■ GET

GET permet d'accéder à un URL spécifié.

■ POST

POST permet d'envoyer les données utilisateur vers un URL spécifié existant.

■ HEAD

HEAD permet d'accéder uniquement aux informations d'en-tête de l'URL spécifié.

■ PUT

Permet d'envoyer des données.



Vue rapide sur les problèmes de développement

Les choix des développeurs

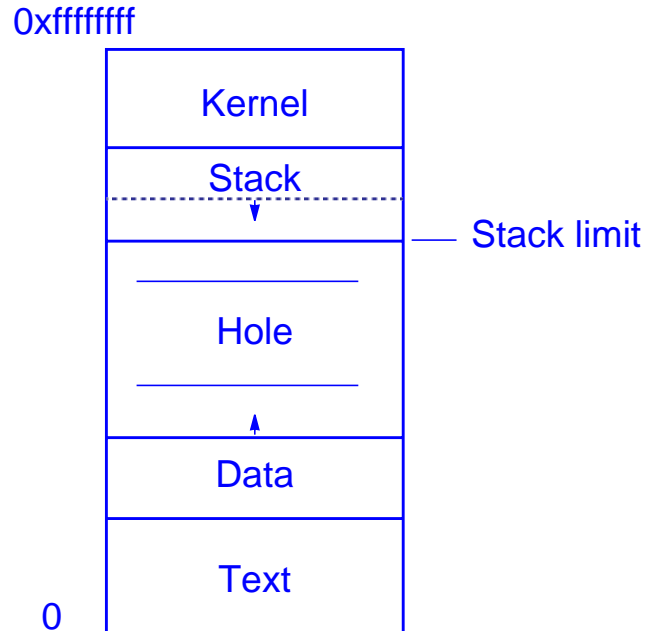
Choix de l'algorithme

Type d'optimisation

Choix des langages

Choix des outils de mise au point

Choix des appels



Vue rapide sur les problèmes de développement

Les choix des développeurs

Nous nous baserons sur le développement d'une application nouvelle où aucun choix technologique n'a encore eu lieu.

Choix de l'algorithme

La phase de design est aussi importante pour une application qu'elle l'est pour une base de données. Le plus souvent le développement est basé sur un générateur d'application (`rpcgen`, etc.) qui simplifie le codage d'une grande partie de l'application mais n'affranchit pas le développeur d'une phase d'optimisation.

Type d'optimisation

L'algorithme peut dépendre du type d'optimisation choisie (temps d'exécution, espace mémoire utilisé). Pour être réellement performant, ce choix doit prendre en compte les spécificités du matériel (zones caches présentes, etc.).

Choix des langages

Le langage peut influencer de façon significative sur le temps de réponse d'une application. Il est nécessaire de prendre en compte ce facteur pour obtenir les meilleures performances possibles d'une application (calcul, graphique, driver, etc.).

Choix des outils de mise au point

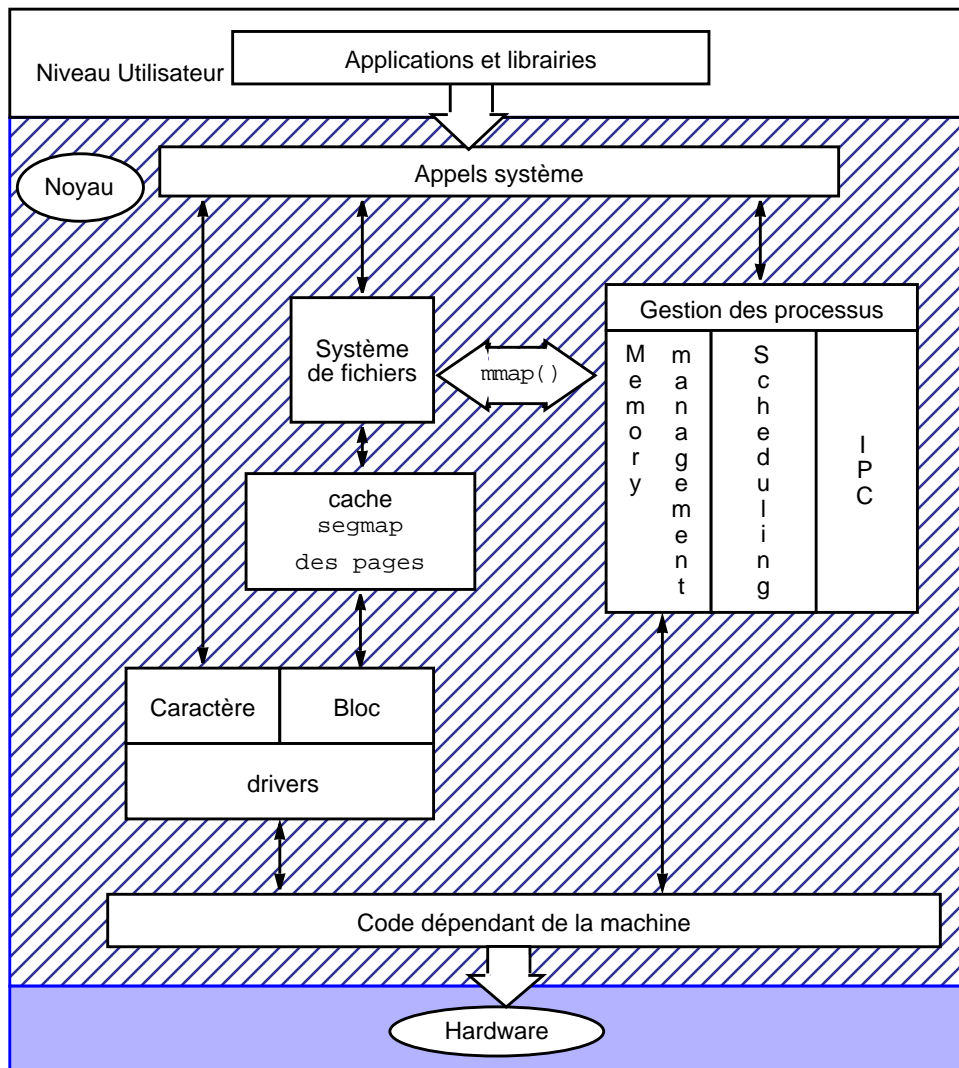
Il est maintenant possible de disposer de compilateurs proposant des phases d'optimisation très performantes (parallélisation, écriture automatique en multi-thread, etc.). Ce choix d'outils ne doit pas faire oublier la phase de mise au point et d'analyse (outils disponibles dans tout debugger) des zones mémoires utilisées (memory leak).



Vue rapide sur les problèmes de développement

Les choix des développeurs

Choix des appels



Vue rapide sur les problèmes de développement

Les choix des développeurs

Choix des appels

Une bonne connaissance des mécanismes internes du système d'exploitation est nécessaire pour obtenir les meilleures performances possibles des applications. Les questions qui doit se poser un développeur mettent en cause les mécanismes suivants :

- utilisation des processus/ des threads,
- gestion de la mémoire centrale (malloc, valloc, etc.),
- traitement des signaux,
- travail au niveau 2 ou 3,
- entrées/sorties synchrones, asynchrone, gestion de mmap,
- l'environnement d'exploitation....

Notes

Objectifs

Les sujets couverts par ce chapitre seront les suivants :

- présentation des outils de surveillance,
- les commandes Berkeley,
- les commandes SVR3,
- les outils freewares,
- les autres outils.



Présentation des outils de surveillance

Importance des outils

Les outils de base du système d'exploitation

Les outils tierce partie

Les outils récapitulatifs

L'accounting

Les outils de surveillance quotidienne

Les scripts de surveillance

Présentation des outils de surveillance

Importance des outils

- Les outils de base du système d'exploitation

Le système d'exploitation dispose d'un ensemble d'outils proposant de récupérer des informations sur les ressources utilisées par le système d'exploitation. Ces outils sont utiles pour suivre l'activité journalière d'un serveur. On peut leur reprocher leur manque de convivialité et leur sortie graphique des plus rudimentaires.

- Les outils tierce partie

Il existe des outils tierce-partie plus conviviaux, nous vous en présenterons un.

Les outils récapitulatifs

- L'accounting

Cet outil est fondamental pour obtenir un compte rendu moyen de l'activité des applications. Il est nécessaire de le valider pour disposer d'un bilan global d'activité.

Les outils de surveillance quotidienne

- Les scripts de surveillance

Les commandes Unix de base seront validées sous forme de script et intégrés dans la `crontab`.



Présentation des outils de surveillance

Outil	Disponibilité	Description
iostat	Solaris 2.x	Statistique sur les disques
netstat	Solaris 2.x	Statistique sur le réseau
nfsstat	Solaris 2.x	Statistique sur les applications RPC
vmstat	Solaris 2.x	Statistique sur la mémoire et le CPU
sar	Solaris 2.x	Statistique locale
snoop	Solaris 2.x	Activité réseau
nfswatch	freeware	Transactions NFS
top	freeware	Statistique sur la mémoire et le CPU
proctool	freeware	Statistique sur la mémoire et le CPU
SyMON	Solaris 2.5 (Ultra E)	Rapports basés sur du SNMP
SunNet Mgr	tierce partie	Rapports basés sur du SNMP
accounting	Solaris 2.6	Rapport global d'activité

Présentation des outils de surveillance

Un certain nombre de produits vont être décrits dans ce chapitre, l'administrateur utilisera celui qui lui permet d'obtenir les résultats les plus parlants pour surveiller l'activité de ses serveurs.



Présentation des outils de surveillance

Surveillance des applications

Surveillance de SunOS

Berkeley

SVR3

Surveillance du matériel

Présentation des outils de surveillance

Surveillance des applications

La surveillance peut être effectuée par l'accounting et par la commande `ps`.

Surveillance de SunOS

La surveillance peut être effectuée par deux types de commandes, soit les commandes issues de BSD :

```
. vmstat
```

```
. iostat
```

soit par les commandes issues de System V :

```
. sar
```

```
. sadc
```

Surveillance du matériel

La surveillance peut être effectuée par:

```
. iostat, sar
```

```
. netstat
```

```
. nfsstat
```



Intervalle de surveillance

Ressource	Temps de base	Echantillonnage
CPU, mémoire	nanoseconde	5 secondes
Réseau	microseconde	10 secondes
Disque	milliseconde	30 secondes

Intervalle de surveillance

Il est nécessaire d'adapter l'intervalle de surveillance aux ressources surveillées. Ainsi nous procéderons en fonction du tableau présent page de gauche.



Les commandes Berkeley

- **vmstat**
- **mpstat**
- **ps**
- **swap**
- **netstat**
- **iostat**
- **fstyp**
- **tunefs**
- **netstat**
- **nfsstat**

Les commandes Berkeley

vmstat

`vmstat` est l'une des premières commandes à lancer pour voir ce que votre système est en train de faire. Elle vous donne de nombreuses informations et vous pouvez en déduire l'activité de votre machine en pagination. L'activité disque est également surveillée ainsi que le nombre de jobs en activité ou en attente. La meilleure manière d'utiliser cette commande est de la lancer avec une limite de temps d'analyse de 5 secondes.

mpstat

Cette commande permet de visualiser la charge de chaque processeur.

ps

Cette commande, possédant de nombreuses options, donne une vue raisonnablement synthétique des processus en cours dans le système.

netstat

La commande `netstat` permet de connaître l'état des tables système concernant le réseau.

iostat

`iostat` donne des résultats statistiques sur les transferts réalisés avec des périphériques comme les terminaux et les disques.

fstyp

`fstyp` donne des informations détaillées sur le système de fichiers installé sur une partition.



Les commandes Berkeley

- **vmstat**
- **mpstat**
- **ps**
- **swap**
- **netstat**
- **iostat**
- **fstyp**
- **tunefs**
- **netstat**
- **nfsstat**

Les commandes Berkeley

nfsstat

Visualise des statistiques sur NFS et RPC.

tunefs

Permet de changer les paramètres d'un système de fichiers.

netstat

Visualise les ressources réseau.



Les commandes Berkeley

■ vmstat

```
# vmstat 1
procs      memory          page          disk          faults        cpu
r  b  w    swap  free  re  mf  pi  po  fr  de  sr  s3  s6    in   sy   cs  us  sy  id
1  0  5     516    0   0   1  0  0  0  0  0  0  0    9   60   48  4   2  94
0  0  3   75980    0   0  13 184 4 236 0 76  1  0  187 245 127 6 14 80
1  0  4   75984    0   0   0 96  0 40  0 60  0  0  148 152 100 7  5 88
0  0  2   75984    0   0   0 120 0 176 0 53  0  0  131  84  86 6  5 89
.
.
.
```

Les commandes Berkeley

■ vmstat

Cette commande permet de visualiser les activités liées au noyau et au CPU. Les goulets d'étranglement venant du noyau sont rapidement détectés par cette commande.

Principales options

- i liste le nombre d'interruptions par périphérique (iostat).
- s affiche les différents évènements système, depuis le boot ainsi que leur nombre.
- S affiche l'activité du swap, plutôt que celle de la pagination.



Les commandes Berkeley

■ vmstat

```
# vmstat 1
procs      memory
r b w   swap free re  mf pi po fr de sr s3 s6   in  sy  cs us sy id
1 0 5     516   0  0  1  0  0  0  0  0  0  0    9  60  48  4  2 94
0 0 3  75980   0  0 13 184 4 236 0 76  1  0  187 245 127 6 14 80
1 0 4  75984   0  0  0 96  0 40  0 60  0  0  148 152 100 7  5 88
0 0 2  75984   0  0  0 120 0 176 0 53  0  0  131  84  86 6  5 89
.
.
.
```


Les commandes Berkeley

■ vmstat

Principales colonnes

procs	nombre de processus dans la <i>runing queue</i> , dans chacun des 3 états
r	en queue d'exécution
b	bloqués pour des ressources
w	swappés
memory	swap libre en ko (<i>swap</i>) et RAM libre (<i>free</i>)
page	défauts de pages et activité de pagination
re	<i>page reclaim</i> , nombre de pages demandées par seconde
mf	<i>minor fault</i>
pi	<i>page in</i> (en ko)
po	<i>page out</i> (en ko)
fr	<i>free list</i> , nombre de pages libérées
de	nombre de pages anticipées
sr	nombre de pages scrutées par <i>pageout</i>
disk	nombre d'opérations disque par seconde.
faults	le nombre d'interruptions logicielles et matérielles par seconde
in	interruptions hors horloge système
sy	nombre d'appels système par seconde
cs	commutations de contexte.
cpu	pourcentage d'activité du temps CPU.
us	temps <i>user</i>
sys	temps <i>system</i>
id	temps d'inactivité
si	<i>swap in</i>
so	<i>swap out</i>



Les commandes Berkeley

■ vmstat -s

```
vancouver (root-sh) # vmstat -s
    0 swap ins
    0 swap outs
    0 pages swapped in
    0 pages swapped out
64789 total address trans. faults taken
    7115 page ins
    306 page outs
11905 pages paged in
    5501 pages paged out
    889 total reclaims
    889 reclaims from free list
    0 micro (hat) faults
64789 minor (as) faults
    6769 major faults
16803 copy-on-write faults
12403 zero fill page faults
15101 pages examined by the clock daemon
    2 revolutions of the clock hand
    8940 pages freed by the clock daemon
    673 forks
    27 vforks
    647 execs
1108023 cpu context switches
5221018 device interrupts
    95556 traps
1491205 system calls
    73083 total name lookups (cache hits 91%)
    38 toolong
    8263 user   cpu
    10731 system cpu
4847460 idle   cpu
    6616 wait   cpu
vancouver (root-sh) #
```

Les commandes Berkeley

■ **vmstat -s**

Cette commande propose un bilan des activités CPU de la machine, nous y trouvons le taux d'occupation des DNLC et le traitement des fichiers dont les noms sont trop longs pour y être mémorisés.



Les commandes Berkeley

■ mpstat

```
cmdtool - /sbin/sh
soleilo # mpstat
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
 0 42 2 0 113 6 36 1 1 2 0 97 1 5 29 65
 1 15 1 0 54 53 39 0 1 2 0 72 1 3 29 67
 2 17 1 0 162 161 25 0 1 3 0 62 1 2 29 68
 3 28 5 0 1 0 60 1 1 2 0 146 1 6 29 64
soleilo #
```

■ psrinfo

```
vancouver (root-sh) # psrinfo -v
Status of processor 0 as of: 04/20/98 16:05:18
Processor has been on-line since 04/19/98 14:20:34.
The sparc processor operates at 110 MHz,
and has a sparc floating point processor.
```

Les commandes Berkeley

■ mpstat

Cette commande indique des statistiques par CPU actifs sur le serveur.

CPU	identificateur du CPU
minf	minor faults
mjf	major faults
xcal	inter-processor cross-calls
intr	interruptions
ithr	interruption sans compter les interruptions dûes à l'horloge
csw	commutation de contexte
icsw	commutation de contexte involontaire
migr	migration de thread
smtx	attente multiple sur MUTEX
srw	attente simple sur MUTEX
syscl	appels système
usr	pourcentage de temps passé en USER
sys	pourcentage de temps passé en SYS
wt	pourcentage de temps passé à attendre
idl	pourcentage d'inactivité

■ psrinfo

Cette commande nous permet d'obtenir des informations sur les processeurs. Il est aussi possible de gérer ces processus via la commande `psradm` et `psrset`.



Les commandes Berkeley

■ ps

```
# ps -el
F S  UID  PID  PPID  C PRI NI  ADDR  SZ  WCHAN TTY  TIME CMD
8 R  7198 22028 19551 80  1 30 ff7c0000 349  ?  83:53 xlock
19 S  0 3 0 80 0 SY ff19d000 0 f00c26ae ? 265:00 fsflush
8 O  0 26070 26053 14 1 20 ff78c000 142 pts/4 0:00 ps
.
.
.
```

Les commandes Berkeley

■ ps

Cette commande donne la liste des processus actifs sur le système.

Principales options

-e	informations sur tous les processus présents
-l	liste longue
-a	informations sur tous les processus attachés à un terminal, sauf démons et background
-f	affiche toutes les colonnes (dont PPID)
-c	affiche la classe de scheduling du processus

Principales colonnes

S	Etat du processus
F	Etat du processus en hexadécimal
SZ (size)	taille de l'image du processus en mémoire (en pages)
CLS	classe de scheduling (SYS, RT, TS)
PRI	priorité du processus (relative par rapport à sa classe)
STIME	heure de lancement du processus
WCHAN	Adresse où le processus attend
NI	Incrément modifiant la priorité du processus (NIce)
TTY	le terminal de contrôle
TIME	le temps CPU pris par le processus
ADDR	L'adresse mémoire du processus



Les commandes Berkeley

■ swap

```
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t3d0s1 32,25      8 187912 127088
#
```

```
# mkfile 10m /exp/swap
#
```

```
# swap -a /exp/swap
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t3d0s1 32,25      8 187912 126920
/exp/swap         -          8  20472  20472
#
```

Les commandes Berkeley

■ swap

La commande `swap` permet de visualiser la quantité de swap disponible pour la machine. Avec l'option `-a`, l'administrateur peut ajouter des zones de swap, avec l'option `-d`, il peut en supprimer.



Les commandes Berkeley

■ iostat

```
# iostat -D 5
          sd0          sd1          sd2          sd3
rps wps util  rps wps util  rps wps util  rps wps util
  0  0  0.0    0  0  0.2    0  0  0.2    19  0 56.5
  0  1  2.6    0  0  0.0    0  0  0.0    0  17 99.2
  4  0  8.0    0  0  0.0    0  0  0.0    14  0 89.3
  0  2  2.3    0  0  0.0    0  0  0.0    7  17 78.0
.
.
.
```

```
# iostat -xnP
extended device statistics
r/s  w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
0.2  0.1   0.8    0.9  0.0  0.0   19.7   24.7   0   0 c0t3d0s0
0.0  0.0   0.0    0.3  0.0  0.0   24.2  102.7   0   0 c0t3d0s1
0.0  0.0   0.0    0.0  0.0  0.0    0.0    0.0   0   0 c0t3d0s2
0.0  0.0   0.0    0.0  0.0  0.0   14.4   22.4   0   0 c0t3d0s7
0.0  0.0   0.0    0.0  0.0  0.0    0.0    0.0   0   0 pancho:vold(pid251)
0.1  0.0   0.7    0.0  0.0  0.0    0.0  305.8   0   4 leghorn:/opt
```

Les commandes Berkeley

■ iostat

Principales options

-c	pourcentage de temps CPU passé en mode <code>user</code> , système inactif et attente d'E/S.
-d	pour chaque disque, affiche le nombre de ko transférés par seconde.
-t	affiche le nombre de caractères lus et écrits sur un terminal (par seconde)
-x	format étendu
-P	par partition

Principales colonnes

wsvc_t	temps d'attente dans la wait queue (milliseconde)
asvc_t	temps de service de la transaction (milliseconde)
kps	taux de transfert de données en kilooctets/seconde
tps	transactions par seconde
r/wps	transactions de lecture (écriture) par seconde
us(%)	temps CPU en mode <code>user</code>
wt(%)	temps d'attente d'Entrées/Sorties
sy(%)	temps CPU en mode <code>system</code>
id(%)	temps d'inactivité
serv	temps moyen d'utilisation du disque en millisecondes
util	pourcentage d'utilisation du disque



Les commandes Berkeley

■ fstyp

```

resa3# fstyp /dev/dsk/c0t1d0s0
ufs
resa3#

resa3# fstyp -v /dev/dsk/c0t1d0s0
ufs
magic      11954      time          Tue May 18 11:27:14 1993
sblkno     16          cblkno        24          iblkno       32dblkno184
sbsize     2048     cgsiz        1024       cgoffset    24cgmask   0xffffffff0
ncg        18          size          46170      blocks      43129
bsize     8192     shift        13          mask        0xffffe000
fsize     1024     shift        10          mask        0xfffffc00
frag      8          shift        3           fsbtodb     1
minfree   10%      maxbpg       2048       optim       time
maxcontig 7          rotdelay     0ms        rps         60
csaddr    184      cssize       1024       shift      9mask0xfffffe00
ntrak     9          nsect        36         spc         324ncyl285
cpg       16          bpg          324        fpg         2592ipg1216
nindir    2048     inopb        64         nspf        2
nbfree    4170     ndir         288        nifree      20352nffree102
cgrotor   8          fmod         0          ronly       0
file system state is valid, fsclean is 0
blocks available in each rotational position
cylinder number 0:
position 0:  0    7    9    16   18
position 1:  5   14
position 2:  3   12
position 3:  1  10   19
position 4:  8   17
position 5:  6   15

```

Les commandes Berkeley

■ **fstyp**

Le système de fichiers à examiner (pas les montages NFS) est passé en argument.

Principales options

sans options affiche le type du système de fichiers
-v affiche des informations sur le système de fichiers

Principales colonnes

Informations du superbloc et des cylindres.

magic	type du file system
ncg	nombre de cylindres par groupe
bsize	taille d'un bloc logique
fsize	taille d'un fragment
nbfree	nombre de bloc libres
mnfree	pourcentage minimum laissé libre
maxcontig	nombre de blocs maximum contigus pour un fichier ordinaire
rotdelay	temps d'attente entre chaque rotation
optim	type d'optimisation

Remarque : le flag `FSCLEAN` est visualisé.



Les commandes Berkeley

tunefs

- `tunefs [-a maxconfig][-d rotdelay]
[-e maxbpg][-m minfree][-o [s|t]]
special|filesystem`
- Le système de fichiers doit être démonté
- L'optimisation doit se faire avant que le taux d'occupation de la partition dépasse 90%

Les commandes Berkeley

tunefs

Principales options

- a *maxconfig* nombre maximum de blocs contigus qui seront lus en un seul accès, (1 par défaut)
- d *rotdelay* temps nécessaire pour servir une interruption de fin de transfert et pour initialiser un nouveau transfert sur le même disque
- e *maxbpg* nombre maximum de blocs utilisables par un même fichier sur un groupe de cylindres
- m *minfree* pourcentage de l'espace disque non accessible aux simples utilisateurs
- o [s|t] change la stratégie d'optimisation pour le système de fichiers (s pour une optimisation en taille et t pour une optimisation en temps)



Les commandes Berkeley

■ netstat

```
vancouver (root-sh) # netstat -i
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 loopback localhost 1268 0 1268 0 0 0
le0 1500 vancouver vancouver 34342 0 33796 0 5 0
```

```
vancouver (root-sh) # netstat -r
```

```
Routing Table:
Destination Gateway Flags Ref Use Interface
-----
150.20.0.0 vancouver U 3 14 le0
224.0.0.0 vancouver U 3 0 le0
localhost localhost UH 0 1212 lo0
```

```
vancouver (root-sh) # netstat -a | more
```

```
UDP
Local Address Remote Address State
-----
*.route Idle
*.* Unbound
*.sunrpc Idle
*.* Unbound
*.32771 Idle
*.name Idle
*.biff Idle
*.talk Idle
*.time Idle
*.echo Idle
*.discard Idle
*.daytime Idle
*.chargen Idle
*.32772 Idle
*.32773 Idle
*.32774 Idle
*.32775 Idle
*.lockd Idle
*.32776 Idle
*.32777 Idle
*.32778 Idle
*.32779 Idle
*.syslog Idle
*.161 Idle
*.32783 Idle
*.32784 Idle
*.32782 Idle
*.* Unbound
```

Les commandes Berkeley

■ netstat

Principales options

- a affiche l'état de toutes les communications (réseau, local)
- i affiche l'état des interfaces TCP/IP utilisées
- p affiche la table arp (arp -a)
- r affiche la table de routage
- s affiche des statistiques par protocole
- n équivalent à l'option -a, mais affiche les adresses réseau sous la forme universelle



Les commandes Berkeley

■ nfsstat

```
resa3# nfsstat -m
/mnt from resa2:/export/home
Flags:  hard,intr,dynamic read size=8192, write size=8192, retrans = 5
All:    srtt=0 (0ms), dev=0 (0ms), cur=0 (0ms)

resa3#

resa3# nfsstat -s

Server rpc:
calls      badcalls  nullrecv  badlen    xdrCALL
19         0         0         0         0

Server nfs:
calls      badcalls
18         0
null      getattr   setattr   root      lookup    readlink  read
1 6%      5 28%    0 0%      0 0%      4 22%    0 0%      5 28%
wrcache   write     create    remove    rename    link      symlink
0 0%      0 0%     0 0%      0 0%      0 0%     0 0%     0 0%
mkdir     rmdir    readdir   statfs
0 0%      0 0%     2 11%     1 6%

resa3#

resa3# nfsstat -c

Client rpc:
calls      badcalls  retrans   badxid    timeout   wait      newcred   timers
55         0         0         0         0         0         0         40

Client nfs:
calls      badcalls  nclget    nclcreate
55         0         55        0
null      getattr   setattr   root      lookup    readlink  read
0 0%      33 60%    0 0%      0 0%      4 7%     1 2%     0 0%
wrcache   write     create    remove    rename    link      symlink
0 0%      0 0%     0 0%      0 0%      0 0%     0 0%     0 0%
mkdir     rmdir    readdir   statfs
0 0%      0 0%     6 11%     11 20%

resa3#
```

Les commandes Berkeley

■ **nfsstat**

Principales options

-c	affiche des informations sur les clientes
-m	affiche des statistiques sur tous les systèmes de fichiers montés
-s	affiche des informations sur les serveurs

Principales colonnes

■ Pour un serveur

calls	le nombre total d'appels RPC reçus
badcalls	le nombre total d'appels rejetés par RPC (badlen , xdr call)
nullrecv	le nombre de faux appels RPC
badlen	le nombre d'appels RPC ayant une taille plus petite que la taille minimum d'un appel RPC.

■ Pour un client

calls	le nombre total d'appels RPC faits
badcalls	le nombre total d'appels RPC rejetés
retrans	le nombre de fois qu'un appel RPC doit être retransmis à cause d'un time out sur la réponse d'un serveur
badxid	le nombre de non reconnaissance d'une trame provenant d'un serveur
wait	le nombre de fois qu'un client n'a pas pu joindre un serveur par manque de canaux de communication.
newcred	le nombre de fois où les informations d'authentification ont du être rafraîchies
timers	le nombre de fois où la valeur du timeout calculée était supérieure ou égale à la valeur minimum spécifiée pour un appel
timout	le nombre de timeout sur des appels qui attendent la réponse d'un serveur



Les autres commandes Berkeley

- **timex, perfmeter**
- **df, snoop**

```
resa3#timex find / -name "*etc*" -print
/usr/share/lib/zoneinfo/etcetera
/usr/openwin/share/include/images/stretchNE.cursor
/usr/openwin/share/include/images/stretchNW.cursor
/usr/openwin/share/include/images/stretchSE.cursor
/usr/openwin/share/include/images/stretchSW.cursor
/usr/openwin/share/include/images/stretch_h.cursor
/usr/openwin/share/include/images/stretch_v.cursor
/usr/openwin/share/etc
/usr/openwin/etc
/etc
/etc/netconfig

real      58.26
user      2.12
sys       20.84

#
resa3# snoop arp
Using device le0 (promiscuous mode)
resa3 -> (broadcast) ARP C Who is 150.10.99.2, resa2 ?
resa2 -> resa3      ARP R 150.10.99.2, resa2 is 8:0:20:d:5b:a5
resa3 -> (broadcast) ARP C Who is 150.10.99.1, resa1 ?
resa1 -> resa3      ARP R 150.10.99.1, resa1 is 8:0:20:2:af:95
resa3# spray -c 100 -d 20 -l 2048 resa1
sending 100 packets of length 2048 to resa1 ...
      9 packets (9.000%) dropped by resa1
      246 packets/sec, 504613 bytes/sec

resa3#
resa3# df -a
Filesystem      kbytes    used    avail  capacity  Mounted on
/dev/dsk/c0t1d0s0  43129    9688    29131    25%      /
/dev/dsk/c0t1d0s6 104215   84208    9587    90%      /usr
/proc            0         0         0         0%      /proc
fd                0         0         0         0%      /dev/fd
swap             19952     140    19812     1%      /tmp
/dev/dsk/c0t1d0s7  2725         9    2446     0%      /export/home
/dev/dsk/c0t1d0s5  8869    7135     854    89%      /opt
resa3:(pid136)    0         0         0         0%      /net
resa3:(pid136)    0         0         0         0%      /home
resa2:/export/home 40727         9   36648     0%      /mnt
resa3#
```

Les autres commandes Berkeley

timex

Détaille le délai d'exécution d'une commande.

perfmeter

Donne des statistiques graphiques sur l'activité du système.

df

Affiche des informations sur les systèmes de fichiers.

snoop

Capture les données d'un réseau et affiche leur contenu.

Les autres commandes Berkeley

La commande snoop

- Permet l'analyse des paquets qui transitent sur le réseau
- Capture dans un fichier binaire :
`snoop -o packet.file`
CTRL-c

Puis visualisation avec l'option `-i`
`snoop -i packet.file`
- Restriction de la capture à l'en-tête des paquets (120 premiers octets) :
`snoop -s 120`
- Capture d'un nombre restreint de paquets :
`snoop -c 1000`
- Utiliser des filtres pour capturer un certain type d'activité réseau :
`snoop broadcast`
- L'utilisation de `awk`, `sed` et `grep` permettront d'interpréter et de synthétiser les résultats de la commande `snoop`.

Les autres commandes Berkeley

La commande snoop

La commande `snoop` permet de capturer et de visualiser le contenu des paquets qui transitent sur le réseau. Elle peut être utilisée pour déterminer la machine *source* ou *destination* de beaucoup d'anomalies réseau comme les retransmissions NFS.

Diverses options permettent de restreindre ou de filtrer les paquets analysés et, de ce fait, empêcher l'affichage d'informations trop denses que ce soit sur disque ou sur la sortie standard.



Les commandes SVR3

sar

sadc

Validation automatique

```
# Uncomment the following lines to enable system accounting. (Also
# see /var/spool/cron/crontabs/sys)

MATCH=`who -r|grep -c "[234][ ]*0[ ]*[S1]"`
if [ ${MATCH} -eq 1 ]
then
    su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
fi
```

```
#ident "@(#)sys 1.5 92/07/14 SMI" /* SVr4.0 1.2 */
#
# The sys crontab should be used to do performance collection. See cron
# and performance manual pages for details on startup.
#
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

```
# sar -f /tmp/sar.file
SunOS bear 5.6 Generic sun4m 09/11/97

01:53:14 %usr %sys %wio %idle
01:53:24 0 0 0 100
01:53:34 0 1 2 97

Average 0 0 1 99
#
```

Les commandes SVR3

sar (*system activity reporter*)

Collecte des données particulières sur l'activité du système.

sadc

Effectue une collecte automatique de données sur l'activité du système.

Validation automatique

Il est possible de valider automatiquement des relevés de mesure toutes les 20 secondes, pour cela l'administrateur supprimera les commentaires du fichier `/etc/init.d/perf` et de la crontab de `sys`.



Les commandes SVR3

sar

- -g : activité du pageur
- -u : activité CPU
- -d : activité disque
- -w : activité du swaper
- -k : occupation mémoire
- -b : buffers cache

Les commandes SVR3

sar

La commande `sar` permet de visualiser les activités de la machine, selon les options utilisées. Elle est l'équivalent des commandes `vmstat`, `iostat` et `swap`.

La commande `sar` utilise les paramètres suivants :

```
sar -options intervalle nombre-échantillons
```

Les principales options sont :

-g : activité du pageur

-u : activité CPU

-d : activité disque

-w : activité du swaper

-k : occupation mémoire

-b : buffers cache



Les commandes SVR3

sar

```
resa3# sar -b 5 10

SunOS resa3 5.3 Generic sun4c    01/18/94

11:44:41 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
11:44:46      0      2     100      2      4      53      0      0
11:44:51      0      0     100      0      0     100      0      0
11:44:56      0      0     100      0      0     100      0      0
11:45:01      6     121     95      0      0      50      0      0
11:45:06     10     123     92      1      4      76      0      0
11:45:11      8     174     95      4     11      66      0      0
11:45:16      9     127     93      6     17      61      0      0
11:45:21     10      67     85     14     27      47      0      0
11:45:26      8     133     94      0      0     100      0      0
11:45:31      6     240     97      0      0     100      0      0

Average      6      99     94      3      6      57      0      0

resa3# sar -b 5 10

SunOS resa3 5.3 Generic sun4c    01/18/94

11:45:38 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
11:45:43      9     145     94      0      0     100      0      0
11:45:49      8      44     81      5     15      66      0      0
11:45:54      3      72     96      5     30      83      0      0
11:45:59      2     177     99      4     24      82      0      0
11:46:04      1      60     98      0      0     100      0      0
11:46:09      1      20     94     17     17      0       2      0
11:46:14      0      0     100      0      0     100     43      0
11:46:19      0      0     100      0      0     100     33      0
11:46:24      0      5      96      4      4      0       7      0
11:46:29      0      1     100      1      2      50     64      0

Average      3      53     95      4      9      60     15      0
resa3#
```

Les commandes SVR3

sar

Principales options

-b vérifie l'activité des buffers

Principales colonnes

bread/s	nombre moyen (par seconde) de lectures de blocs physiques
lread/s	nombre moyen (par seconde) de lectures logiques à partir des buffers système
%rcache	pourcentage des lectures logiques effectuées dans les buffers système
bwrit/s	nombre moyen (par seconde) d'écritures physiques des buffers système vers le disque
lwrit/s	nombre moyen (par seconde) d'écritures logiques vers les buffers système
%wcache	pourcentage d'écritures logiques effectuées dans les buffers système
pread/s	nombre moyen (par seconde) de requêtes de lecture physique sur des périphériques en mode <i>raw</i>
pwrit/s	nombre moyen (par seconde) de requêtes d'écriture physique sur des périphériques en mode <i>raw</i>



Les commandes SVR3

sar

```
resa3# sar -c 5 10

SunOS resa3 5.3 Generic sun4c    01/18/94

11:49:42 scall/s  sread/s  swrit/s   fork/s   exec/s  rchar/s  wchar/s
11:49:47      414      49        25      0.59    0.79    2693    2512
11:49:52      298      39        16      0.00    0.00   186517   1236
11:49:57      345      65        27      0.20    0.20   317513   1572
11:50:02      295      37        14      0.20    0.60   105258    839
11:50:07      243      32        10      0.00    0.00   13458    665
11:50:12      377      63        20      0.00    0.00   19601   1462
11:50:17      386      59        16      0.20    0.40   28832   1227
11:50:22      485      86        39      0.00    0.00    6268   4684
11:50:27      381      83        31      0.79    0.79    6979   5841
11:50:32      536      44        20      0.40    0.40    5089   4588

Average      376      56        22      0.24    0.32   69195   2463
resa3#

resa3# sar -d 5 10

SunOS resa3 5.3 Generic sun4c    01/18/94

11:51:10  device      %busy  avque  r+w/s  blks/s  await  avserv
11:51:15  sd1          81     1.1    26     321     0.4    43.3
11:51:20  sd1          74     1.1    20     951     0.4    55.3
11:51:25  sd1          69     1.0    68     370     0.8    14.2
11:51:30  sd1          68     2.5    27     211    45.6    48.5
11:51:35  sd1          56     0.7    20     216     1.8    35.0
11:51:40  sd1          64     0.9    21     228     2.3    38.1
11:51:45  sd1          64     2.6    21     295    71.2    48.6
11:51:50  sd1          81     7.2    28     425   202.1    61.0
```

Les commandes SVR3

sar

Principales options

- c résume les appels système
- d résume l'activité disque

Principales colonnes

■ Avec l'option -c

scall/s	nombre d'appels système par seconde (tous types d'appels système)
sread/s	nombre d'appels système de lecture par seconde
swrit/s	nombre d'appels système d'écriture par seconde
fork/s	nombre d'appels système <code>fork</code> par seconde
exec/s	nombre d'appels système <code>exec</code> par seconde
rchar/s	nombre d'octets transférés par seconde par des appels système de lecture
wchar/s	nombre d'octets transférés par seconde par des appels système d'écriture

■ Avec l'option -d

device	nom du disque analysé
%busy	pourcentage de temps utilisé pendant une requête de transfert
avque	nombre moyen de requêtes non comprises durant la période analysée
r+w/s	nombre de transferts par seconde de lecture et d'écriture vers le périphérique
blks/s	nombre de blocs physiques par seconde transférés vers le périphérique
await	temps moyen (en millisecondes) d'attente en mode idle des requêtes de transfert dans une file d'attente
avserv	temps moyen, en millisecondes, pour qu'une requête de transfert soit terminée par le périphérique



Les commandes SVR3

sar

Principales options

- g** vérifie les pageout et la libération de la mémoire
- k** vérifie l'allocation mémoire du noyau

```
resa3# sar -g 5 10

SunOS resa3 5.3 Generic sun4c    01/18/94

12:30:41  pgout/s  ppgout/s  pgfree/s  pgscan/s  %ufs_ipf
12:30:46      2.19      3.38      5.17     30.62    100.00
12:30:51      0.00      0.00      0.00      0.00      0.00
12:30:56      0.00      0.00      0.00      0.00      5.12
12:31:01      0.00      0.00      0.00      0.00     13.05
12:31:06      0.00      0.00      0.00      0.00      4.51
12:31:11      0.00      0.00      0.00      0.00      5.34
12:31:16      0.00      0.00      0.00      0.00      8.60
12:31:21      0.00      0.00      0.00      0.00      4.04
12:31:26      0.00      0.00      0.00     15.34      1.99
12:31:31     14.74     18.92     42.63    492.63      2.35

Average      1.69      2.23      4.78     53.90      5.60
```

```
resa3# sar -k 5 10

SunOS resa3 5.3 Generic sun4c    01/18/94

12:34:05  sml_mem   alloc   fail   lg_mem   alloc   fail   ovsz_alloc   fail
12:34:10  842240   612224    0   851968   804864    0     2232320    0
12:34:15  842240   627792    0   851968   813056    0     2215936    0
12:34:20  842240   640080    0   851968   817152    0     2191360    0
12:34:25  842240   653472    0   851968   821248    0     2215936    0
12:34:30  842240   675616    0   851968   825856    0     2240512    0
12:34:35  849408   692192    0   851968   833536    0     2158592    0
12:34:40  856576   708640    0   851968   842752    0     2174976    0
12:34:45  856576   711072    0   851968   844288    0     2224128    0
12:34:50  856576   710976    0   851968   841728    0     2240512    0
12:34:55  856576   713296    0   868352   845312    0     2240512    0

Average   848691   674536    0   853606   828979    0     2213478    0
```


Les commandes SVR3

sar

Principales colonnes

■ Avec l'option -g

pgout/s	nombre de fois par seconde que les systèmes de fichiers reçoivent des requêtes de <i>page out</i>
ppgout/s	nombre de pages qui sont <i>pageout</i> , par seconde
pgfree/s	nombre de pages par seconde, qui sont placées dans la <i>free list</i>
pgscan/s	nombre de pages par seconde analysées par <i>pageout</i>
%ufs_ipf	pourcentage d'inodes ufs libérés

■ Avec l'option -k

sml-mem	mémoire en octets dont le KMA (Kernel Memory Allocation) dispose dans son réservoir de faibles requêtes mémoire (une faible requête mémoire est inférieure à 256 octets)
alloc	quantité de mémoire en octets que le KMA a affecté aux faibles requêtes
fail	requêtes qui ont échoué pour les faibles quantités mémoire
lg-mem	mémoire en octets dont le KMA dispose dans son réservoir de fortes requêtes mémoire (une forte requête mémoire est comprise entre 512 octets et 4 ko)
alloc	mémoire, en octets, que le KMA a affecté aux fortes requêtes
fail	requêtes qui ont échouées pour les fortes quantités mémoire
ovsz_alloc	mémoire affectée pour répondre aux requêtes supérieures à 4 ko
fail	requêtes qui ont échoué pour des quantités de mémoire supérieures à 4 ko



Les commandes de surveillance liées au développement

truss

truss -f

truss -p pid

truss -o /tmp/fic

truss -t!syscall

Les commandes de surveillance liées au développement

truss

La commande `truss` permet de suivre l'exécution d'un processus. Elle est utile tant au développeur qu'à l'administrateur.

Principales options

- f trace aussi les processus fils
- p pid du processus à tracer
- o nom du fichier de sortie
- t nom des appels systèmes à prendre en compte (ou non)



L'accounting

Qu'est-ce que l'accounting

Présentation

Les packages : SUNWaccr, SUNWaccu

Accounting périodique et cumulatif

Le script /usr/lib/acct/runacct

Facturation

Sécurité

L'accounting

Qu'est-ce que l'accounting

Présentation

L'*accounting* est un ensemble de programmes présents dans les packages SUNWaccr et SUNWaccu, qui enregistrent des informations sur l'utilisation du système et qui fournissent des rapports d'activité. L'accounting gère différents genres d'information : les connexions, les processus, l'utilisation des ressources. Ce chapitre décrit comment fonctionnent les différents programmes et comment interpréter les rapports obtenus.

Accounting périodique et cumulatif

Une fois initialisé, le système d'accounting effectue seul différentes tâches. Généralement une fois par jour, `/etc/cron` lance le script `/usr/lib/acct/runacct` qui traite les différents fichiers d'accounting pour finalement produire un fichier résumé et des rapports. La commande `/usr/lib/acct/prdaily` imprime ces rapports. Les fichiers résumés peuvent être cumulés, traités et imprimés chaque mois avec le script `/usr/lib/acct/monacct`.

Facturation

Grâce à ces rapports il est possible de facturer les utilisateurs pour l'utilisation des ressources du système. Le programme `/usr/lib/acct/chargefee` enregistre les sommes dues par chaque utilisateur dans le fichier `/var/adm/fee`.

Sécurité

Qui s'est connecté et quand ? Qui a lancé telle application et quand ?

L'accounting

Mise en œuvre de l'accounting

Programmes et scripts shell, sous les répertoires

```
/usr/lib/acct
/usr/bin
```

Fichiers et répertoires utilisés

```
/var/adm/ {
  wtmp
  pacct
  acct/nite
  acct/sum
  acct/fiscal
  ...
}
```

Crontabs des utilisateurs adm et root

<code>/usr/lib/acct/ckpacct</code>	teste <code>/var/adm/pacct</code> pour surveiller sa taille
<code>/usr/lib/acct/dodisk</code>	réalise l'accounting des disques sur les systèmes
<code>/usr/lib/acct/monacct</code>	crée des fichiers résumés mensuels dans <code>/var/adm/acct/fiscal</code> et reprend des fichiers à résumer dans <code>/var/adm/acct/sum</code>
<code>/usr/lib/acct/runacct</code>	crée les fichiers résumés quotidiens dans <code>/var/adm/acct/sum</code>

L'accounting

Mise en œuvre de l'accounting

Programmes et scripts shell

Tous les programmes et les scripts shell nécessaires pour l'accounting sont placés dans `/usr/lib/acct`. Le programme `acctcom` se situe dans `/usr/bin`. Tous les programmes d'accounting appartiennent à `bin`, sauf `acctcom` qui appartient à `root`. Le script `/usr/lib/acct/startup` est lancé depuis `/etc/init.d/acct` lors du démarrage si les liens `/etc/rc0.d/K22acct` et `/etc/rc2.d/S22acct` sont créés.

Ce script lance le script `turnacct` qui activera entre autres la commande `accton`.

Fichiers et répertoires

Le fichier `/var/adm/wtmp` pour l'accounting des connexions et le répertoire `/var/adm/acct` pour l'accounting des processus sont utilisés pour conserver les informations liées à l'accounting.

Crontab

Il faut ajouter les 3 lignes suivantes au fichier `/var/spool/cron/crontabs/adm` :

```
0 * * * * /usr/lib/acct/ckpacct
30 3 * * * /usr/lib/acct/runacct \
                2>/var/adm/acct/nite/fd2log
30 9 * * 5 /usr/lib/acct/monacct
```

Il faut rajouter la ligne suivante au fichier `/var/spool/cron/crontabs/root` :

```
30 2 * * * /usr/lib/acct/dodisk
```

Vous pouvez modifier la périodicité de lancement de ces commandes en fonction de la fréquence que vous souhaitez pour l'accounting.



L'accounting

Le script runacct

runacct

- Lancé par cron
- Procédure principale d'accounting
- Termine proprement le traitement
- Ecrit à la console
- Processus divisé en différentes étapes réentrantes

Met à jour les fichiers concernant

- les connexions
- les charges
- l'usage du disque
- les processus

Prépare les rapports journaliers et cumulatifs

L'accounting

Le script runacct

Ce programme est normalement lancé par `cron` en dehors des heures de travail. Ce script constitue la procédure principale d'accounting. Il traite les fichiers de connexion, de taxation, d'accounting disque et d'accounting processus. Il prépare les fichiers d'accounting quotidiens et cumulatifs en vue de leur impression par les commandes `prdaily` et `monacct`.

Le script comporte un ensemble de mécanismes destiné à protéger les fichiers, à reconnaître certaines erreurs, à fournir des diagnostics intelligents et à terminer proprement le travail de telle façon que `runacct` puisse être relancé. Il signale les étapes qu'il exécute en écrivant des messages dans `/var/adm/acct/nite/active` et la sortie des diagnostics est enregistrée dans `fd2log`. Quand `runacct` est lancé, il crée deux fichiers `lock` et `lock1`. Ces fichiers provoqueront l'affichage d'erreurs si `runacct` est relancé. Ils servent à empêcher le lancement de plusieurs `runacct` simultanément. Si `runacct` détecte une erreur, un message est écrit sur la console, un message est envoyé aux administrateurs `root` et `adm`, et les verrous sont supprimés. Puis les fichiers de diagnostics sont sauvegardés et l'exécution se termine.

Pour permettre à `runacct` d'être relancé en cas d'arrêt avant sa terminaison, le processus est divisé en plusieurs étapes réentrantes (l'ensemble se compose de 11 étapes). Un fichier mémorise le dernier état exécuté. Lorsque chaque étape se termine, `statefile` est mis à jour et contient la prochaine étape à exécuter. Lorsque le traitement d'une étape se termine, `statefile` est lu et l'on passe à l'étape suivante. Lorsque `runacct` atteint l'étape de nettoyage final, il retire les verrous et termine le traitement.

Un fichier `lastdate` sous `/var/adm/acct/nite` contient la date de dernière activation de `runacct`.

L'accounting

Le script runacct

Accounting quotidien

- Le script `/usr/lib/startup` est lancé quand le système passe en mode multi-utilisateurs.
- Le fichier `/var/adm/wtmp` contient les informations fournies par les commandes suivantes :
 - connexion `login, init`
 - changement de date `date`
 - reboot `acctwtmp`
 - arrêt système `shutacct`
- Le noyau écrit les noms des processus terminés dans `/var/adm/pacct`. Si `pacct` dépasse *500 blocs*, il est archivé et recréé.
- L'utilisation du disque est mémorisée par `acctdusq` et `diskusg` et est gérée par la commande `dodisk`.
- `chargefee` met à jour `fee`

Remarques :

- `runacct` s'exécute une seule fois par jour
- `prdaily` s'exécute une seule fois par jour
- `monacct` s'exécute une seule fois par mois

L'accounting

Le script runacct

1. Lors de la phase de démarrage du système en multi-utilisateurs, `/usr/lib/acct/startup` est lancé. D'autres programmes liés à l'accounting sont lancés durant cette phase de démarrage. `acctwtmp` ajoute un enregistrement dans `/var/adm/wtmp` en prenant le nom de la machine comme nom de login ; `turnacct` lance l'accounting des processus s'il est lancé avec l'option `on` ; plus particulièrement il lance `accton /var/adm/pacct` et, via le script `shell remove`, nettoie les fichiers `pacctN` et `wtmpN` laissés dans le répertoire `sum` par `runacct`.
2. Les programmes `login` et `init` enregistrent les connexions en écrivant dans `wtmp`. De même, tout changement de date est enregistré dans `wtmp` ainsi que `reboot` et `arrêt` du système.
3. Quand un processus se termine, le noyau ajoute un enregistrement dans `/var/adm/pacct`.
4. L'utilisation du disque est suivie par `acctdusg` et `diskusg` via `dodisk` en utilisant les noms de login.
5. Chaque heure, `cron` exécute le programme `ckpacct` pour vérifier que la taille de `/var/adm/pacct` n'excède pas *500 blocs*, sinon `turnacct switch` est exécuté, renommant `pacct` en `pacctN`, et créant un nouveau fichier `pacct`.
6. Si le système est arrêté, `shutacct` est exécuté et écrit un enregistrement dans `wtmp`.
7. Si un utilisateur demande la restauration d'un fichier, `chargefee` ajoute un enregistrement dans `fee`, qui sera traité au prochain lancement de `runacct` et ajouté aux autres enregistrements d'accounting.



L'accounting

runacct : Etapes réentrantes

- Processus divisé en plusieurs étapes (ou états)
- Le fichier `statefile`
- Les étapes :
 - SETUP
 - WTMPFIX
 - **CONNECT**
 - PROCESS
 - MERGE
 - FEES
 - DISK
 - MERGEACCT
 - **CMS**
 - **USEREXIT**
 - **CLEANUP**

L'accounting

runacct : Etapes réentrantes

Pour permettre à `runacct` d'être relancé, le processus est divisé en plusieurs étapes réentrantes. Le fichier `statefile` mémorise le dernier état exécuté. Lorsque chaque étape se termine, `statefile` est mis à jour et contient la prochaine étape à exécuter. Lorsque le traitement d'une étape se termine, `statefile` est lu et l'on passe à l'étape suivante. Lorsque `runacct` atteint l'étape CLEANUP, il retire les verrous et le traitement se termine

Nous ne détaillerons pas les différentes étapes, mais elles sont exécutées de la manière suivante :

- L'étape SETUP lance `turnacct switch` pour créer un nouveau fichier `pacct`, puis les fichiers dans `/var/adm/pacctN` sont déplacés dans `/var/adm/SpacctN.MMDD`. Le fichier `wtmp` est renommé et déplacé dans `/var/adm/acct/nite/wtmp.MMDD` avec la date ajouté à la fin de son nom. Un nouveau `wtmp` est créé. `closewtmp` et `utmp2wtmp` ajoutent des enregistrements dans le nouveau `wtmp` et dans `wtmp.MMDD` pour les utilisateurs actuellement connectés.
- Dans l'étape CONNECT, la commande `accton` enregistre les informations liées aux connexions dans le fichier `ctacct.MMDD` et crée les fichiers `lineuse` et `reboots`.
- Dans l'étape PROCESS, `acctprt` convertit les fichiers d'accounting des processus `/var/adm/SpacctN.MMDD` en enregistrements généraux dans `ptacctN.MMDD`.
- L'étape USEREXIT permet à l'administrateur l'insertion de tout traitement personnalisé.
- Dans l'étape CLEANUP, tous les fichiers temporaires sont effacés ou nettoyés. Puis `prdaily` est lancé et son résultat est sauvegardé dans `/var/adm/acct/sum/rprtMMDD`. Finalement, les verrous sont retirés et `runacct` se termine.



L'accounting

Rapports quotidiens d'accounting

Les quatre rapports de base, formatés par `prdaily`

■ **Rapport quotidien**

Etape CONNECT commande `acctcon`

Fournit les ligne utilisées.

■ **Rapport quotidien d'utilisation**

Etape DISK commande `diskacct`

Fournit l'utilisation des ressources par utilisateur.

■ **Rapport quotidien des commandes**

Etape CMS commande `acctcms`

Fournit l'utilisation des ressources par les commandes.

■ **Rapport des dernières connexions (login)**

Etape CMS commande `lastlogin`

Fournit par ordre chronologique les connexions réalisées.

L'accounting

Rapports quotidiens d'accounting

Les quatre rapports de base

Chaque fois qu'il est lancé, `runacct` produit quatre rapports. Ils couvrent l'accounting relatif aux connexions, l'utilisation du système par compte sur une base journalière, l'utilisation des commandes sur une base journalière et mensuelle, et l'heure et la date de la dernière connexion de chaque utilisateur.

Tous ces rapports sont écrits dans le fichier `/var/adm/acct/sum/rprtMMDD`.

Rapport quotidien

montre l'utilisation des lignes ainsi que les enregistrements d'arrêt, reboot,...

Rapport quotidien d'utilisation

montre l'utilisation des ressources système par chaque utilisateur (cpu, mémoire, lignes, disque, process, sessions).

Rapport quotidien des commandes

montre l'utilisation des ressources systèmes commande par commande.

Rapport des dernières connexions

donne la date de la dernière connexion de chaque utilisateur.



L'accounting

Rapports quotidiens : Etape CONNECT

■ Arrêts, redémarrages et reprises : reboots

```
vancouver (root-sh) # cat reboots
from Sat Jan  3 17:14:26 1998
to   Mon Apr 20 16:53:24 1998
15   system boot
8    run-level 3
3    run-level 0
3    run-level 6
5    run-level S
3    acctg on
1    run-level 1
1    runacct
1    acctcon
vancouver (root-sh) #
```

■ Utilisation des lignes de terminaux

fichier lineuse

- LINE le terminal ou port d'accès
- MINUTES la durée pendant laquelle la ligne a été utilisée
- PERCENT durée d'occupation de la ligne par rapport au temps total
- #SESS nombre de sessions réalisées sur cette ligne
- #ON sans signification
- #OFF nombre de fins de sessions et d'interruptions sur la ligne

L'accounting

Rapports quotidiens : Etape CONNECT

Dans l'étape CONNECT, le script `runacct` lance la commande `acctcon` qui génère un rapport placé à la fin du fichier `/var/adm/acct/sum/rprtMMDD`. Cette commande `acctcon` convertit une séquence d'enregistrement de début/fin de session lue depuis son entrée standard. `acctcon` peut être utilisée manuellement :

```
# acctcon -l lineuse -o reboots < /var/adm/wtmp
```

Les options `-l` et `-o` donnent les noms des deux fichiers dans lesquels la commande écrit le rapport.

La première partie du rapport affiche les lignes `from` et `to` définissant la période du rapport. Elle est suivie par les renseignements d'arrêt de la machine, `reboot`, reprises, etc... écrits dans `/var/adm/wtmp` par `acctwtmp`.

La seconde partie du rapport montre les ports utilisés, pendant combien de temps, le temps total d'utilisation, le nombre d'accès, le nombre de déconnexions

```
vancouver (root-sh) # cat lineuse
TOTAL DURATION IS 153999 MINUTES
LINE      MINUTES  PERCENT  # SESS  # ON  # OFF
/dev/pts/0 0         0         0      0    2
/dev/pts/1 0         0         0      0    7
/dev/pts/2 0         0         0      0    6
/dev/pts/3 0         0         0      0    2
/dev/pts/6 0         0         0      0    6
/dev/pts/7 0         0         0      0    2
console   3059     2         5      5   21
ftp316    0         0         2      1    6
ftp580    5         0         2      1    2
ftp581    5         0         2      1    2
pts/0     1581     1         4      4    6
pts/1     288      0         2      2    6
pts/2     1104     1         4      4    9
pts/3     1268     1         1      1    3
pts/6     8         0         1      1    6
pts/7     56       0         1      1    3
TOTALS    7375     --        24     21   89
vancouver (root-sh) #
```

L'accounting

Rapport quotidien d'utilisation : Etape PROCESS

- Utilisation des ressources par l'utilisateur
 - UID numéro du compte
 - LOGIN nom de login de l'utilisateur
 - CPU minutes pendant lequel le processus de l'utilisateur a utilisé la CPU (PRIME, NPRIME)
 - KCORE-MINS valeur cumulée de l'espace mémoire utilisé par un processus pendant son exécution, en segments de 1ko par minute (PRIME, NPRIME)
 - CONNECT(MINS) temps réel d'utilisation (temps pendant lequel l'utilisateur a été connecté au système) (PRIME, NPRIME)
 - DISK BLOCKS un bloc correspond à 512 octets
 - #OF PROCS processus lancés par l'utilisateur
 - #OF SESS sessions lancées par un utilisateur sur le système
 - #DISK SAMPLES nombre de fois que l'accounting disque a été lancé pour donner le nombre moyen de blocs disques
 - FEE livres/dollars/pesetas à la charge de l'utilisateur
 - PRIME et NPRIME sont déterminés par
/etc/acct/holidays

L'accounting

Rapport quotidien d'utilisation : Etape PROCESS

Dans l'étape PROCESS, le script `runacct` lance la commande `acctprc` et le rapport qu'elle génère sera utilisé dans le fichier `/var/adm/acct/sum/rprtMMDD`.

La commande `acctprcl` peut être utilisée manuellement. Elle fournit l'utilisation des ressources par utilisateur.

```
#acctprcl tmp < /var/adm/pacct
#cat tmp
Apr 7 19:36 1993 DAILY USAGE REPORT FOR lilas Page 1
```

UID	LOGIN NAME	CPU (MINS)		KCORE-MINS		CONNECT (MINS)		DISK BLOCKS	# OF PROCS	# OF SESS	# DISK SAMPLES	FEE
		PRIME	NPRIME	PRIME	NPRIME	PRIME	NPRIME					
0	TOTAL	7	8	93	159	1964	5416	0	2494	12	0	0
0	root	7	8	86	155	1964	5416	0	2341	12	0	0
4	adm	0	0	7	3	0	0	0	148	0	0	0
5	uucp	0	0	0	0	0	0	0	4	0	0	0
60001	nobody	0	0	0	0	0	0	0	1	0	0	0
#												

L'accounting

Rapport quotidien des commandes : Etape CMS

- Utilisation des ressources système par commande
 - COMMAND NAME nom de la commande.
Toutes les procédures shell s'appellent sh car seuls les modules objets sont pris en compte.
 - PRIME NUMBER CMDS total d'appels de cette commande
 - TOTAL KCOREMIN mesure cumulative du nombre de segments de 1ko de mémoire utilisés par un processus pendant une minute
 - PRIME TOTAL CPU-MIN temps CPU total utilisé par le programme
 - PRIME TOTAL REAL-MIN temps réel accumulé par le programme
 - MEAN SIZE-K moyenne de *TOTAL KCOREMIN* divisé par le nombre d'appels *NUMBER CMDS*
 - MEAN CPU-MIN moyenne déduite de *NUMBER CMDS* et de *TOTAL CPU-MIN*
 - HOG FACTOR temps CPU total divisé par le temps écoulé. Donne le rapport entre le temps de disponibilité et le temps d'utilisation du système
 - CHARS TRNSFD nombre total de caractères manipulés par les appels read et write (négatif en cas d'*overflow*)
 - BLOCKS READ nombre total de lectures et d'écritures de blocs physiques traitées par un processus

L'accounting

Rapport quotidien des commandes : Etape CMS

Ce rapport donne des indications sur la période d'accounting en cours tandis que le rapport mensuel donne des statistiques depuis la dernière fois que la commande `monacct` a été lancée. Le format de ces deux rapports est identique. Utilisez la commande `acctcms` comme indiqué ci-dessous si vous souhaitez obtenir manuellement un rapport sur l'utilisation quotidienne des commandes. Sinon, `runacct` la lance une fois par jour et les résultats sont consignés dans `/var/adm/acct/sum/rprtMMDD`. Ce rapport est trié sur le champ *TOTAL KCOREMIN*, valeur utile pour calculer l'utilisation des ressources du système.

```
#acctcms /var/adm/pacct > today_file
#acctcms -a -s today_file
```

L'option `-a` convertit les données en ascii et l'option `-s` annonce le nom du fichier résumé.

```
Apr  4 03:00 1998  DAILY COMMAND SUMMARY Page 1
```

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL COMMAND SUMMARY			HOG FACTOR	CHARS TRNSFD	BLOCKS READ
				TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN			
TOTALS	1303	2257.25	1.81	1327.35	1249.98	0.00	0.00	8239360	22154
dtexec	160	554.07	0.30	480.76	1839.74	0.00	0.00	1434880	6
dtscreen	160	467.55	0.30	480.56	1548.16	0.00	0.00	2167040	7
find	14	405.53	0.60	10.94	678.52	0.04	0.05	1314	21147
sh	131	171.35	0.14	100.59	1218.16	0.00	0.00	135094	2
rpc.nisd	16	103.70	0.04	0.07	2550.07	0.00	0.56	1467392	0
nisadden	32	54.76	0.03	0.18	1651.02	0.00	0.18	113456	2
NISPLUS.	1	53.49	0.01	89.00	3914.15	0.01	0.00	174208	16
nisgrep	16	49.10	0.03	0.11	1444.16	0.00	0.31	70840	0
domainna	32	33.72	0.02	0.04	1927.01	0.00	0.44	576	1
sendmail	28	33.65	0.02	0.05	1641.56	0.00	0.39	74023	31
grep	37	32.63	0.02	0.03	1938.22	0.00	0.64	415291	0
echo	32	29.12	0.01	0.02	2361.08	0.00	0.75	376	0
nisaddcr	16	28.32	0.02	0.04	1503.86	0.00	0.43	35856	1
nistblad	16	26.33	0.02	0.07	1564.44	0.00	0.24	45632	4
uudemon.	114	24.11	0.02	0.14	997.74	0.00	0.17	67765	25
sadc	44	16.67	0.03	0.06	636.94	0.00	0.44	84392	47
rm	22	15.33	0.01	0.27	1107.95	0.00	0.05	0	501
nawk	16	15.11	0.01	0.11	2324.31	0.00	0.06	1712	2
awk	18	13.95	0.01	0.03	996.48	0.00	0.43	1674	19
uuxqt	48	13.01	0.02	0.08	765.57	0.00	0.21	35568	6



L'accounting

La commande acctcom

- Utiliser `/usr/bin/acctcom` pour examiner `pacct`
 - `#acctcom /var/adm/pacct`
 - **COMMAND** nom de la commande, commence par un # si lancée par un super utilisateur
 - **USER** nom de l'utilisateur
 - **TTY** nom du terminal, ? si inconnu
 - **START TIME**
 - **END TIME**
 - **REAL TIME** en secondes
 - **CPU** en secondes
 - **MEAN SIZE** en kilo-octets

L'accounting

La commande acctcom

Le contenu des fichiers `/var/adm/pacctN` ainsi que de tous les fichiers dont les enregistrements sont au format décrit dans `acct.h`, est lu par `acctcom`. Le fichier lu par défaut est `pacct`. Le résultat donne des informations sur les processus terminés. Différentes options permettent d'avoir des informations supplémentaires.

En voici un exemple :

```
ACCOUNTING RECORDS FROM: Wed Apr 7 19:35:16 1998
COMMAND
NAME      USER      TTYNAME      START      END          REAL        CPU        MEAN
          (SECS)    (SECS)    SIZE(K)
#accton   root      ?            19:35:16  19:35:16    0.46        0.11       33.45
turnacct  adm       ?            19:35:13  19:35:16    3.01        0.08       24.00
mv        adm       ?            19:35:16  19:35:16    0.30        0.13       28.31
closewtm  adm       ?            19:35:17  19:35:17    0.15        0.09       27.11
listen    root     ?            19:35:16  19:35:16    0.81        0.31       18.58
listen    root     ?            19:35:17  19:35:17    0.81        0.28       21.00
cp        adm       ?            19:35:17  19:35:17    0.92        0.19       19.58
acctwtmp  adm       ?            19:35:18  19:35:18    0.27        0.10       23.20
cp        adm       ?            19:35:18  19:35:18    0.42        0.14       26.00
listen    root     ?            19:35:18  19:35:18    0.96        0.34       17.06
chmod     adm       ?            19:35:19  19:35:19    0.25        0.09       34.67
chgrp     adm       ?            19:35:19  19:35:19    0.46        0.13       32.92
chmod     adm       ?            19:35:21  19:35:21    0.38        0.13       27.08
chgrp     adm       ?            19:35:21  19:35:21    0.72        0.16       27.00
chown     adm       ?            19:35:22  19:35:22    0.54        0.16       27.00
...
ckpacct   adm       ?            09:00:04  09:00:06    2.44        0.18       10.67
#sh       toto     ?            09:00:04  09:00:06    2.68        0.19       23.79
ls        root     wscons       09:00:08  09:00:08    0.36        0.13       33.23
ls        root     wscons       09:00:18  09:00:18    0.36        0.13       33.54
file      root     wscons       09:00:22  09:00:23    1.59        0.24       20.50
pr        root     wscons       09:01:23  09:01:23    0.96        0.32       13.88
lp        root     wscons       09:01:23  09:01:24    1.36        0.27       19.41
ls        root     wscons       09:01:31  09:01:31    0.39        0.13       30.46
file      root     wscons       09:01:40  09:01:41    1.88        0.25       19.84
pr        root     wscons       09:02:05  09:02:06    1.19        0.29       15.45
...
more      root     wscons       09:06:57  09:06:57    0.72        0.14       33.71
ls        root     wscons       09:06:59  09:06:59    0.37        0.12       33.33
more      root     wscons       09:07:21  09:07:22    1.19        0.15       30.40
ls        root     wscons       09:08:56  09:08:56    0.39        0.13       30.46
more      root     wscons       09:09:07  09:09:09    2.42        0.15       31.73
file      root     wscons       09:09:13  09:09:13    0.47        0.21       23.62
ls        root     wscons       09:10:40  09:10:40    0.36        0.13       31.08
ls        root     wscons       09:10:49  09:10:49    0.36        0.13       33.54
acctcom   root     wscons       09:11:10  09:11:35    25.57       0.74       6.65
```



Les outils freewares

top

nfswatch

proctool

Adrian Monitor

Les outils freewares

Il existe un certain nombre d'outils freeware permettant de récupérer des résultats de tuning.

Ces derniers sont soit graphique, soit semi-graphique. De maniement plus aisé que la ligne de commande, ils fournissent une interface simple à l'administrateur.

- top

Ce logiciel est semi-graphique. Il représente une interface à la commande ps.

- nfwatch

Ce logiciel permet de suivre l'activité des services nfs.

- proctool

Ce logiciel est graphique. Il représente une interface à la commande ps.

- Adrian Monitor

Ce logiciel permet de suivre l'activité de toute une machine.



Les outils freewares

top

```

last pid: 904; load averages: 0.03, 0.03,
0.04                                     17:16:36
55 processes: 53 sleeping, 1 running, 1 on cpu
CPU states: 82.7% idle, 2.4% user, 1.4% kernel, 13.5% iowait, 0.0% swap
Memory: 27M real, 512K free, 61M swap, 22M free swap

  PID USERNAME PRI NICE  SIZE  RES STATE   TIME  WCPU   CPU COMMAND
  404 root      20  0   35M   12M sleep   12:07  0.49%  1.83% maker5X.exe
  330 root      34  0   16M  4024K sleep   16:17  0.52%  1.55% Xsun
  904 root      33  0 1904K 1336K cpu      0:00  1.12%  0.64% top
  558 root      34  0 3656K 1760K sleep    0:07  0.02%  0.05% cmdtool
  354 root      34  0 2256K 1016K sleep    0:09  0.01%  0.03% olwm
  901 root      23  0 3584K 2336K sleep    0:00  0.05%  0.02% cmdtool
  887 root      23  0 3616K 2336K sleep    0:00  0.01%  0.01% cmdtool
  890 root      23  0 1408K  864K sleep    0:00  0.01%  0.01% rlogin
  552 root      23  0 3656K  672K sleep    0:10  0.00%  0.00% cmdtool
  472 root      24  0 3640K  768K sleep    0:08  0.00%  0.00% cmdtool
  817 root      24  0 5616K 1288K run      0:00  0.00%  0.00% x_cdplayer
  903 root      -4  0 1080K  848K sleep    0:00  0.00%  0.00% csh
  889 root      15  0 1080K  688K sleep    0:00  0.00%  0.00% csh
  891 root      23  0 1408K  536K sleep    0:00  0.00%  0.00% rlogin
  374 root      23  0 3592K  312K sleep    0:00  0.00%  0.00% cmdtool

```

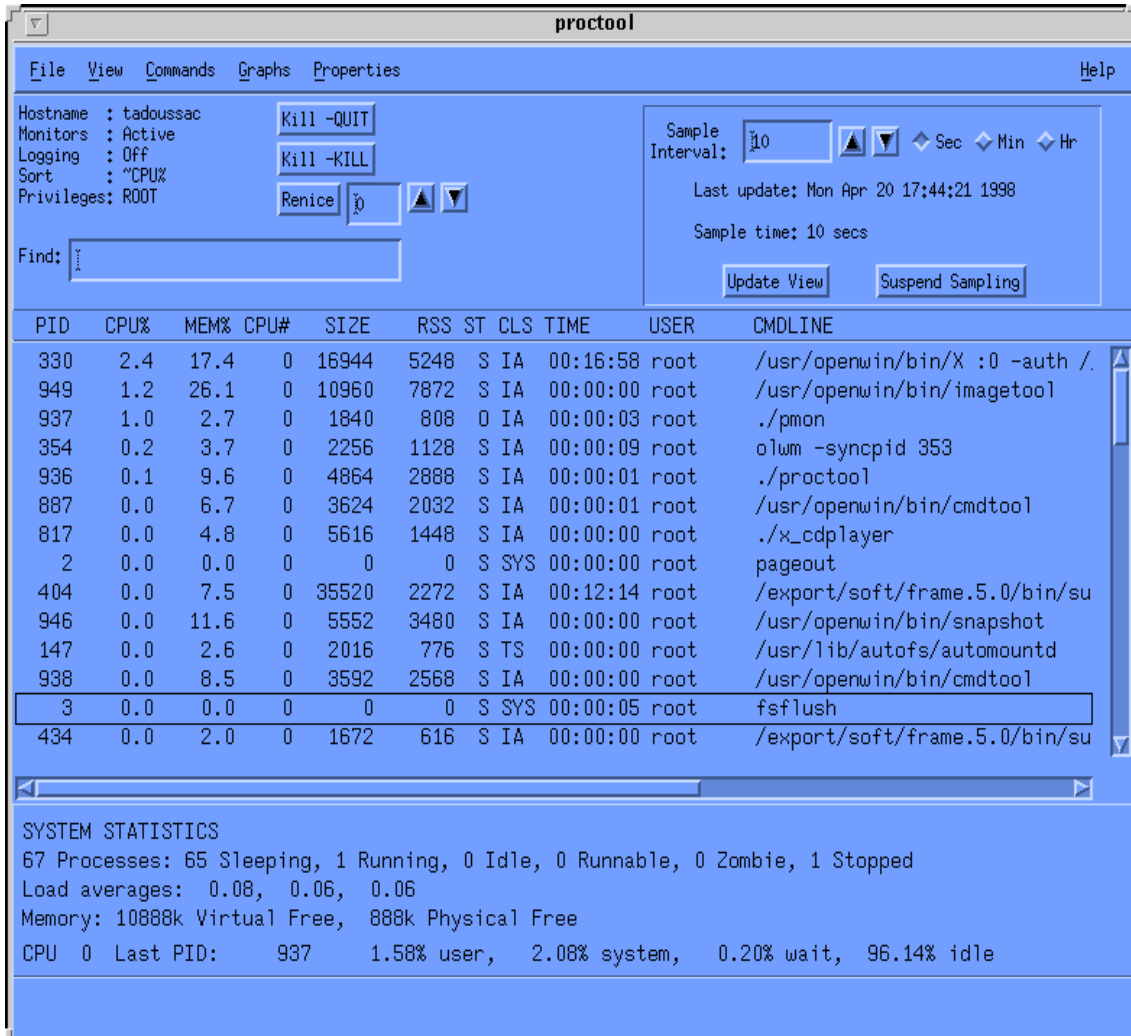
Les outils freewares

top

`top` est un outil multi-plates-formes. Il propose une interface semi-graphique à la commande `ps` et propose la surveillance de la zone de swap.

Les outils freewares

proctool



The screenshot shows the proctool application window. The title bar reads "proctool". The menu bar includes "File", "View", "Commands", "Graphs", "Properties", and "Help".

Configuration options on the left:

- Hostname : tadoussac
- Monitors : Active
- Logging : Off
- Sort : ~CPU%
- Privileges: ROOT

Control buttons: Kill -QUIT, Kill -KILL, Renice, and a numeric input field with up/down arrows.

Sampling settings on the right:

- Sample Interval: 10 (with up/down arrows and units Sec, Min, Hr)
- Last update: Mon Apr 20 17:44:21 1998
- Sample time: 10 secs
- Buttons: Update View, Suspend Sampling

A "Find:" search box is located below the configuration options.

PID	CPU%	MEM%	CPU#	SIZE	RSS	ST	CLS	TIME	USER	CMDLINE
330	2.4	17.4	0	16944	5248	S	IA	00:16:58	root	/usr/openwin/bin/X :0 -auth /.
949	1.2	26.1	0	10960	7872	S	IA	00:00:00	root	/usr/openwin/bin/imagetool
937	1.0	2.7	0	1840	808	O	IA	00:00:03	root	./pmon
354	0.2	3.7	0	2256	1128	S	IA	00:00:09	root	olum -syncpid 353
936	0.1	9.6	0	4864	2888	S	IA	00:00:01	root	./proctool
887	0.0	6.7	0	3624	2032	S	IA	00:00:01	root	/usr/openwin/bin/cmdtool
817	0.0	4.8	0	5616	1448	S	IA	00:00:00	root	./x_cdplayer
2	0.0	0.0	0	0	0	S	SYS	00:00:00	root	pageout
404	0.0	7.5	0	35520	2272	S	IA	00:12:14	root	/export/soft/frame.5.0/bin/su
946	0.0	11.6	0	5552	3480	S	IA	00:00:00	root	/usr/openwin/bin/snapshot
147	0.0	2.6	0	2016	776	S	TS	00:00:00	root	/usr/lib/autofs/automountd
938	0.0	8.5	0	3592	2568	S	IA	00:00:00	root	/usr/openwin/bin/cmdtool
3	0.0	0.0	0	0	0	S	SYS	00:00:05	root	fsflush
434	0.0	2.0	0	1672	616	S	IA	00:00:00	root	/export/soft/frame.5.0/bin/su

SYSTEM STATISTICS

67 Processes: 65 Sleeping, 1 Running, 0 Idle, 0 Runnable, 0 Zombie, 1 Stopped

Load averages: 0.08, 0.06, 0.06

Memory: 10888k Virtual Free, 888k Physical Free

CPU 0 Last PID: 937 1.58% user, 2.08% system, 0.20% wait, 96.14% idle

Les outils freewares

proctool

`proctool` est un outil freeware proposant une interface graphique pour la surveillance de la machine. Il est possible de surveiller un processus particulier et de conserver dans un fichier les traces de son exécution.

Il est nécessaire de disposer d'une version de `proctool` par version de système d'exploitation.

Les outils freewares

nfswatch

```

vancouver      Mon Apr 20 18:01:54 1998   Elapsed time: 00:02:10
Interval packets: 7 (network)          7 (to host)          0 (dropped)
Total packets:  282 (network)         282 (to host)        0 (dropped)
Monitoring packets from interface le0
      int  pct  total
ND Read      0  0%    0  TCP Packets      7 100%    281
ND Write     0  0%    0  UDP Packets      0  0%     0
NFS Read     0  0%    0  ICMP Packets    0  0%     0
NFS Write    0  0%    0  Routing Control 0  0%     0
NFS Mount    0  0%    0  Address Resolution 0  0%     1
YP/NIS/NIS+  0  0%    0  Reverse Addr Resol 0  0%     0
RPC Authorization 0  0%    0  Ethernet/FDDI Bdcst 0  0%     1
Other RPC Packets 0  0%    0  Other Packets   0  0%     0
^L   Redraw screen          u   Toggle sort by % usage
a   Display RPC authentication >   Increase cycle time by one sec
c   Display NFS client hosts <   Decrease cycle time by one sec
f   Display file systems    +   Increase cycle time by 10 secs
l   Toggle logging          -   Decrease cycle time by 10 secs
n   Toggle host numbers/names =   Reset cycle time to 10 secs
p   Display NFS procedures  ]   Scroll forward
q   Quit                    [   Scroll back
s   Write snapshot          *space bar resumes display*
--COMMANDS--

```

Les outils freewares

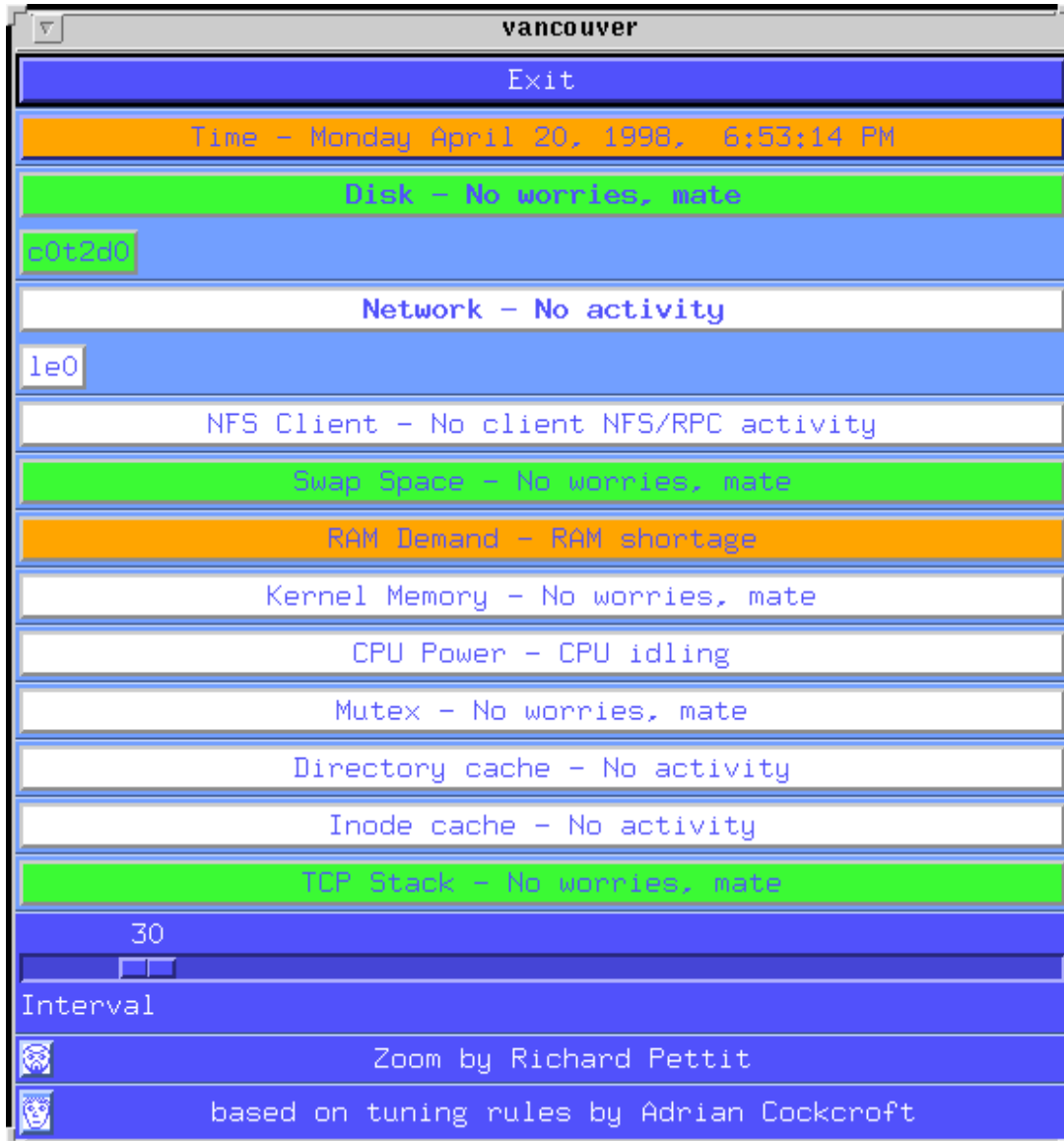
nfswatch

Cet outil encapsule les commandes `netstat` et `nfsstat`, il permet de visualiser en dynamique les échanges avec les clients NFS.



Les outils freewares

Adrian Monitor



Les outils freewares

Adrian Monitor

Ce produit se présente sous forme ligne de commande (mouchard en arrière plan) ou sous forme graphique.

Il permet de suivre l'activité des machines Sun et de remonter des alertes dès qu'un problème apparaît. Ces alertes peuvent être locales, redirigées vers `syslogd` ou vers une plate-forme SNMP (voir plus loin).

Il propose un langage complet de programmation permettant de modifier les lois induisant les alertes, voire la reprogrammation de l'interface graphique.



Les autres outils

Les outils intégrés dans les logiciels

Les outils tierce-partie

Le protocole de remonté des informations

Les autres outils

Les outils intégrés dans les logiciels

Chaque logiciel type SGBD propose son propre outil de surveillance. Il est souvent du ressort de l'administrateur de traiter les remontés d'informations et d'en déduire la marche à suivre pour améliorer les performances de la plate-forme.

Les outils tierce-partie

Il existe aussi des outils multi-plates-formes permettant de surveiller, à partir d'un seul poste, tous les serveurs voire toutes les applications présentes sur les serveurs. L'avantage de ces outils est de proposer une interface unique de surveillance et qui, de plus, souvent propose des calculs de moyennes ou des sorties graphiques plus faciles à exploiter.

Il convient de vérifier, toute fois, qu'il est possible de programmer l'intervalle d'échantillonnage en fonction des périphériques étudiés.

Le protocole de remonté des informations

Tous ces produits proposent une remonté des informations via le protocole SNMP.

Rappels sur SNMP

Manager et Agent

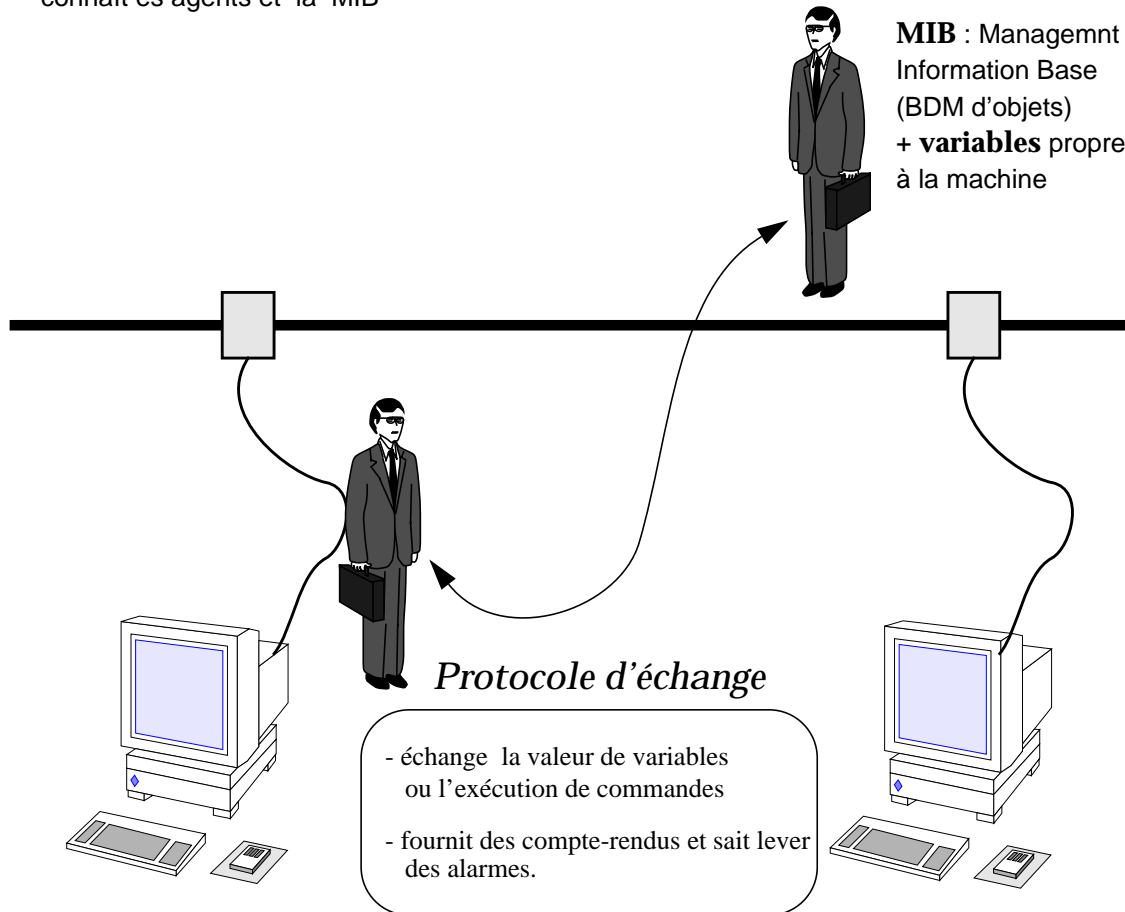
Manager Agent

connaît es agents et la MIB

Network Agent

(surveille la machine)

MIB : Managemnt
Information Base
(BDM d'objets)
+ **variables** propres
à la machine



Station d'administration

Équipement à administrer

MIB : définit le logiciel et le matériel
Base de données hiérarchisée

Rappels sur SNMP

Manager et agent

Les agents *Network Agents* sont des composants d'administration (souvent logiciels) résidant dans les entités administrables du réseau (routeurs, ponts, machines voire applicatifs).

Le poste maître ou *Network Management Station* peut communiquer avec les entités du réseau et mettre à la disposition de l'administrateur réseau les informations récoltées.

Chaque *Network Agent* maintient une base de données de gestion (appelée MIB : Management Information Base) comprenant un ensemble d'objets traduisant les éléments administrables du réseau. Le format de la MIB est normalisé par l'ISO.

Chaque composant à surveiller possède sa propre MIB d'événements et relève régulièrement la valeur des événements. Un échange entre le poste d'administration et les postes du réseau permet d'enregistrer les valeurs.

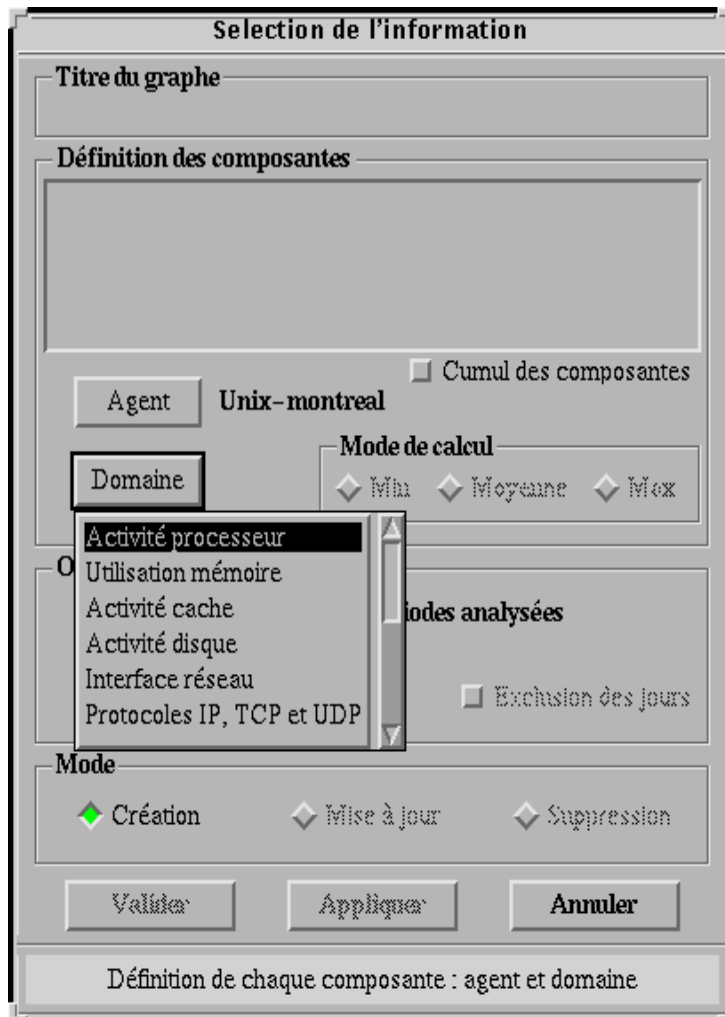
Le poste maître dispose d'un poste maître graphique lui symbolisant les éléments de son équipement.

Il peut demander la levée d'alarmes lorsque des valeurs critiques sont atteintes (ces valeurs sont programmables à partir du poste maître et peuvent être différentes pour chaque équipement).

Pour dialoguer, les deux éléments *agent* et *manager* nécessitent un protocole. A l'heure actuelle, les protocoles les plus répandus sont SNMP (Simple Network Management Protocol) et son rival de l'ISO CMIP (Common Information Management Protocol).

Produit tierce-partie

Sysload



Produit tierce-partie

Sysload

Sysload est un produit de collecte d'informations via le protocole SNMP. Il propose une partie agent (mouchard) et une partie console de surveillance.

Les agents peuvent travailler sur tout système Unix, Netware, Windows NT et sur les bases de données Oracle. La console est disponible sur tout système Unix ou windows-NT.

Le produit propose de choisir les informations à relever sur les machines (charge CPU, utilisation de la mémoire, utilisation du réseau, etc.) et d'afficher ces dernières en « temps réel » ou par historique.

Surveillance

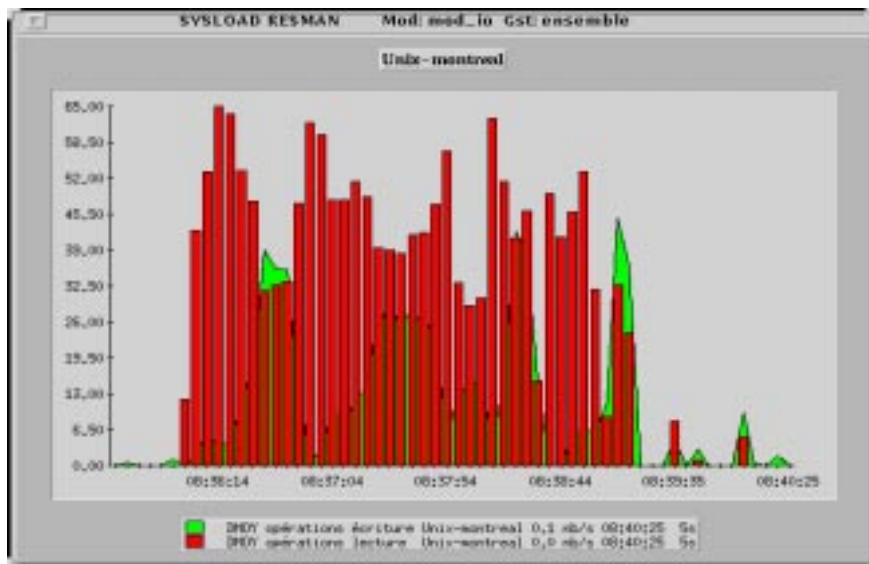
Le produit est pré-configuré pour fournir des relevés sur :

- activité processeur,
- utilisation de la mémoire,
- activité des caches,
- interface réseau
- protocole IP, TCP, UDP,
- NFS,
- analyse des connexions,
- analyse des IPC,
- analyse du système de fichiers.

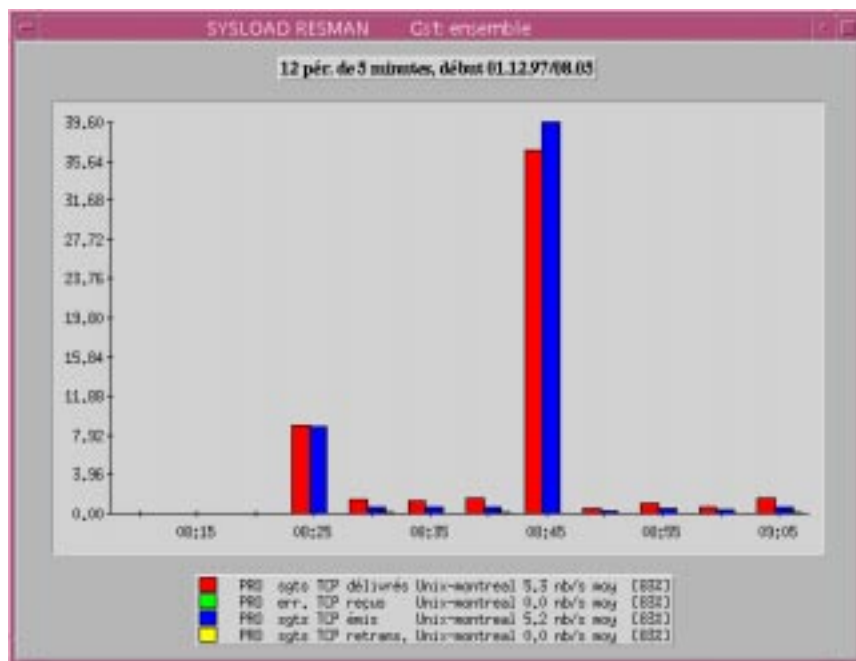
Produit tierce-partie

Sysload

Analyse temps réel



Analyse de l'historique



Produit tierce-partie

Sysload

Analyse temps réel

Cette analyse permet de suivre l'activité d'un ensemble de serveurs ou d'applications.

Analyse de l'historique

Cette analyse permet de faire un bilan sur une activité particulière, pour une période donnée.



Notes

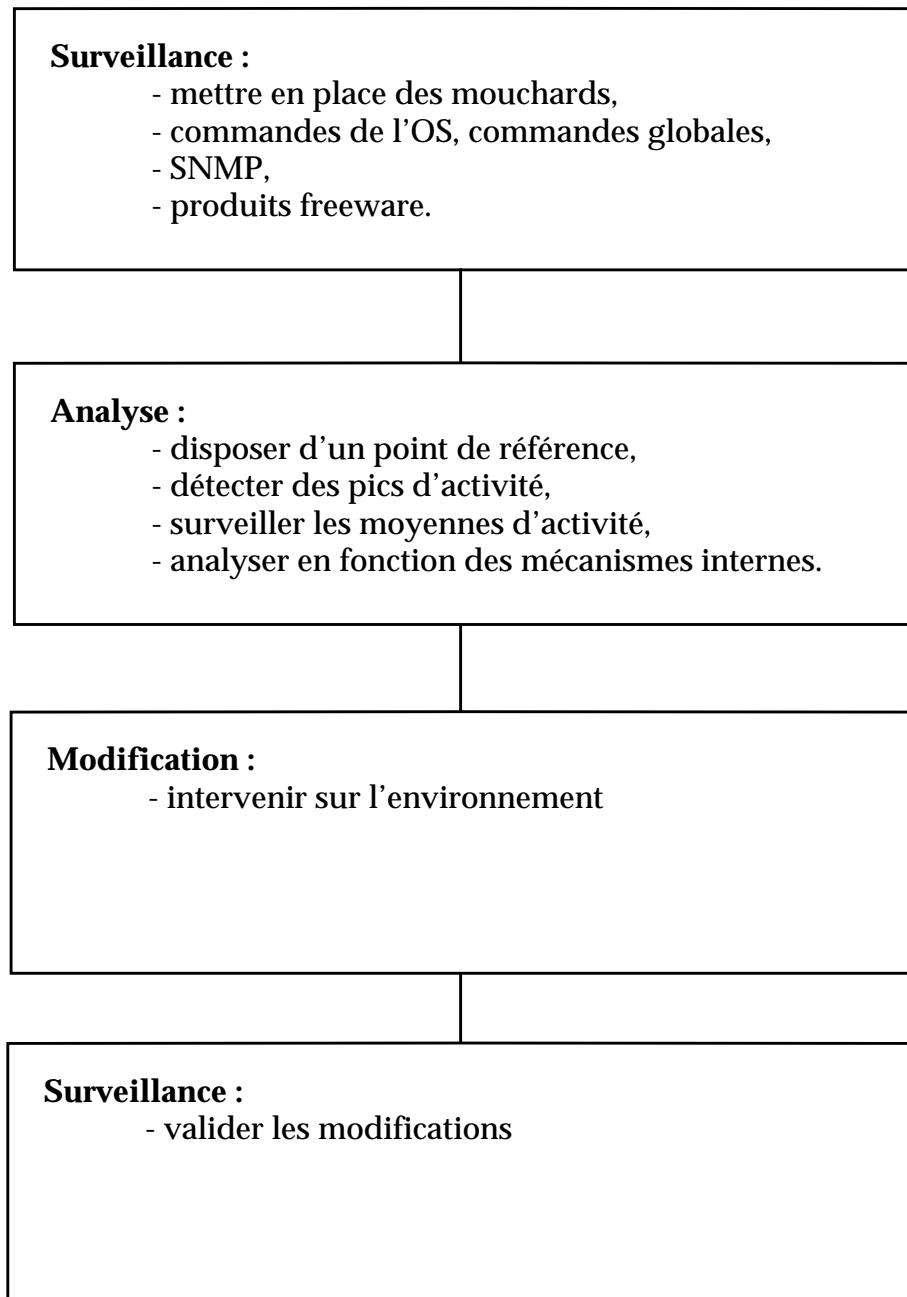
Objectifs

Les sujets couverts par ce chapitre seront les suivants :

- surveillance des activités,
- détection des goulets d'étranglement.



Algorithme de tuning



Algorithme de tuning

Maintenant que nous disposons d'une connaissance des mécanismes internes et des commandes de surveillance, nous allons analyser le résultat des commandes pour pouvoir en extraire des alertes nous indiquant que des points peuvent être améliorés sur les serveurs.



Détection des problèmes

Dysfonctionnement d'un applicatif

Faibles performances

CPU

Mémoire

Processus

Swap

Cache disque

Disque

Réseau

NFS

Détection des problèmes

Dysfonctionnement d'un applicatif

Le tuning commence par l'adéquation du logiciel et du système d'exploitation, il est donc déjà nécessaire d'arriver à faire fonctionner le logiciel.

Faibles performances

Maintenant que l'applicatif fonctionne (hors activité d'exploitation, dans un premier temps), nous allons surveiller les résultats des mouchards de surveillance pour voir où se situent les problèmes. Nous allons surveiller les activités suivantes :

- le CPU,
- la mémoire,
- les processus,
- la zone de swap,
- les caches disque (buffers des entrées/sorties, inodes, etc.),
- les activités des disques
- le réseau,
- les activités NFS.



Dysfonctionnement d'un applicatif

Dysfonctionnement lié au noyau

Sous-dimensionnement des ressources générales

Sous-dimensionnement des ressources liées au processus

```
montreal (sh) # Unable to open /dev/ptmx: No such device
montreal (sh) # Unable to open /dev/ptmx: No such device
montreal (sh) # Unable to open /dev/ptmx: No such device
montreal (sh) # sh toto
grantpt: Not enough space
System warning: Resource temporarily unavailable, call to alloc function
returned NULL pointer
XView warning: Object 0x35520, Menu_create_item: unable to allocate menu_item
(Command Menu package)

montreal% ps -edf | grep nico | wc -l
Vfork failed
montreal%

Sun Microsystems Inc.   SunOS 5.5.1   Generic May 1996
montreal% Dec  1 13:47:43 montreal unix: out of per-user processes for uid 201
```

Dysfonctionnement lié à l'applicatif

Non adéquation avec l'environnement

Fonctionnalités non implémentées

Dysfonctionnement d'un applicatif

Dysfonctionnement lié au noyau

- Sous-dimensionnement des ressources générales

Il est possible que des ressources telles :

- le nombre global de fichiers ouverts possibles,
- la limite logicielle pour le nombre de fichiers ouverts par processus,
- la taille de la zone de swap,
- le nombre de processus pouvant être validés,
- la limite des IPC,
- le nombre d'utilisateurs pouvant se connecter,
- etc.

soient sous-dimensionnées pour les applications.

Il est alors nécessaire de trouver la limite liée à ces ressources et de la modifier dans le fichier `/etc/system`.

L'administrateur se rend compte du problème via les messages systèmes présents dans la console, ou en analysant le fichier `/var/adm/messages`.

Pour certaines applications, il est nécessaire d'utiliser une commande de type `truss` pour visualiser plus nettement le problème pouvant survenir.



Dysfonctionnement d'un applicatif

Dysfonctionnement lié au noyau

Sous-dimensionnement des ressources générales

Sous-dimensionnement des ressources liées au processus

Dysfonctionnement lié à l'applicatif

Non adéquation avec l'environnement

Fonctionnalités non implémentées

Dysfonctionnement d'un applicatif

Dysfonctionnement lié au noyau

- Sous-dimensionnement des ressources liées au processus

Il est possible que des ressources telles :

- le nombre de fichiers ouverts par processus,
- la limite logicielle pour le nombre de fichiers ouverts par processus,
- le nombre de processus utilisateurs pouvant être validés,
- etc.

soient sous-dimensionnées pour les applications.

Il est alors nécessaire de trouver la limite liée à ces ressources et de la modifier dans le fichier `/etc/system`. Pour chaque application (ou utilisateur), il est aussi nécessaire de prendre en compte le résultat de la commande `limit` ou `ulimit`.

Il existe deux variables noyau :

- `rlim_fd_cur`
- `rlim_fd_max`

qui peuvent être positionnées globalement sur le système.

Pour certaines applications, il est nécessaire d'utiliser une commande de type `truss` pour visualiser plus nettement le problème pouvant survenir. Il convient de se méfier de cette commande qui alourdit légèrement le temps d'exécution des applications, voire qui sature les systèmes de fichiers.



Dysfonctionnement d'un applicatif

Dysfonctionnement lié au noyau

Sous-dimensionnement des ressources générales

Sous-dimensionnement des ressources liées au processus

Dysfonctionnement lié à l'applicatif

Non adéquation avec l'environnement

```
montreal (sh) [ora] $ svrmgrm
ld.so.1: svrmgrm: fatal: libXm.so.3: can't open file: errno=2
Killed
montreal (sh) [ora] $
```

Fonctionnalités non implémentées

Dysfonctionnement d'un applicatif

Dysfonctionnement lié à l'applicatif

- Non adéquation avec l'environnement

Certains logiciels demandent des ressources graphiques particulières (niveau de protocole X spécifique, librairies ou variables d'environnement positionnées à certaines valeurs, etc.). Il est alors nécessaire d'adapter l'environnement aux besoins de l'applicatif.

- Fonctionnalités non implémentées

Le problème ne vient pas forcément du système d'exploitation, il est possible que l'application ne propose pas encore toutes les fonctionnalités décrites dans la documentation.



Faibles performances

CPU

```
# vmstat 1
procs      memory          page          disk          faults          cpu
r  b  w    swap  free  re  mf  pi  po  fr  de  sr  s3  s6    in  sy   cs  us  sy  i
1  0  5     516    0   0   1  0  0  0  0  0  0  0    9  60   48  4  2  9
0  0  3   75980    0   0  13 184 4 236 0 76  1  0   187 245 127  6 14 8
1  0  4   75984    0   0   0 96  0 40  0 60  0  0   148 152 100  7  5 8
0  0  2   75984    0   0   0 120 0 176 0 53  0  0   131  84  86  6  5 8
.
.
.
```

Nombre de CPU

Attente sur les entrées/sorties

Faibles performances

CPU

Run Queues

Le nombre de processus dans la running queue est déterminé par la colonne `r`. Cette somme doit être divisée par le nombre de processeurs présents sur la machine.

Somme/nproc	Signification
$0 < \text{total} < 3$	normal
$3 < \text{total} < 5$	CPU chargé
$\text{total} > 5$	manque de CPU

Les processus en état `b` sont bloqués en attente d'entrées/sorties disques, réseau, terminal.

Les processus `w` sont en zone de swap. Un nombre important de processus indique un manque de mémoire centrale.

Idle time

La dernière colonne de la commande `vmstat` indique le temps de repos du CPU.

Idle time	Signification
$\text{total} > 10 \%$	normal
$\text{total} < 10 \%$	CPU chargé



Faibles performances

Mémoire

```
# sar -g 10 2
```

```
SunOS bear 5.6 Generic sun4m 09/11/97
```

Time	pgout/s	ppgout/s	pgfree/s	pgscan/s	%ufs_ipf
02:22:34					
02:22:44	1.20	12.08	21.06	23.85	7.06
02:22:54	1.10	9.00	19.30	21.30	10.81

Average	1.15	10.54	20.18	22.58	7.35
#					

```
#
```

```
# sar -r 10 2
```

```
SunOS bear 5.6 Generic sun4m 09/11/97
```

Time	freemem	freeswap
02:55:11		
02:55:22	2686	178006
02:55:32	2208	160269

Average	2447	169151
#		

```
#
```

```
# vmstat 1
```

procs			memory		page						disk			faults			cpu		
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	s3	s6	in	sy	cs	us	sy	id
1	0	5	516	0	0	1	0	0	0	0	0	0	0	9	60	48	4	2	94
0	0	3	75980	0	0	13	184	4	236	0	76	1	0	187	245	127	6	14	80
1	0	4	75984	0	0	0	96	0	40	0	60	0	0	148	152	100	7	5	88
0	0	2	75984	0	0	0	120	0	176	0	53	0	0	131	84	86	6	5	89

Faibles performances

Mémoire

La colonne `pgscan/s` indique le nombre de pages scannées par seconde. Si sa valeur excède 20 en permanence, le système manque de mémoire centrale.

sr	Signification
sr ~ 0	aucun problème
0 < sr < 15	utilisation normale
15 < sr < 30	manque de RAM
sr > 30	manque de beaucoup de RAM

Il peut être alors intéressant d'augmenter la valeur de `slowscan` pour que la scrutation ait lieu plus efficacement.

Free Memory

La colonne `free memory` indique le nombre de pages mémoire libre pour les processus. Si cette valeur est toujours inférieure à 6%, elle peut indiquer un manque de mémoire centrale.

Colonne w et sr

Si beaucoup de processus sont swappés et que la colonne `sr` prend des valeurs importantes, la machine manque de mémoire centrale.

Colonne d

Le système tente d'anticiper un manque de mémoire mais n'y parvient plus.



Faibles performances

Mémoire

```
montreal (sh) # prtconf -v | more
System Configuration: Sun Microsystems sun4m
Memory size: 32 Megabytes
System Peripherals (Software Nodes):

montreal (sh) # netstat -k | grep pp_kernel
pp_kernel 2160
           pagesfree 136
           pageslocked 3238
           pagesio 187
           pagestotal 7624

montreal (sh) #

montreal (sh) # pagesize
4096
montreal (sh) #
```

Faibles performances

Mémoire

Il est possible de vérifier la taille prise par les divers constituants du serveur.

La machine prise pour exemple possède 32 M octets de mémoire découpée en page de 4 K octets.

Le noyau voit 7624 pages au moment du boot (30 M octets).

Il utilise 2160 pages (8 M octets).

Les buffers des entrées/sorties mobilisent 187 pages (765 K octets).

Les applications laissent libres 136 pages soit 557 K octets.



Faibles performances

Processus

```
# ps -el
F S  UID  PID  PPID  C PRI NI  ADDR  SZ  WCHAN TTY  TIME CMD
8 R  7198 22028 19551 80  1 30 ff7c0000 349  ?  83:53 xlock
9 S  0 3 0 80 0 SY ff19d000 0 f00c26ae ? 265:00 fsflush
8 O  0 26070 26053 14  1 20 ff78c000 142  pts/4 0:00 ps
```

```
montreal (sh) # /usr/proc/bin/pmap 336
336:
ora_pmon_jb10
00010000 9752K read/exec dev: 32,31 ino: 165218
009A5000 184K read/write/exec dev: 32,31 ino: 165218
009D3000 84K read/write/exec
009D6000 72K [ heap ]
E000000016384K read/write/exec/shared
EF5A0000 28K read/exec /usr/lib/libw.so.1
EF5B6000 4K read/write/exec /usr/lib/libw.so.1
EF5C0000 12K read/exec /usr/lib/libmp.so.1
EF5D2000 4K read/write/exec /usr/lib/libmp.so.1
EF5E0000 12K read/exec /usr/lib/libintl.so.1
EF5F2000 4K read/write/exec /usr/lib/libintl.so.1
EF600000 508K read/exec /usr/lib/libc.so.1
EF68E000 28K read/write/exec /usr/lib/libc.so.1
EF695000 8K read/write/exec
EF6A0000 20K read/exec /usr/lib/libaio.so.1
EF6B4000 4K read/write/exec /usr/lib/libaio.so.1
EF6D0000 84K read/exec /usr/lib/libm.so.1
EF6F4000 8K read/write/exec /usr/lib/libm.so.1
EF700000 388K read/exec /usr/lib/libnsl.so.1
EF770000 36K read/write/exec /usr/lib/libnsl.so.1
EF779000 28K read/write/exec
EF790000 52K read/exec /usr/lib/libsocket.so.1
EF7AC000 8K read/write/exec /usr/lib/libsocket.so.1
EF7B0000 4K read/exec/shared /usr/lib/libdl.so.1
EF7C0000 4K read/write/exec
EF7D0000 104K read/exec /usr/lib/ld.so.1
EF7F9000 8K read/write/exec /usr/lib/ld.so.1
EFFFA000 24K read/write/exec
EFFFA000 24K [ stack ]
montreal (sh) #
```

Faibles performances

Processus

Il est important de surveiller les activités de processus. Ainsi, un processus fuyant peut être éliminé (utilisation très importante du temps CPU indiquant un dysfonctionnement).

Cette commande permet aussi de surveiller l'activité de `fsflush`.

Il est aussi possible de surveiller la quantité de mémoire utilisée par chaque processus via `proctool` par exemple.



Faibles performances

Swap

```
# vmstat 1
procs          memory          page          disk          faults          cpu
r b w   swap  free  re  mf pi po fr de sr s3 s6   in  sy  cs us sy ic
1 0 5     516    0   0   1  0  0  0  0  0  0  0    9  60  48  4  2 94
0 0 3   75980    0   0  13 184 4 236 0 76  1  0   187 245 127  6 14 80
1 0 4   75984    0   0   0 96  0 40  0 60  0  0   148 152 100  7  5 88
0 0 2   75984    0   0   0 120 0 176 0 53  0  0   131  84  86  6  5 89
```

Swap par processus

```
montreal (sh) # ls -al /proc/336
-rw-----  1 ora      dba          28450816 Dec  1 14:12 /proc/336

montreal (sh) # /usr/ucb/ps uax | grep 336
root      914  0.2  1.8   756  524 pts/8      S 17:58:37  0:00 grep 336
root      239  0.0  2.5  1336  756 ?           S 14:10:09  0:00 /usr/lib/saf/sac
ora       336  0.0  3.4 27784 1020 ?           S 14:12:23  0:00 ora_pmon_jb10
montreal (sh) # .
```

Faibles performances

Swap

La surveillance de la quantité de swap disponible est fournie par la colonne `swap` de la commande `vmstat`.

Swap	Signification
<code>swap > 100</code>	aucun problème
<code>0 < sr < 15</code>	utilisation normale
<code>15 < sr < 30</code>	manque de RAM
<code>sr > 30</code>	manque de beaucoup de RAM

Swap par processus

La quantité de swap que peut utiliser un processus peut être approchée par la visualisation de la place prise en mémoire par ce processus. Cette taille est fournie par la commande `/usr/ucb/ps aux`, ou via la visualisation de la taille prise dans `/proc` par le processus.



Faibles performances

Swap

```
# swap -a /exp/swap
# swap -l
swapfile          dev  swaplo blocks  free
/dev/dsk/c0t3d0s1 32,25    8 187912 126920
/exp/swap         -         8  20472  20472
#
```

Faibles performances

Swap

Une zone de swap doit s'équilibrer entre plusieurs disques. Chaque zone ne peut excéder 2 Go. Il est préférable de gérer des accès au raw device plutôt qu'aux fichiers.



Faibles performances

Cache disque

```
montreal (sh) # vmstat -s
    0 swap ins
    0 swap outs
    0 pages swapped in
    0 pages swapped out
...
 313486 pages examined by the clock daemon
   41 revolutions of the clock hand
 88227 pages freed by the clock daemon
   860 forks
   72 vforks
   915 execs
736977 cpu context switches
1947790 device interrupts
218588 traps
2350136 system calls
726761 total name lookups (cache hits 95%)
   970 toolong
 22961 user   cpu
 26369 system cpu
1409506 idle   cpu
 55617 wait   cpu
montreal (sh) #
```

```
montreal (sh) # sar -a 1 10

SunOS montreal 5.5.1 Generic sun4m      12/01/97

18:32:08  iget/s namei/s dirbk/s
18:32:09          0          3          2
18:32:10          0          1          2
18:32:11          0          0          0
18:32:12          0          0          0
```

Faibles performances

Cache disque

Ici, nous allons étudier les zones caches du système d'exploitation.

DNLC

L'occupation des DNLC est fournie par la colonne `cache hits`.

Cette information est à compléter avec la sortie de la commande `sar -a` (analyse de la colonne `namei` (nombre de fichiers scrutés)).

DNLC	Signification
<code>dnlc > 90 %</code>	aucun problème
<code>namei < 3</code>	aucun problème
<code>dnlc < 90 %</code>	voir ligne suivante
<code>namei > 3</code>	augmenter <code>ncsize</code>



Faibles performances

Cache disque

```
# sar -g 10 2

SunOS bear 5.6 Generic sun4m    09/11/97

02:22:34  pgout/s  ppgout/s  pgfree/s  pgscan/s  %ufs_ipf
02:22:44      1.20    12.08    21.06    23.85      7.06
02:22:54      1.10     9.00    19.30    21.30     10.81

Average      1.15     10.54    20.18    22.58      7.35
#
```

```
# netstat -k
inode_cache:
size 1223

      maxsize 583
      hits 116063
      misses 563529
      mallocs 1920
      frees 96
      maxsize reached 1897
      puts at frontlist 520643
      puts at backlist 52386
      queues to free 0 scans 4577700

#
```

Faibles performances

Cache disque

Table des inodes

L'occupation de la table des inodes est disponible via la commande `sar -g` (pourcentage des inodes libérées par manque de place).

Table des inodes	Signification
<code>ufs_ips = 0</code>	aucun problème
<code>ufs_ips != 0</code>	augmenter <code>ufs_ninode</code>

Il est aussi possible d'analyser le résultat de la commande `netstat -k`. Si la valeur `maxsize_reached` est supérieure à la valeur `maxsize`, cela signifie que la limite du cache a été atteinte. Il est alors nécessaire d'augmenter `ufs_ninode`.



Faibles performances

Cache disque

```
# netstat -k
biostats:
        buffer_cache_lookups 876534
        buffer_cache_hits 802531
        new_buffer_requests 0
        waits_for_buffer_allocs 0
        buffers_locked_by_someone 521
        duplicate_buffers_found 0
#
```

Faibles performances

Cache disque

Zone cache des buffers

Il est possible de vérifier que la machine utilise au mieux les caches associés aux systèmes de fichiers. Pour cela, nous allons calculer le ratio `buffer_cache_hits/buffer_cache_lookups` si la valeur est supérieure à 90 %, il n'est pas nécessaire d'augmenter `bufhwm`.



Faibles performances

Disque

```
# iostat -D 5
          sd0          sd1          sd2          sd3
 rps wps util  rps wps util  rps wps util  rps wps util
  0  0  0.0    0  0  0.2    0  0  0.2    19  0 56.5
  0  1  2.6    0  0  0.0    0  0  0.0    0  17 99.2
  4  0  8.0    0  0  0.0    0  0  0.0    14  0 89.3
  0  2  2.3    0  0  0.0    0  0  0.0    7  17 78.0
```

```
# iostat -D 5
          sd0          sd1          sd2          sd3
 rps wps util  rps wps util  rps wps util  rps wps util
  6  8  8.0   10  9 40.2    7  6 20.2    9  0 25.5
  0  1  2.6    8  4 36.8    5 14 45.0    7  9 39.2
  4  0  8.0    7  4 27.0    6 12 37.0    4  7 29.3
  0  2  2.3    4  6 23.0    0  8 23.3    7  1 38.0
```

```
# sar -d 10 2

SunOS bear 5.6 Generic sun4m    09/11/97

02:03:05  device          %busy  avque  r+w/s  blks/s  await  avserv
02:03:15  fd0                0      0.0    0      0      0.0    0.0
          sd2          27      0.3    17     217    0.2    17.9
          sd2,a       0      0.0    0      0      0.0    0.0
          sd2,c       0      0.0    0      0      0.0    0.0
          sd2,g       27      0.3    17     217    0.2    17.9
02:03:25  fd0                0      0.0    0      0      0.0    0.0
          sd2          35      0.4    20     235    0.8    18.4
          sd2,a       0      0.0    0      0      0.0    0.0
          sd2,c       0      0.0    0      0      0.0    0.0
          sd2,g       35      0.4    20     235    0.8    18.4
```

Faibles performances

Disque

Equilibrage des charges

Le premier problème à résoudre est celui de l'équilibrage des charges disques. La sortie de la commande `iostat` montre une différence entre les trois disques présents sur la machine.

En Solaris 2.6, les charges de chaque partition sont disponibles.



Faibles performances

Disque

```
# iostat -xnP
extended device statistics
r/s  w/s   kr/s   kw/s  wait  actv  wsvc_t  asvc_t   %w   %b  device
0.2  0.1   0.8    0.9   0.0   0.0   19.7    24.7     0    0  c0t3d0s0
0.0  0.0   0.0    0.3   0.0   0.0   24.2    102.7    0    0  c0t3d0s1
0.0  0.0   0.0    0.0   0.0   0.0    0.0     0.0     0    0  c0t3d0s2
0.0  0.0   0.0    0.0   0.0   0.0   14.4    22.4     0    0  c0t3d0s7
0.0  0.0   0.0    0.0   0.0   0.0    0.0     0.0     0    0  pancho:vold(pid251
0.1  0.0   0.7    0.0   0.0   0.0    0.0    305.8    0    4  leghorn:/opt
```

```
tadoussac# iostat -x
                                extended disk statistics
disk      r/s  w/s   Kr/s   Kw/s  wait  actv  svc_t   %w   %b
fd0       0.0  0.0   0.0    0.0   0.0   0.0   0.0     0    0
sd0       5.9  3.0  43.1   30.9   0.3   0.2  61.1     3   13
sd1       0.4  0.7   2.9    9.1   0.0   0.0  22.5     0    1
sd2       0.0  0.0   0.0    0.0   0.0   0.0  10.7     0    0
sd3       0.0  0.0   0.0    0.0   0.0   0.0  12.6     0    0
sd5       0.7  0.5   5.0    4.1   0.0   0.1  79.6     0    5
sd6       1.7  0.0   0.6    0.0   0.0   0.0   3.4     0    1
tadoussac#
```

Faibles performances

Disque

Charge des disques et des bus

Une fois les charges disques équilibrées, nous allons étudier le temps de réponse des disques et la charge de chaque disque, voire des bus SCSI.

Pour chaque disque	Signification
busy < 35 %	aucun problème
35 % < busy < 65 %	le disque est chargé
busy > 65 %	acheter d'autres disques

Pour chaque disque	Signification
svc_t < 20 ms	aucun problème
svc_t > 20 ms	le disque est lent

Pour chaque disque	Signification
w < 5 %	aucun problème
w > 5 %	le bus SCSI est saturé



Faibles performances

Système de fichiers

```
tadoussac# df -k
Filesystem      kbytes    used   avail  capacity  Mounted on
/dev/dsk/c0t0d0s0 384847 294813  51554    86%      /
/proc            0         0       0         0%      /proc
fd               0         0       0         0%      /dev/fd
/dev/dsk/c0t0d0s7 516678 446305  18713    96%      /export
swap            10680     304   10376     3%      /tmp
/dev/dsk/c0t1d0s0 1855797 898366 771861   54%      /apps
montreal:/export 384847 294813  51554    86%      /mnt
/dev/dsk/c0t5d0s0  91445   80059   2246    98%      /b
/export/home/support 516678 446305  18713    96%      /home/support
tadoussac#
```

```
# fstyp -v /dev/dsk/c0t3d0s0
ufs
magic          11954    time      Thu Sep  1 14:06:52 1994
sblkno         16       cblkno    24        iblkno    32dblkno184
sbsize         2048     cgsiz     1024      cgoffset  24cgmask0xffffffff0
ncg            39       size      100602    blocks    94033
bsize        8192     shift     13        mask      0xffffe000
fsize          1024     shift     10        mask      0xfffffc00
frag           8         shift     3         fsbtodb   1
minfree        10%      maxbpg    2048      optim    time
maxcontig 7       rotdelay 0ms       rps       60
csaddr         184      cssize    1024      shift     9mask0xfffffe00
ntrak          9         nsect     36        spc       324ncyl621
cpg            16       bpg       324       fpg       2592ipg1216
nindir         2048     inopb     64        nspf      2
nbfree         7882     ndir      1120     nifree    43382nffree429
cgrotor        1         fmod      0         ronly     0
file system state is valid, fsclean is 2
blocks available in each rotational position
....
```

Faibles performances

Systeme de fichiers

Il convient de surveiller aussi le système de fichiers. tant au niveau de son taux d'occupation que de son optimisation.

Taux d'occupation

Si le taux d'occupation est supérieur à 80 % (sur une partition accédée en lecture/écriture), le système de fichiers commence à être saturé.

Optimisation

Il est nécessaire de vérifier les paramètres utilisés pour créer les systèmes de fichiers, ainsi que le champs `optim` de la commande `fstyp`. Si ce champ passe à la valeur `space`, le système de fichiers commence à être désorganisé.



Faibles performances

Réseau

Les collisions

```
# netstat -i 10
      input   le0           output
packets errs  packets errs  colls  input (Total)  output
packets errs  packets errs  colls  packets errs  packets errs  colls
1929138 0    1590861 3    73176  1946182  0    1607905 3    73176
16      0    1        0    0      16      0    1        0    0
3       0    3        0    0      3       0    3        0    0
.
.
.
```

```
tadoussac# netstat -i
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 loopback localhost 506 0 506 0 0 0
le0 1500 150.20.0.0 tadoussac 233923 0 205601 0 62 0
tadoussac#
```

Faibles performances

Réseau

Les collisions

Dans un premier temps, le taux de collisions doit être calculé, il correspond aux nombres de collisions par le nombre de paquets émis.

Ethernet	Signification
$\text{coll}/(\text{output} \times 100) < 5 \%$	aucun problème
$\text{coll}/(\text{output} \times 100) > 5 \%$	le réseau est chargé

Détection des problèmes matériels

Il est aussi possible de détecter des problèmes matériels via la commande netstat :

Ethernet	Signification
$\text{Ierrs}/(\text{Ipkts} \times 1000) < 0,2 \%$	aucun problème
$\text{Ierrs}/(\text{Ipkts} \times 1000) > 0,2 \%$	problème matériel



Faibles performances

Réseau

Nombre de sessions ouvertes

```
tadoussac# netstat -s
TCP      tcpRtoAlgorithm      =      4  tcpRtoMin            =    200
         tcpRtoMax            = 60000  tcpMaxConn           =     -1
         tcpActiveOpens      =    102  tcpPassiveOpens      =    154

         tcpCurrEstab        =      7
```

Syn Attack

```
tadoussac# netstat -s
.      tcpListenDrop        =      0  tcpListenDropQ0     =      0
      tcpHalfOpenDrop    =      0
.
.
```

Fin de connexion

```
tadoussac# netstat -a | grep TIME
tadoussac.login      vancouver.1023  8760      0  8760      0 TIME_WAIT
tadoussac#
```


Faibles performances

Réseau

Nombre de sessions ouvertes

Chaque session est gourmande en mémoire. Il est donc nécessaire de prendre en compte ce paramètre pour dimensionner la mémoire centrale.

Le nombre de connexions est fourni par le paramètre `tcpPassiveOpens`. Cette commande peut être complétée avec `netstat -a | grep LIS | wc -l`.

Syn Attack

Ici, il est nécessaire de surveiller le réseau pour ne pas tomber dans des refus de service. Il peut être nécessaire de reprogrammer `tcp_conn_req_max` qui est positionné par défaut à 128, et `tcp_conn_req_max_q0` qui est positionné par défaut à 1024.

Les SYN Attack sont mémorisés dans le compteur `tcpHalfOpenDrop`.

Si le contenu du compteur `tcpListenDrop` est différente de 0, le serveur reçoit trop de connexions.

Fin de connexion

Si le temps de persistance d'une connexion fermante est long. Il est nécessaire de reprogrammer la variable `tcp_close_wait_interval`.



Faibles performances

NFS

```
# nfsstat -rc
```

```
Client rpc:
```

```
Connection oriented:
```

calls	badcalls	badxids	timeouts	newcreds	badverf
3394	0	0	0	0	0
timers	cantconn	nomem	interrupts		
0	0	0	0		

```
Connectionless:
```

calls	badcalls	retrans	badxids	timeouts	newcred
16	1	0	0	0	0
badverfs	timers	nomem	cantsend		
0	7	0	0		

```
# nfsstat
```

```
Client nfs:
```

calls	badcalls	clgets	cltoomany
387	0	387	0

```
Version 2: (0 calls)
```

null	getattr	setattr	root	lookup	readlink
0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
read	wrcache	write	create	remove	rename
0 0%	0 0%	0 0%	0 0%	0 0%	0 0%
link	symlink	mkdir	rmdir	readdir	statfs
0 0%	0 0%	0 0%	0 0%	0 0%	0 0%

```
Version 3: (370 calls)
```

null	getattr	setattr	lookup	access	readlink
0 0%	37 10%	0 0%	0 0%	207 55%	0 0%
read	write	create	mkdir	symlink	mknod
2 0%	0 0%	0 0%	0 0%	0 0%	0 0%
remove	rmdir	rename	link	readdir	readdirp
0 0%	0 0%	0 0%	0 0%	0 0%	121 32%
fsstat	fsinfo	pathconf	commit		
0 0%	3 0%	0 0%	0 0%		

Faibles performances

NFS

Évaluation des performances chez le client

Nous commencerons par analyser les performances sur un client TCP.

Pour chaque client	Signification
timeout < 5 % calls	aucun problème
timeout > 5 % calls et badxid ~ 0	problème réseau
timeout > 5 % calls et badxid ~ timeout	serveur lent (modifier rsize et wsize)



Faibles performances

NFS

```
# nfsstat -m
/usr/dist/local from softdist:/usr/dist/local
Flags:  hard,intr,dynamic read size=8192, write size=8192,  retrans = 5
Lookups: srtt=7 (17ms), dev=3 (15ms), cur=2 (40ms)
All:      srtt=7 (17ms), dev=3 (15ms), cur=2 (40ms)

/Doc from gotlib:/Doc
Flags:  hard,intr,dynamic read size=8192, write size=8192,  retrans = 5
Lookups: srtt=7 (17ms), dev=3 (15ms), cur=2 (40ms)
Reads:  srtt=7 (17ms), dev=3 (15ms), cur=2 (40ms)
All:      srtt=7 (17ms), dev=4 (20ms), cur=2 (40ms)

/home/Mail from mygale:/var/mail
Flags:  hard,intr,dynamic read size=8192, write size=8192,  retrans = 5
Lookups: srtt=8 (20ms), dev=5 (25ms), cur=3 (60ms)
Reads:  srtt=15 (37ms), dev=7 (35ms), cur=5 (100ms)
All:      srtt=8 (20ms), dev=5 (25ms), cur=3 (60ms)

/usr/Local/bin from softdist:/usr/Local/bin
Flags:  hard,intr,dynamic read size=8192, write size=8192,  retrans = 5
Lookups: srtt=8 (20ms), dev=4 (20ms), cur=3 (60ms)
Reads:  srtt=30 (75ms), dev=7 (35ms), cur=7 (140ms)
All:      srtt=26 (65ms), dev=8 (40ms), cur=7 (140ms)

/home/afaucill from poulpe:/export/home0/SMCC/afaucill
Flags:  hard,intr,dynamic read size=8192, write size=8192,  retrans = 5
Lookups: srtt=7 (17ms), dev=4 (20ms), cur=2 (40ms)
Reads:  srtt=7 (17ms), dev=4 (20ms), cur=2 (40ms)
Writes: srtt=48 (120ms), dev=11 (55ms), cur=11 (220ms)
All:      srtt=7 (17ms), dev=4 (20ms), cur=2 (40ms)

# nfsstat -rc

Client rpc:
calls      badcalls  retrans   badxids   timeouts  waits     newcreds
26863      1         27        7         27        0         0
badverfs   timers    toobig    nomem     cantsend  buflocks
0          44        0         0         0         0         0
```

Faibles performances

NFS

Évaluation des performances chez le client

Maintenant, nous allons travailler sur un client UDP.

Il convient de savoir sur quel point de montage intervenir. Une indication nous est fournie via la commande `nfsstat -m` (qui est à croiser avec des commandes `iostat` et `sar -d` sur le serveur).

Pour chaque client	Signification
<code>timeout < 5 % calls</code>	aucun problème
<code>timeout > 5 % calls</code> et <code>badxid ~ 0</code>	problème réseau
<code>timeout > 5 % calls</code> et <code>badxid ~ timeout</code>	serveur lent (modifier <code>timeo</code>)
<code>timeout > 5 % calls</code> et <code>badxid ~ retrans</code>	serveur lent (modifier <code>timeo</code>)
<code>retrans > 5 % calls</code>	réseau lent (modifier <code>rsize</code> et <code>wsize</code>)



Faibles performances

NFS

```
# nfsstat -rc

Client rpc:
Connection oriented:
calls      badcalls   badxids    timeouts  newcreds   badverf
3394       0          0          0         0          0
timers     cantconn   nomem      interrupts
0          0          0          0
Connectionless:
calls      badcalls   retrans    badxids    timeouts   newcred
16         1          0          0          0          0
badverfs   timers     nomem      cantsend
0          7          0          0
```

```
# nfsstat
Client nfs:
calls      badcalls   clgets     cltoomany
387        0          387        0
Version 2: (0 calls)
null       getattr    setattr    root        lookup      readlink
0 0%       0 0%       0 0%       0 0%       0 0%       0 0%
read       wrcache    write      create      remove      rename
0 0%       0 0%       0 0%       0 0%       0 0%       0 0%
link       symlink    mkdir      rmdir      readdir     statfs
0 0%       0 0%       0 0%       0 0%       0 0%       0 0%
Version 3: (370 calls)
null       getattr    setattr    lookup      access      readlink
0 0%       37 10%     0 0%       0 0%       207 55%    0 0%
read       write      create     mkdir      symlink     mknod
2 0%       0 0%       0 0%       0 0%       0 0%       0 0%
remove     rmdir      rename     link        readdir     readdirp
0 0%       0 0%       0 0%       0 0%       0 0%       121 32%
fsstat     fsinfo     pathconf   commit
0 0%       3 0%       0 0%       0 0%
```

Faibles performances

NFS

Évaluation des performances chez le client

Quel que soit le client, des paramètres sont systématiquement à étudier.

Pour chaque client	Signification
readlink < 5 %	aucun problème
readlink > 5 %	supprimer les liens symboliques
newcred = 0	aucun problème
newcred != 0	synchroniser le serveur et le client
null = 0	aucun problème
null != 0	augmenter le timeout du processus automountd
getattr < 40 %	aucun problème
getattr > 40 %	augmenter actimeo

Il est aussi conseillé de vérifier que les points de montage sont bien en read only (quand nécessaire). Ainsi, la vérification de la cohérence (valeur actimeo) n'est pas enclenchée.



Faibles performances

NFS

Cas du serveur

```
# nfsstat
Server nfs:
calls      badcalls
33767      0
Version 2: (594 calls)
null      getattr      setattr      root      lookup      readlink      read
0 0%      34 5%        3 0%         0 0%       378 63%     0 0%          0 0%
wrcache   write         create        remove     rename      link          symlink
0 0%      134 22%     3 0%         0 0%       0 0%        0 0%          0 0%
mkdir     rmdir         readdir       statfs
0 0%      0 0%        36 6%        6 1%
Version 3: (23015 calls)
null      getattr      setattr      lookup      access      readlink      read
61 0%     4638 20%    715 3%      3235 14%    8369 36%     213 0%      731 3%
write     create        mkdir         symlink      mknod       remove        rmdir
311 1%    186 0%     40 0%       0 0%        0 0%        3 0%         0 0%
rename    link          readdir       readdir+    fsstat      fsinfo        pathconf
3 0%      0 0%       263 1%      4154 18%    6 0%        62 0%        6 0%
commit
19 0%

Server nfs_acl:
Version 2: (0 calls)
null      getacl       setacl        getattr      access
0 0%      0 0%         0 0%         0 0%         0 0%
Version 3: (10158 calls)
null      getacl       setacl
0 0%      10158 100%  0 0%
```


Faibles performances

NFS

Cas du serveur

Le serveur est avant tout un serveur de fichiers (espace disques), il doit donc disposer de bonnes performances sur ce type de périphériques.

Pour chaque serveur	Signification
read + write > 50 %	serveur de data vérifier les buffers disques
read + write < 50 %	serveur d'attribut vérifier les DNLC
write >> read	utiliser des caches disques
write << read	utiliser caches sur les clients

Il est aussi conseiller de vérifier que les points de montage sont bien en read only (quand nécessaire). Ainsi, la vérification de la cohérence (champ dupchecks) n'aura pas lieu systématiquement.

Nombre de nfsd

Pour chaque serveur	Nombre de nfsd
1 CPU	64
1 interface réseau 10 Mb/s	16
1 interface réseau 100 Mb/s	160
1 client actif	2



Faibles performances

NFS

Cas du serveur

```
# vmstat 1
procs      memory          page          disk          faults          cpu
r b w    swap  free  re  mf pi po fr de sr s3 s6   in   sy   cs us sy
1 0 5     516    0   0   1 0 0 0 0 0 0 0    9   60   48 4 2
0 0 3   75980    0   0  13 184 4 236 0 76 1 0   187 245 127 6 14
1 0 4   75984    0   0   0 96 0 40 0 60 0 0   148 152 100 7 5
0 0 2   75984    0   0   0 120 0 176 0 53 0 0   131 84 86 6 5
.
.
.
```

Faibles performances

NFS

Cas du serveur

Un serveur NFS valide des processus en priorité Time Sharing, il a besoin de temps en mode SYSTEM :

Temps CPU	Signification
SYSTEM > 60 %	aucun problème
SYSTEM < 60 %	trop de temps passé en mode USER (supprimer des applications)



Faibles performances

Bases de données

CPU

Mémoire

Processus

Swap

Cache disque

Disque

Réseau

Faibles performances

Bases de données

Les serveurs de base de données vont avoir des besoins spécifiques que nous allons analyser en terme de :

- CPU,
- mémoire,
- processus,
- swap,
- cache disque,
- disque,
- réseau.

Nous allons reprendre les explications vues précédemment, dans une optique base de données.



Faibles performances

Bases de données

CPU

```
# vmstat 1
procs      memory
r b w      swap  free  re  mf pi po fr de sr s3 s6      faults      cpu
1 0 5       516   0    0   1  0 0 0 0 0 0 0      9   60   48  4  2 9
0 0 3      75980  0    0  13 184 4 236 0 76 1 0     187 245 127 6 14 8
1 0 4      75984  0    0   0 96 0 40 0 60 0 0     148 152 100 7  5 8
0 0 2      75984  0    0   0 120 0 176 0 53 0 0     131  84  86 6  5 8
.
.
.
```

Faibles performances

Bases de données

CPU

Une base de données valide des processus en priorité Time Sharing, elle a besoin de temps en mode USER :

Temps CPU	Signification
USER > 60 %	aucun problème
USER < 60 %	trop de temps passé en mode SYSTEME (réorganisation)

Ce temps doit aussi être calculé sur une journée pour prendre en compte les variations qui peuvent être révélatrices d'une désorganisation des zones de stockage.



Faibles performances

Bases de données

Mémoire

```
montreal (sh) # ipcs
IPC status from <running system> as of Tue Dec  2 10:55:09 1997
Message Queue facility not in system.
Shared Memory:
m      0 0x500182ac --rw-r--r--      root      root
Semaphore facility not in system.
```

```
SVRMGR> startup
ORACLE instance started.
Total System Global Area          4183756 bytes
Fixed Size                          39696 bytes
Variable Size                       4012988 bytes
Database Buffers                    122880 bytes
Redo Buffers                          8192 bytes
Database mounted.
Database opened.
SVRMGR>
```

```
montreal (sh) # ipcs
IPC status from <running system> as of Tue Dec  2 10:57:01 1997
Message Queue facility not in system.
Shared Memory:
m      0 0x500182ac --rw-r--r--      root      root
m      1 0x08071dfa --rw-r-----     ora       dba
Semaphores:
s      0 00000000 --ra-r-----     ora       dba
s      1 00000000 --ra-r-----     ora       dba
```

Faibles performances

Bases de données

Mémoire

IPC

Une base de données utilise beaucoup d'IPC de type share memory. Il est donc nécessaire d'adapter ces limites dans `/etc/system`. Le segment utilisé est en mémoire centrale, il ne faut donc jamais espérer fournir plus que la RAM disponible (voir début du chapitre et la commande `netstat -k`).

Zone cache

En fonction du type d'implantation des accès disques (raw devices, ou systèmes de fichiers), la base de données peut nécessiter une allocation supplémentaire de buffers (voir début du chapitre).



Faibles performances

Bases de données

Processus

```
montreal (sh) [ora] $ ps -edf | grep ora
ora 378 349 0 11:03:29 pts/2 0:00 grep ora
ora 349 344 0 10:55:51 pts/2 0:00 -sh
ora 361 1 0 10:56:51 ? 0:00 ora_pmon_jb10
ora 363 1 0 10:56:51 ? 0:00 ora_dbwr_jb10
ora 365 1 0 10:56:51 ? 0:00 ora_lgwr_jb10
ora 367 1 0 10:56:51 ? 0:00 ora_ckpt_jb10
ora 369 1 0 10:56:52 ? 0:00 ora_smon_jb10
ora 371 1 0 10:56:52 ? 0:00 ora_reco_jb10
montreal (sh) [ora] $
```

Faibles performances

Bases de données

Processus

Les processus attachés à la base de données sont :

- lourds en espace mémoire utilisée (analyse de `/proc`),
- validés en priorité Time sharing (`ps -ec`),
- peuvent être validés sous forme de thread.

Il est donc important de surveiller :

- la taille et l'occupation de la zone de swap,
- vérifier qu'ils sont bien prioritaires (cas de plusieurs bases de données, ou plusieurs applications), voire en changer la priorité via la commande `priocntl`,
- si le choix du multi-thread a été effectué, il est nécessaire de surveiller la zone de swap (ce type d'architecture peut être très gourmande en quantité de mémoire secondaire utilisée).



Faibles performances

Bases de données

Swap

```
montreal (sh) # vmstat 5
procs      memory          page          disk          faults          cpu
 r  b  w    swap  free  re  mf  pi  po  fr  de  sr  s3  --  --  --  in  sy  cs  us  sy  id
0  0  0  121536  3964   2  74  59  13  29   0  25  13   0  0  0  44  231  56  6  8  86
0  0  0  121132   632   0   2   0   0   0   0   0   0   0  0  0  4  52  23  0  0  100
0  0  0  121068   644   0  126   0  50  62   0  19   1   0  0  0  12  303  42  4  10  86
0  0  0  120836   720   0  97   9  25  28   0   6   3   0  0  0  21  265  43  5  8  88
0  0  0  120776   720   0   0   0   0   0   0   0   0   0  0  0  4  52  22  0  0  100
```

SVRMGR> **startup**

ORACLE instance started.

Total System Global Area 4183756 bytes

Fixed Size 39696 bytes

Variable Size 4012988 bytes

Database Buffers 122880 bytes

Redo Buffers 8192 bytes

Database mounted.

Database opened.

```
montreal (sh) # vmstat 5
procs      memory          page          disk          faults          cpu
 r  b  w    swap  free  re  mf  pi  po  fr  de  sr  s3  --  --  --  in  sy  cs  us  sy  id
0  0  0  120856  3340   1  73  66  29  60   0  36  14   0  0  0  48  226  58  6  8  87
0  0  0  107768   808   0   2   0   0   0   0   0   0   0  0  0  5  62  26  0  0  100
0  0  0  107768   808   0   0   0   0   0   0   0   0   0  0  0  5  66  25  0  0  100
0  0  0  107768   808   0   0   0   0   0   0   0   9   0  0  0  46  105  30  1  2  98
0  0  0  107768   816   0   0   0   3   3   0   0   1   0  0  0  8  67  25  0  0  100
```

Faibles performances

Bases de données

Swap

La taille de la zone de swap peut être soumise à forte contribution dans ce type d'environnement. Les paramètres cités précédemment indiquent qu'une surveillance de ce mécanisme est plus que nécessaire (se référer au début du chapitre).



Faibles performances

Bases de données

Cache disque

```
# netstat -k
biostats:
    buffer_cache_lookups 876534
    buffer_cache_hits 802531
    new_buffer_requests 0
    waits_for_buffer_allocs 0
    buffers_locked_by_someone 521
    duplicate_buffers_found 0
#
```

Caches internes

```
SQL> select sum(gets) "nombre d'accès au dico",
 2 sum (getmisses) "nombre d'accès sans cache"
 3 from v$rowcache;

nombre d'accès au dico nombre d'accès sans cache
-----
                 3856                 316

SQL>
```

Faibles performances

Bases de données

Cache disque

Les caches disques seront essentiellement sollicités si une implantation sur un système de fichiers a été choisie. Dans le cas du raw device, les buffers sont (dans leur grande majorité) gérés par le SGBD.

Caches internes

La base de données dispose de caches internes qu'il est nécessaire de surveiller. Les mêmes limites seront reconnues (si le cache est utilisé à plus de 90 %, il est bien dimensionné, sinon une investigation est nécessaire).



Faibles performances

Bases de données

Disque

```
SQL> select name, phyrd, phywrts
 2  from v$datafile df, v$filestat fs
 3  where df.file# = fs.file#;
```

NAME	PHYRDS	PHYWRTS
/ORACLE/DESSIN/base/system.dbs	866	15
/ORACLE/DESSIN/base/donnee_user.dbf	3	1
/ORACLE/DESSIN/base/donnee_user1.dbf	1	1
/ORACLE/DESSIN/base/burps.dbs	13	0
/ORACLE/DESSIN/base/burps1.dbs	2	1
/ORACLE/DESSIN/base/litoto	0	0
/ORACLE/DESSIN/base/sys2.dbf	0	0

7 rows selected.

Système de fichiers

Faibles performances

Bases de données

Disque

L'équilibrage des charges disques est aussi très importante, il est nécessaire de gérer cet équilibrage avec l'administrateur de la base de données. Il est fortement conseillé de ne pas stocker de zone de swap sur un disque contenant un journal de la base de données.

Système de fichiers

Si un système de fichiers a été choisit pour stocker les informations de la base de donnée, il est nécessaire de modifier les paramètres de ce système de fichiers :

- nombre d'inodes par blocs,
- minfree.



Faibles performances

Bases de données

Réseau

Nombre de sessions ouvertes

```
tadoussac# netstat -s
TCP      tcpRtoAlgorithm      =      4 tcpRtoMin      =      200
         tcpRtoMax            = 60000 tcpMaxConn        =      -1
         tcpActiveOpens     =      102 tcpPassiveOpens   =      154
         tcpCurrEstab      =          7
.
```

Syn Attack

```
tadoussac# netstat -s
.         tcpListenDrop        =          0 tcpListenDropQ0    =          0
         tcpHalfOpenDrop    =          0
.
```

Fin de connexion

```
tadoussac# netstat -a | grep TIME
tadoussac.login      vancouver.1023  8760      0 8760      0 TIME_WAIT
tadoussac#
```

Faibles performances

Bases de données

Réseau

Le réseau est soumis à forte contribution sur un serveur de base de données, il est nécessaire de surveiller :

- le nombre de connexions (elles sont souvent longues en temps),
- le nombre de sessions pendantes (`IDLE_TIME` de la connexion),
- le temps de déconnexion.

Pour les bases de données interrogeables par des utilisateurs distants, la question du SYN Attack peut se poser.

Faibles performances

Serveur WEB

Processus

Réseau

Faibles performances

Serveur WEB

Les serveurs WEB vont avoir des besoins spécifiques que nous allons analyser en terme de :

- processus,
- réseau.

Nous allons reprendre les explications vues précédemment, dans une optique WEB.

Un serveur WEB doit prendre en compte un grand nombre de connexions. Les informations échangées sont souvent en petites quantités, mais il peut s'avérer que des téléchargements soient lourds en ressources mobilisées.



Faibles performances

Serveur WEB

Processus

- Serveur concurrent
- Exécution de scripts

Faibles performances

Serveur WEB

Processus

■ Serveur concurrent

Le serveur `httpd` est un processus concurrent. Il crée un processus fils par client (ou par groupe de clients). Le nombre de processus validés au démarrage est programmable, ainsi que le nombre de requêtes traitées par processus. Dans le serveur Netscape, un fichier de message (`log`) indique si la limite a été atteinte.

Chaque processus mobilise 440K octets de mémoire.

■ Exécution de scripts

Il faut y ajouter les exécutions des scripts CGI, voire les transferts d'images pouvant être demandés lors d'une requête.

Cette machine va donc nécessiter beaucoup de RAM, voire beaucoup de swap.

L'équilibrage des charges disques reste fondamental comme dans tout type de serveur.



Faibles performances

Serveur WEB

Réseau

Nombre de sessions ouvertes

```
tadoussac# netstat -s
TCP          tcpRtoAlgorithm      =      4 tcpRtoMin          =    200
              tcpRtoMax            = 60000 tcpMaxConn              =     -1
              tcpActiveOpens      =    102 tcpPassiveOpens        =    154
              tcpCurrEstab       =         7
```

Syn Attack

```
tadoussac# netstat -s
. tcpListenDrop      =      0 tcpListenDropQ0        =      0
  tcpHalfOpenDrop    =      0
```

Fin de connexion

```
tadoussac# netstat -a | grep TIME
tadoussac.login      vancouver.1023  8760      0 8760      0 TIME_WAIT
tadoussac#
```

Faibles performances

Serveur WEB

Réseau

Les ressources réseau sont fondamentales sur ce type de serveur. La connexion doit être libérée au plus vite et la surveillance des SYN Attacks est impérative.

Ici, le temps de réponse obtenu n'est pas forcément compatible avec un lien LAN. Ainsi, un processus peut être longtemps mobilisé par un seul client. Le temps de réponse va donc être lié au nombre de processus qui auront été générés.

Il est conseillé de demander aux clients de plutôt effectuer des requêtes `ftp` pour transférer des données très importantes (même problème que NFS).

Notes

Objectifs

Les sujets couverts par ce chapitre seront les suivants :

- action sur le noyau,
- action sur les processus,
- action sur les disques,
- action sur le réseau,
- action sur les utilisateurs.



Introduction

Visualiser les valeurs des variables

Modifier les valeurs

Vérifier la modification

Introduction

Maintenant que nous savons où intervenir, nous allons décrire les outils nous permettant de mettre en oeuvre les modifications que nous avons choisies pour le système.

L'algorithme sera le suivant :

- Visualiser les valeurs des variables

nous allons décrire les commandes nous permettant de prendre connaissance des valeurs actuelles des variables que nous cherchons à modifier,

- Modifier les valeurs

nous allons expliquer les méthodes nous permettant (à coup sûr) de modifier la valeur des variables choisies,

- Vérifier la modification

comme nous l'avons vu précédemment, il est important de continuer la surveillance pour savoir si la modification est efficace...



Action sur le noyau

Visualiser les valeurs des variables

sysdef -i

adb

crash

Modifier la valeur

modification dans le fichier /etc/system

reboot de la machine

Vérifier la modification

Action sur le noyau

Visualiser les valeurs des variables

Dans le cas des variables noyau, les principales commandes d'investigation que nous allons vous proposer sont :

- `sysdef -i`,
- `adb`,
- `crash`.

Modifier la valeur

La modification des valeurs aura lieu via le fichier `/etc/system`, puis nous redémarrerons la machine.

Nous ne vous proposons pas de changer les variables en dynamique (via `adb`, par exemple), car :

- la manipulation peut toujours finir par un reboot,
- certaines variables ne sont prises en compte qu'au reboot de la machine.

Vérifier la modification

La phase de surveillance est reprise.



Action sur le noyau

Visualiser les valeurs des variables

La commande nm

```
tadoussac# /usr/ccs/bin/nm /platform/sun4u/kernel/unix | \
grep OBJ | grep GLO > /tmp/var
tadoussac#
tadoussac# vi /tmp/var
...
[1174] |      0 |      0 | OBJT | GLOB | 0 | UNDEF | maxmem
[1579] | 272646836 | 4 | OBJT | GLOB | 0 | 10 | maxphys
[1495] | 272664940 | 4 | OBJT | GLOB | 0 | 10 | maxsepgcnt
[1479] | 272706588 | 4 | OBJT | GLOB | 0 | 13 | maxswinum
[1994] | 272712024 | 4 | OBJT | GLOB | 0 | 13 | maxuprc
[2615] | 272723024 | 4 | OBJT | GLOB | 0 | 13 | maxusers
...
```

Action sur le noyau

Visualiser les valeurs des variables

La commande nm

Nous vous avons fourni un certain nombre de variables noyau. Ces dernières sont disponibles dans l'AnswerBook Administrateur, ou dans le support des Internals.

Il se peut que l'orthographe des variables évolue au cours des releases. Il est possible de s'assurer du nom d'une variable via la commande nm.

Les variables sont des OBJETS dits GLOBAUX se trouvant dans les fichiers qui constituent le noyau (`/kernel/genunix`, `/platform/sun4xxx/kernel/unix...`, ou dans les modules présents dans `/kernel`, `/usr/kernel` et `/platform/sun4xxx/kernel`).



Action sur le noyau

Visualiser les valeurs des variables

La commande sysdef

```
tadoussac# sysdef -i
*
* Process Resource Limit Tunables (Current:Maximum)
*
Infinity:Infinity      cpu time
Infinity:Infinity      file size
7ffff000:7ffff000      heap size
800000:7ffff000        stack size
Infinity:Infinity      core file size
40: 400                 file descriptors
Infinity:Infinity      mapped memory
*
...
*
* IPC Semaphores
*
10                      entries in semaphore map (SEMMAP)
70                      semaphore identifiers (SEMMNI)
200                     semaphores in system (SEMMNS)
30                      undo structures in system (SEMMNU)
25                      max semaphores per id (SEMMSL)
10                      max operations per semop call (SEMOPM)
10                      max undo entries per process (SEMUME)
32767                   semaphore maximum value (SEVMVMX)
16384                   adjust on exit max value (SEMAEM)
*
* IPC Shared Memory
*
83886008               max shared memory segment size (SHMMAX)
1                      min shared memory segment size (SHMMIN)
100                   shared memory identifiers (SHMMNI)
10                   max attached shm segments per process
                      (SHMSEG)
...

```

Action sur le noyau

Visualiser les valeurs des variables

La commande sysdef

Cette commande permet d'interroger un certain nombre de ressources pré-chargées dans le noyau (forceload). Il s'agit essentiellement des :

- limites liées aux utilisateurs,
- IPC,
- streams,
- tables de scheduling.



Action sur le noyau

Visualiser les valeurs des variables

La commande adb

```
tadoussac# adb -k /dev/ksyms /dev/mem
physmem          eb8
maxusers/D
maxusers:
maxusers:        29
maxuprc/D
maxuprc:
maxuprc:         469
ufs_ninode/D
ufs_ninode:
ufs_ninode:      583
ncsize/D
ncsize:
ncsize:          583

tadoussac#
```

Action sur le noyau

Visualiser les valeurs des variables

La commande adb

La commande `adb` permet d'interroger (voire de modifier) tout code chargé en mémoire, dans ce cas précis nous interrogeons le noyau (dont l'image s'appelle `/dev/ksyms`). Il est ainsi possible de retrouver la valeur de toutes les variables fournies par la commande `nm`.

Il est possible d'obtenir une sortie en hexadécimal en spécifiant un `x` à la place du `D`.



Action sur le noyau

Visualiser les valeurs des variables

La commande crash

```
tadoussac# crash
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> kmastat
```

cache name	buf size	buf avail	buf total	memory in use	#allocations succeed fail	
-----	-----	-----	-----	-----	-----	-----
kmem_magazine_1	8	1008	1020	8192	1661	0
kmem_magazine_3	16	485	510	8192	7438	0
kmem_magazine_7	32	221	255	8192	9299	0
kmem_magazine_15	64	233	254	16384	5620	0
kmem_magazine_31	128	0	0	0	0	0
kmem_magazine_47	192	0	0	0	0	0
kmem_magazine_63	256	0	0	0	0	0
kmem_magazine_95	384	0	0	0	0	0
kmem_magazine_143	576	0	0	0	0	0
kmem_slab_cache	32	85	255	8192	7732	0
kmem_bufctl_cache	12	657	1020	16384	25935	0
kmem_alloc_8	8	394	3060	24576	369498	0
kmem_alloc_16	16	135	2550	40960	106377	0
kmem_alloc_24	24	493	1020	24576	89373	0
kmem_alloc_32	32	43	765	24576	16456	0
kmem_alloc_40	40	120	408	16384	96834	0
kmem_alloc_48	48	72	170	8192	49628	0
kmem_alloc_56	56	45	1305	73728	5101	0
kmem_alloc_64	64	9	254	16384	58779	0

Action sur le noyau

Visualiser les valeurs des variables

La commande crash

Cette commande permet de visualiser toutes les tables du noyau, en cours de fonctionnement.



Action sur le noyau

Modifier la valeur

Modification dans le fichier /etc/system

Introduction

/kernel	{	/drv	Drivers
		/exec	exec() Modules
		/fs	File Systems
		/misc	Miscellaneous Modules
		/sched	Scheduling Classes
		/strmod	STREAMS Modules
		/sys	Loadable System Calls
		/genunix	Noyau unix

Action sur le noyau

Modifier la valeur

Modification dans le fichier `/etc/system`

Introduction

Un point important de *Solaris 2.x* est la modularité du noyau. Cette dernière permet de charger ou de décharger du logiciel noyau de façon modulaire, ce qui permet une gestion souple ainsi qu'une optimisation de l'occupation de la mémoire centrale.

Le noyau comporte un ensemble de modules de base qui constituent l'*Operating System* : `/kernel/unix`.

Les modules noyau tels que :

- appels système
- fonctions
- drivers
- ...

sont chargés soit à l'appel de ce module, soit au moment du boot (voir le fichier de configuration `/etc/system`).

Le mécanisme d'autoconfiguration

L'autoconfiguration est le mécanisme utilisé par le système pour valider de nouveaux drivers ou de nouvelles options logicielles. Ce mécanisme permet au système de se configurer avec un minimum d'intervention de la part de l'administrateur.

Les tables du noyau

Certaines tables et certaines variables du noyau sont modifiables par l'administrateur.



Action sur le noyau

Modifier la valeur

Modification dans le fichier `/etc/system`

Le fichier `/etc/system`

- Paramètres de configuration
 - répertoire des modules noyau
 - type et *device* des partitions `root` et `swap`
 - exclusion de modules
 - pré-chargement de modules
 - variables système

Action sur le noyau

Modifier le contenu de /etc/system

```

*ident      "@(#)system 1.1592/11/14 SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
* moddir:
*
*   Set the search path for modules.  This has a format similar to the
*   csh path variable.  If the module isn't found in the first directory
*   it tries the second and so on.  The default is /kernel /usr/kernel
*   Example:
*
*           moddir: /kernel /usr/kernel /other/modules
* root device and root filesystem configuration:
*   The following may be used to override the defaults provided by
*   the boot program:
*   rootfs:   Set the filesystem type of the root.
*   rootdev:  Set the root device.  This should be a fully
*             expanded physical pathname.  The default is the
*             physical pathname of the device where the boot
*             program resides.  The physical pathname is
*             highly platform and configuration dependent.
*   Example:
*
*           rootfs:ufs
*           rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
*   (Swap device configuration should be specified in /etc/vfstab.)
* exclude:
*   Modules appearing in the moddir path which are NOT to be loaded,
*   even if referenced.  Note that `exclude' accepts either a module name,
*   or a filename which includes the directory.
*   Examples:
*
*           exclude: win
*           exclude: sys/shmsys
* forceload:
*   Cause these modules to be loaded at boot time, (just before mounting
*   the root filesystem) rather than at first reference.  Note that
*   forceload expects a filename which includes the directory.  Also
*   note that loading a module does not necessarily imply that it will
*   be installed.
*   Example:
*
*           forceload: drv/foo
* set:
*   Set an integer variable in the kernel or a module to a new value.
*   This facility should be used with caution.  See system(4).
*
*   Examples:
*
*   To set variables in 'unix':
*
*           set nautopush=32
*           set maxusers=40

```



Action sur le noyau

Modifier la valeur

Modification dans le fichier /etc/system

Affectation d'une valeur à un paramètre

Ajouter une ligne dans le fichier /etc/system

```
set nom_variable=valeur
```

exemple : set max_nprocs=500

Rebooter le système

Affectation d'une valeur à une variable d'un module

Ajouter une ligne dans le fichier /etc/system

```
set nom_du_module:nom_variable=valeur
```

exemple : set msgsys:msginfo_msgmap=150

Rebooter le système

Modifier la valeur des paramètres

Affectation d'une valeur à un paramètre

Une fois le nom du paramètre trouvé, il suffit de changer sa valeur dans le fichier `/etc/system` et de rebooter pour que cette modification soit prise en compte (lire les messages au moment du reboot, voire interdire le `dtlogin` sur la console du serveur).

Affectation d'une valeur à une variable d'un module

Il est aussi possible de mettre à jour une variable dans un module spécifique, il suffit pour cela de préfixer le nom de la variable du nom du module.



Modifier la valeur des paramètres

Cas des IPC

File de messages

<code>msginfo_msgmap</code>	défaut	100
<code>msginfo_msgmax</code>	défaut	2048
<code>msginfo_msgmnb</code>	défaut	4096
<code>msginfo_msgmni</code>	défaut	50
<code>msginfo_msgssz</code>	défaut	8
<code>msginfo_msgtql</code>	défaut	40
<code>msginfo_msgseg</code>	défaut (<32768)	1024

■ Affectation des valeurs

```
set msgsys:msginfo_variable=valeur
```

Modifier la valeur des paramètres

Cas des IPC

File de messages

msginfo_msgmap

nombre maximum de file de messages allouables par le système.

msginfo_msgmax

taille maximum d'un message (octets).

msginfo_msgmnb

taille maximum de la file de messages (octets).

msginfo_msgmni

nombre d'identifiant utilisables pour une file de messages. (règle générale : taille de la tables de la file des messages / 2).

msginfo_msgssz

taille minimum d'un message (règle générale : multiple de la taille d'un mot).

msginfo_msgtql

nombre de messages headers dans le système.

msginfo_msgseg

nombre maximum de messages de taille minimale.



Modifier la valeur des paramètres

Cas des IPC

Sémaphores

<code>seminfo_semmap</code>	défaut	10
<code>seminfo_semmni</code>	défaut	10
<code>seminfo_semmns</code>	défaut	60
<code>seminfo_semmsl</code>	défaut	25
<code>seminfo_semopm</code>	défaut	10
<code>seminfo_semvmx</code>	défaut	32767

■ Affectation des valeurs

```
set semsys:seminfo_variable=valeur
```

Modifier la valeur des paramètres

Cas des IPC

Sémaphores

seminfo_semmap

nombre maximum de sémaphores dans une famille.

seminfo_semmni

nombre d'identifiants de sémaphores.

seminfo_semmns

nombre maximum de sémaphores allouable par le système.

seminfo_semmsl

nombre maximum de sémaphores par identifiants.

seminfo_semopm

nombre maximum de processus pouvant effectuer des opérations simultanées sur un sémaphore.

seminfo_semvmx

valeur maximum affectable à un sémaphore.



Modifier la valeur des paramètres

Cas des IPC

Mémoire partagée

<code>shminfo_shmmax</code>	défaut	131072
<code>shminfo_shmmin</code>	défaut	1
<code>shminfo_shmseg</code>	défaut	6

■ Affectation des valeurs

```
set shmsys:shminfo_variable=valeur
```

Modifier la valeur des paramètres

Cas des IPC

Mémoire partagée

`shminfo_shmmax`

taille maximum d'un segment de mémoire partagée.

`shminfo_shmmin`

taille minimum d'un segment de mémoire partagée.

`shminfo_shmseg`

nombre maximum de segments utilisables par processus.



Action sur les processus

Visualiser les valeurs des variables

ps -ef

proctool

debugger

Modifier la valeur

modification via prionctl

Vérifier la modification

Action sur les processus

Visualiser les valeurs des variables

la principale action possible sur un processus est de changer sa priorité, Pour cela nous allons commencer par visualiser la priorité d'un processus puis la modifier. Pour visualiser les priorités, les commandes sont :

- `ps -ef`,
- `proctool`,
- `debugger`.

Modifier la valeur

La modification aura lieu via la commande `prionctl`.

Vérifier la modification

La phase de surveillance est reprise.



Action sur les processus

Visualiser les valeurs des priorités

```
tadoussac# ps -ecf | grep nfs
  root   263     1   TS  58   Apr 19 ?           0:00 /usr/lib/nfs/rpc.pcnfsd
  root   126     1   TS  58   Apr 19 ?           0:00 /usr/lib/nfs/statd
  root   128     1   TS  58   Apr 19 ?           0:00 /usr/lib/nfs/lockd
  root  2019     1   IA  59 10:11:36 ?       0:00 /usr/lib/nfs/nfsd
                                     -a 1
  root  2021     1   IA  59 10:11:36 ?       0:00
                                     /usr/lib/nfs/mountd
  root  2112   2025   IA  59 14:39:56 pts/6    0:00 grep nfs
```

Action sur les processus

Visualiser les valeurs des priorités

Les priorités des processus peuvent être visualisées avec les commandes :

- `ps -ec`,
- `proctool`,
- ou un debugger.



Action sur les processus

Modifier la valeur

```
tadoussac# ps -ecf | grep nfs
  root   263     1    TS   58   Apr 19 ?           0:00 /usr/lib/nfs/rpc.pcnfsd
  root   126     1    TS   58   Apr 19 ?           0:00 /usr/lib/nfs/statd
  root   128     1    TS   58   Apr 19 ?           0:00 /usr/lib/nfs/lockd
  root  2019     1    IA   59 10:11:36 ?       0:00 /usr/lib/nfs/nfsd
                                     -a 1
  root  2021     1    IA   59 10:11:36 ?       0:00
                                     /usr/lib/nfs/mountd
  root  2112  2025    IA   59 14:39:56 pts/6    0:00 grep nfs

tadoussac# priocntl -s -c RT -i pid 2019

tadoussac# ps -ecf | grep nfs
  root   263     1    TS   58   Apr 19 ?           0:00 /usr/lib/nfs/rpc.pcnfsd
  root   126     1    TS   58   Apr 19 ?           0:00 /usr/lib/nfs/statd
  root   128     1    TS   58   Apr 19 ?           0:00 /usr/lib/nfs/lockd
  root  2019     1    RT  100 10:11:36 ?       0:00 /usr/lib/nfs/nfsd
                                     -a 1
  root  2021     1    IA   59 10:11:36 ?       0:00
                                     /usr/lib/nfs/mountd

tadoussac#
```

Action sur les processus

Modifier la valeur

La modification d'une priorité s'effectue via la commande `priocntl`.

Principales options

- l liste les classes de scheduling
- d visualise les paramètres du processus
- s positionne les paramètres

Il est aussi possible d'agir sur le processus via les commandes `pbind` (allocation d'un processus à un processeur), `psrset` (allocation d'un groupe de processus à un processeur), `psradm` (gestion d'un processeur).



Action sur les disques

Visualiser les valeurs des variables

iostat

fstyp

Modifier la valeur

implantation des niveaux de raid

tunefs

mkfs

Vérifier la modification

Action sur les disques

Visualiser les valeurs des variables

Dans les chapitres précédents, nous avons étudié des commandes permettant de visualiser les choix faits sur les disques :

- `iostat` : taux de transfert, et type de transfert,
- `fstyp` : paramétrage du système de fichiers.

Modifier la valeur

Les changements à opérer passent par l'utilisation de logiciels implémentants des niveaux de raid (SDS, VM, Raid matériel), et permettant de changer les paramètres du système de fichiers.

Vérifier la modification

La phase de surveillance est reprise.



Action sur les disques

Visualiser les valeurs des variables

fstyp

```

resa3# fstyp /dev/dsk/c0t1d0s0
ufs
resa3#

resa3# fstyp -v /dev/dsk/c0t1d0s0
ufs
magic          11954          time          Tue May 18 11:27:14 1993
sblkno         16           cblkno        24           iblkno        32           dblkno        184
sbsize         2048          cgsiz         1024          cgooffset     24cgmask0xffffffff0
ncg            18           size          46170          blocks        43129
bsize          8192          shift         13           mask          0xffffe000
fsize          1024          shift         10           mask          0xfffffc00
frag           8           shift         3           fsbtodb       1
minfree        10%          maxbpg        2048          optim         time
maxcontig      7           rotdelay      0ms          rps           60
csaddr         184          cssize        1024          shift         9           mask          0xfffffe00
ntrak          9           nsect         36           spc           324          ncyl          285
cpg            16           bpg           324          fpg           2592          ipg           1216
nindir         2048          inopb         64           nspf          2
nbfree         4170          ndir          288          nifree        20352          nffree        102
cgrotor        8           fmod          0           ronly         0
file system state is valid, fsclean is 0
blocks available in each rotational position
cylinder number 0:
  position 0:    0    7    9    16    18
  position 1:    5   14
  position 2:    3   12
  position 3:    1   10   19
  position 4:    8   17
  position 5:    6   15

```

Action sur les disques

Visualiser les valeurs des variables

fstyp

Le système de fichiers à examiner (pas les montages NFS) est passé en argument.

Principales options

<i>sans options</i>	affiche le type du système de fichiers
-v	affiche des informations sur le système de fichiers

Principales colonnes

Informations du superbloc et des cylindres.

magic	type du file system
ncg	nombre de cylindres par groupe
bsize	taille d'un bloc logique
fsize	taille d'un fragment
nbfree	nombre de bloc libres
mnfree	pourcentage minimum laissé libre
maxcontig	nombre de blocs maximum contigus pour un fichier ordinaire
rotdelay	temps d'attente entre chaque rotation
optim	type d'optimisation

Remarque : le flag `FSCLEAN` est visualisé.



Action sur les disques

Modifier la valeur

tunefs

- `tunefs [-a maxconfig][-d rotdelay]
[-e maxbpg][-m minfree][-o [s|t]]
special | filesystem`

- Le système de fichiers doit être démonté

- L'optimisation doit se faire avant que le taux d'occupation de la partition dépasse 90%

Action sur les disques

Modifier la valeur

tunefs

Principales options

- a *maxconfig* nombre maximum de blocs contigus qui seront lus en un seul accès, (1 par défaut)
- d *rotdelay* temps nécessaire pour servir une interruption de fin de transfert et pour initialiser un nouveau transfert sur le même disque
- e *maxbpg* nombre maximum de blocs utilisables par un même fichier sur un groupe de cylindres
- m *minfree* pourcentage de l'espace disque non accessible aux simples utilisateurs
- o [s|t] change la stratégie d'optimisation pour le système de fichiers (s pour une optimisation en taille et t pour une optimisation en temps)



Action sur les disques

Modifier la valeur

mkfs, newfs

```
tadoussac# mkfs -F ufs -o cgsize=200 free=2 nbpi=200000 rps=90 \  
/dev/r....
```

Action sur les disques

Modifier la valeur

La commande de création d'un système de fichiers est `mkfs` ou `newfs`.

Principales options

<code>-csize</code>	nombre de cylindres par groupe (doit être important dans la cas d'un fichier mis en oeuvre dans une base de données)
<code>-free</code>	pourcentage de place libre laissée sur le disque (2% pour un système de fichiers de taille supérieure à 1 G octets, pour les zones utilisateurs, peut être mis à zéro pour de la base de données)
<code>-nbpi</code>	nombre d'inodes par kilo octets présents sur le système de fichiers (2048 par défaut)
<code>-rps</code>	vitesse de rotation du disque (par seconde), la valeur par défaut est de 60

Ces options peuvent être passées à la commande `mkfs` ou `newfs`.



Action sur le réseau

Visualiser les valeurs des variables

ndd

Modifier la valeur

ndd

Vérifier la modification

Action sur le réseau

Visualiser les valeurs des variables

La commande permettant de visualiser les paramètres liés au réseau est `ndd`.

Modifier la valeur

La commande permettant de positionner les paramètres liés au réseau est `ndd`.

Elle est prise en compte sans reboot de la machine. L'administrateur veillera à positionner ses choix dans les fichiers de démarrage pour qu'ils soient permanents après un reboot (`/etc/init.d/inetinit`).

Vérifier la modification

La phase de surveillance est reprise.



Action sur le réseau

Modifier la valeur

ndd

```
tadoussac# ndd /dev/tcp
name to get/set ? ?
?
tcp_close_wait_interval (read and write)
tcp_conn_req_max (read and write)
tcp_conn_grace_period (read and write)
tcp_cwnd_max (read and write)
tcp_debug (read and write)
tcp_smallest_nonpriv_port (read and write)
tcp_ip_abort_cinterval (read and write)
tcp_ip_abort_interval (read and write)
tcp_ip_notify_cinterval (read and write)
tcp_ip_notify_interval (read and write)
tcp_ip_ttl (read and write)
...
name to get/set ? tcp_close_wait_interval
value ?
length ?
240000
name to get/set ? tcp_close_wait_interval
value ? 1000

name to get/set ? tcp_close_wait_interval
value ?
length ?
1000
name to get/set ?
```

Action sur le réseau

Modifier la valeur

ndd

La commande `ndd` permet de changer les paramètres des modules réseaux de Solaris 2.x.



Action sur les utilisateurs

Visualiser les valeurs des variables

limit, ulimit

accounting

Modifier la valeur

limit, ulimit

cron, at

Vérifier la modification

Action sur les utilisateurs

Visualiser les valeurs des variables

Les limites induites par les utilisateurs sont visibles par les commandes `limit` ou `ulimit` (en fonction de l'interpréteur de commande utilisé).

L'accounting permet aussi de connaître le comportement de l'utilisateur en terme de répartition de charges et de nombres de processus activés sur un serveur.

Modifier la valeur

Il est possible de modifier les valeurs limites de chaque utilisateur (ou processus) via les mêmes commandes `limit` ou `ulimit`. Les développeurs disposent des appels `getrlimit` et `setrlimit` pour changer les paramètres des applications. Il est aussi possible d'intervenir de façon globale sur le système (positionnement des valeurs noyau), mais ceci est fort peu recommandé.

Il est aussi possible de modifier le comportement des utilisateurs via des commandes de type `cron`, etc.

Vérifier la modification

La phase de surveillance est reprise.



Action sur les utilisateurs

Visualiser les valeurs des variables

```
tadoussac# ulimit
4194303
tadoussac# csh
tadoussac# limit
cputime                unlimited
filesize               unlimited
datasize               2097148 kbytes
stacksize              8192 kbytes
coredumpsize          unlimited
descriptors            64
memorysize             unlimited
tadoussac#
```

Action sur les utilisateurs

Visualiser les valeurs des variables

Les commande `limit` et `utlimit` permettent de visualiser les valeurs des limites induites par processus.

Principales options

<code>-c</code>	taille du fichier core (en secteur)
<code>-d</code>	taille du heap (en ko)
<code>-f</code>	taille du fichier (en secteur)
<code>-n</code>	nombre maximum de file descriptors
<code>-s</code>	taille de la stack (en ko)
<code>-t</code>	temps CPU maximum (en seconde)
<code>-v</code>	taille de la mémoire virtuelle (en ko)

Ces commandes permettent aussi de modifier la valeur de ces variables.



Action sur les utilisateurs

Modifier la valeur

limit, utlimit

cron, at

quota

maxnprc

/etc/inet/inetd.conf

IDLE_TIME

games

Action sur les utilisateurs

Modifier la valeur

Il est aussi possible de modifier le « comportement » de l'utilisateur. Les commandes permettant d'agir sur le comportement sont :

- `limit, utlimit` : limitation des ressources utilisées par connexion ou par application,
- `maxnprc` : nombre maximum de processus validés par utilisateur (environnement graphique),
- `/etc/inet/inetd.conf` : interdiction de se connecter sur un serveur,
- `cron, at` : lancement de travaux sur les temps peu utilisés de la machine,
- `quota` : limitation de l'espace utilisé sur un système de fichiers,
- `IDLE_TIME` : déconnexion automatique d'une base de données,
- `games` : ouverture de sessions ... intéressantes.



Notes

Objectifs

Les sujets couverts par ce chapitre seront les suivants :

- cas de la machine desktop,
- cas du serveur générique,
- cas du serveur de calcul,
- cas du serveur NFS,
- cas du serveur de base de données,
- cas du serveur WEB.



Introduction

Cas de la machine desktop

Cas du serveur générique

Cas du serveur de calcul

Cas du serveur NFS

Cas du serveur de base de données

Cas du serveur WEB

Introduction

Le but de ce chapitre est de proposer des études de cas sur des machines typiques ayant une fonction bien définie. Les cas traités recouvrent :

- la machine desktop : cas typique de la machine cliente utilisant des applications de compilation ou des applications graphiques,
- le serveur générique : nous traiterons du serveur de terminaux X, d'impression et de nom,
- le serveur de calcul : ici, nous ne recherchons que la puissance de calcul,
- le serveur NFS : ce serveur est avant tout un serveur d'espace disque. nous commencerons par traiter de ce sujet avant de mettre en oeuvre une amélioration des performances liées à NFS,
- le serveur de base de données : nous travaillerons exclusivement sur la partie base de données,
- le serveur WEB : nous travaillerons exclusivement sur la partie WEB.



Cas de la machine desktop

Description de la machine

Les choix liés au système d'exploitation

Les choix liés aux applicatifs

Cas de la machine desktop

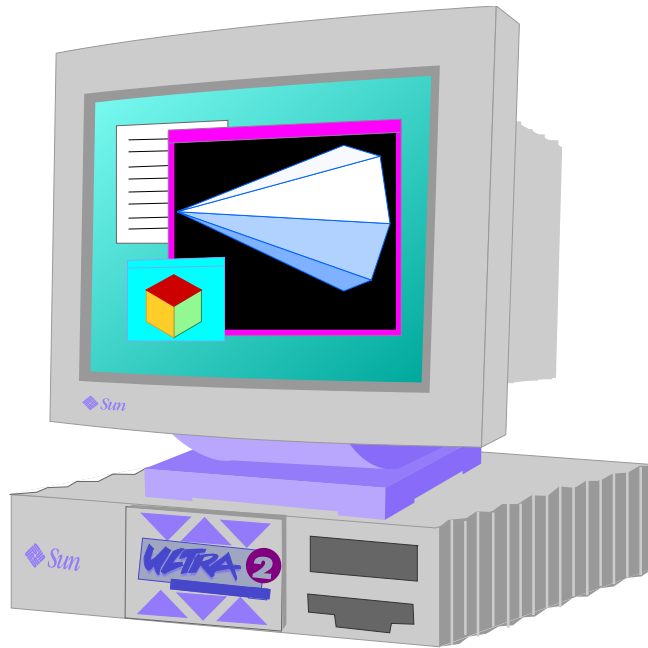
Les sujets que nous allons traiter recouvrent :

- la description de la machine : matériel disponible,
- les choix liés au système d'exploitation : les implémentations des services de base du système d'exploitation,
- les choix liés aux applicatifs : nous traiterons essentiellement le cas du service NFS.

Cas de la machine desktop

Description de la machine

Matériel : Ultra™ 2 and Ultra™ Enterprise™ 2



Disque interne de 2 G octets

Logiciel

Cluster developer

Cas de la machine desktop

Description de la machine

Matériel : Ultra™ 2 and Ultra™ Enterprise™ 2

Le matériel doit disposer de fortes capacités graphiques (donc il est nécessaire de surveiller la quantité de RAM présente) et un disque interne de 2 Giga octets est disponible.

Aucune entrée/sortie supplémentaire n'est nécessaire sur ce type de machine si cette dernière est présente sur un réseau (ce qui est fortement conseillé pour une machine cliente).

Logiciel

Le disque interne ne contient aucune donnée utilisateur (pour ne pas avoir à sauvegarder les disques de chaque machine), le système d'exploitation mobilise 500 à 600 M octets, le reste sera utilisé comme zone cache.

Le cluster developper est installé par défaut.



Cas de la machine desktop

Les choix liés au système d'exploitation

Zone de swap

/tmp

```
tadoussac# df -k
Filesystem          kbytes    used    avail capacity  Mounted on
/dev/dsk/c0t0d0s0   384847   295076    51291    86%      /
/proc                0         0         0         0%      /proc
fd                   0         0         0         0%      /dev/fd
/dev/dsk/c0t0d0s7   516678   446305    18713    96%      /export
swap                 12800     320    12480     3%      /tmp
/dev/dsk/c0t1d0s0   1855797  898837   771390    54%      /apps
montreal:/export    384847   295076    51291    86%      /mnt
```

Energy star

Les processus

Cas de la machine desktop

Les choix liés au système d'exploitation

Zone de swap

La zone de swap est locale à la machine (sur des postes disposant de moins d'espace disque, il est possible de mobiliser tout le disque pour disposer d'un swap local).

/tmp

le répertoire `/tmp` est utilise le système de fichier `tmpfs`. Ainsi, les développeurs peuvent disposer de bonnes performances lors des compilations (s'ils utilisent cette zone !).

Energy start

Si le produit est validé sur le poste client, nous veillerons à expliquer qu'il est nécessaire de se déconnecter de certaines applications lors de l'arrêt de la machine (mise sous le contrôle de l'utilisateur).

Les processus

Nous validerons un minimum de processus sur cette machine (pour la concentrer sur sa tâche principale), ainsi l'administrateur créera un fichier `/etc/defaultrouter` pour invalider le routage statique.

On pourra aussi invalider le processus `sendmail` et monter NFS la partition de mail (voire utiliser `mailtool` à la place de `dtmail`).

Il en est de même de tous les processus (`kerbd`, etc.) n'ayant aucune utilité sur le site administré.



Cas de la machine desktop

Les choix liés aux applicatifs

NFS

cacheFS

```
vancouver (root-sh) # dfshares tadoussac
RESOURCE                                SERVER ACCESS    TRANSPORT
tadoussac:/export/home/support         tadoussac -      -
tadoussac:/b                            tadoussac -      -
tadoussac:/usr                          tadoussac -      -
vancouver (root-sh) #
```

Acces sans cache

```
vancouver (root-sh) # iostat -xPn
                                extended device statistics
 r/s  w/s   kr/s  kw/s wait actv wsvc_t asvc_t  %w  %b device
 1.0  0.5   5.3   3.0  0.0  0.0   0.0   25.6   0   2 c0t2d0s0
 0.5  0.1   1.9   6.1  0.0  0.0   0.0   14.4   0   1 c0t2d0s1
 0.0  0.0   0.0   0.0  0.0  0.0   0.0    0.0   0   0 c0t2d0s2
 1.7  0.2  12.6   1.4  0.0  0.0   0.0   25.0   0   2 c0t2d0s5
 0.0  0.0   0.0   0.0  0.0  0.0   0.0   10.2   0   0 c0t2d0s6
 0.0  0.0   0.0   0.0  0.0  0.0   0.0    5.7   0   0 c0t2d0s7
 0.0  0.0   0.5   0.0  0.0  0.0   6.4   90.4   0   0
                                tadoussac:/usr
vancouver (root-sh) #
```

Cas de la machine desktop

Les choix liés aux applicatifs

NFS

Il est conseillé de surveiller via `nfsstat` les accès NFS et de modifier (en fonction des résultats) les ordres de montage (se référer au chapitre 4).

cacheFS

Dans le cas où la machine est cliente NFS d'une application (en lecture seule), nous déclarerons cette zone en cache FS.

Nous disposons de deux machines :

- `tadoussac` est serveur de `/usr` (applicatifs quelconques du système d'exploitation),
- `vancouver` est client NFS de `/usr` qu'il monte sur `/c`

La première manipulation consiste à travailler sans cache (nous ne modifions aucun paramètre du système d'exploitation, ni de NFS). La seconde manipulation consiste à monter la même ressource mais avec cacheFS.



Cas de la machine desktop

Les choix liés aux applicatifs

cacheFS

Acces avec cache

```
vancouver (root-sh) # cfsadmin -c /ORACLE/mon_cache
vancouver (root-sh) #

vancouver (root-sh) # mount -F cachefs -o
backfstype=nfs,ro,cachedir=/ORACLE/mon_cache tadoussac:/usr /c
vancouver (root-sh) #

vancouver (root-sh) # iostat -xPn
                                extended device statistics
  r/s  w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
  0.9  0.5   4.7    2.8  0.0  0.0   0.0   27.5  0   1 c0t2d0s0
  0.6  0.1   2.5    9.1  0.0  0.0   0.0   14.3  0   1 c0t2d0s1
  0.0  0.0   0.0    0.0  0.0  0.0   0.0    0.0  0   0 c0t2d0s2
  1.5  0.2  11.1    1.4  0.0  0.0   0.0   25.6  0   2 c0t2d0s5
  0.1  0.3   9.8   19.8  0.0  0.0   0.0   46.3  0   1 c0t2d0s6
  0.0  0.0   0.0    0.0  0.0  0.0   0.0    5.7  0   0 c0t2d0s7
  0.1  0.0   1.3    0.0  0.0  0.0   5.4   24.1  0   0
                                                tadoussac:/usr

vancouver (root-sh) #
```

Cas de la machine desktop

Les choix liés aux applicatifs

cacheFS

Durant cette manipulation, nous n'avons pas pris en compte :

- l'absence de trame NFS lors de l'utilisation de l'application par `vancouver` (charge réseau),
- la libération du point de connexion (NFS version 3) sur le serveur `tadoussac`,
- le fait que nous utilisons une partie d'un système de fichiers prévu pour une toute autre occupation.

Nous vous rappelons que ces ordres de montage peuvent être mis dans le fichier `/etc/vfstab` du client ou dans une table d'automount (voir le cas du serveur).



Cas du serveur générique

Description de la machine

Les choix liés au système d'exploitation

Les choix liés aux applicatifs

Cas du serveur générique

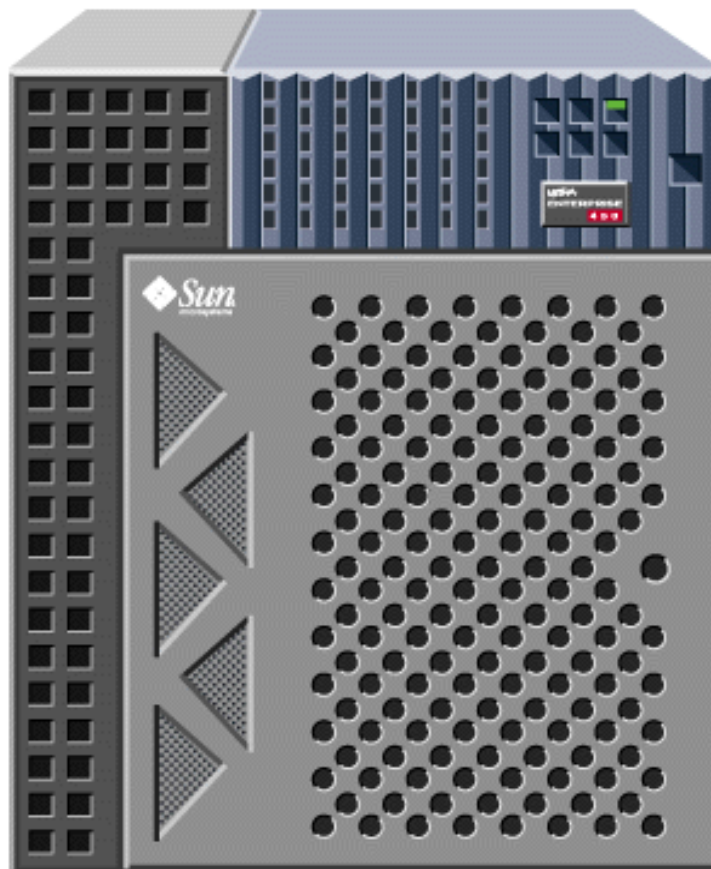
Les sujets que nous allons traiter recouvrent :

- la description de la machine : matériel disponible,
- les choix liés au système d'exploitation : les implémentations des services de base du système d'exploitation,
- les choix liés aux applicatifs : nous traiterons essentiellement le cas du service NISplus, terminal X, impression.

Cas du serveur générique

Description de la machine

Ultra™ Enterprise™ 450



Cas du serveur générique

Description de la machine

Matériel : Ultra™ Enterprise™ 450

Le matériel doit disposer de fortes capacités disques, puisque son but est d'être le plus polyvalent possible sur les services de « confort ». S'il doit vraiment être polyvalent un E 10000 répondra sûrement mieux aux attentes des utilisateurs (cette remarque sera valable pour tous les autres cas étudiés après).

Logiciel

Le cluster all est installé par défaut.



Cas du serveur générique

Les choix liés au système d'exploitation

/tmp

/var

swap

Cas du serveur générique

Les choix liés au système d'exploitation

/tmp

Le contenu de ce répertoire peut devenir important, il peut alors être nécessaire de créer une partition spécifique prévue à cet effet.

/var

Ce répertoire contient tous les spools, il peut être préférable de disposer d'une partition `/var/spool` dédiée pour ne pas gêner le comportement du système d'exploitation, lorsque cette partition est pleine.

swap

Ce type de serveur va proposer une grande quantité de connexions, il est donc nécessaire de surveiller cette zone au vue de la quantité de processus validés sur la machine.



Cas du serveur générique

Les choix liés aux applicatifs

Serveur de terminal X

Serveur d'impressions

Serveur de noms

Cas du serveur générique

Les choix liés aux applicatifs

Serveur de terminal X

Ce type de service demande beaucoup de potentialités de terminaux virtuels et beaucoup de connexions. Il est donc nécessaire de reprogrammer la variable noyau suivante :

- `pn_cnt` : nombre de pseudo-terminaux (il est nécessaire de rebooter avec l'option `-r` pour prendre en compte cette modification).

Les connexions des utilisateurs vont nécessiter de gérer un grand nombre de connexions (voir les modifications réseaux) et un grand nombre de processus (`max_nprocs`, `ufs_ninode`, etc.)

Serveur d'impressions

Cette ressource nécessite beaucoup de place disque sous `/var` (partition à part, comme dans le cas du serveur de mail), et demande une surveillance accrue des processus d'impression (processus fuyant) et de `/tmp`.

Serveur de noms

Si cette machine sert de serveur de nom (NISplus), nous veillerons à suivre l'activité des processus, voire à lui ajouter de la mémoire supplémentaire pour supporter ce service.

De même, il est conseillé d'utiliser les commandes spécifiques de l'application (`nismatch`), plutôt que des commandes détournées (`niscat | grep xx`) qui induisent des surcharges de processus et de transfert réseau. Nous vous rappelons qu'il n'est pas nécessaire d'être connecté au serveur pour administrer ce service de nom.

La validation de `nscd` est fortement conseillée sur les machines clientes.



Cas du serveur de calcul

Description de la machine

Les choix liés au système d'exploitation

Les choix liés aux applicatifs

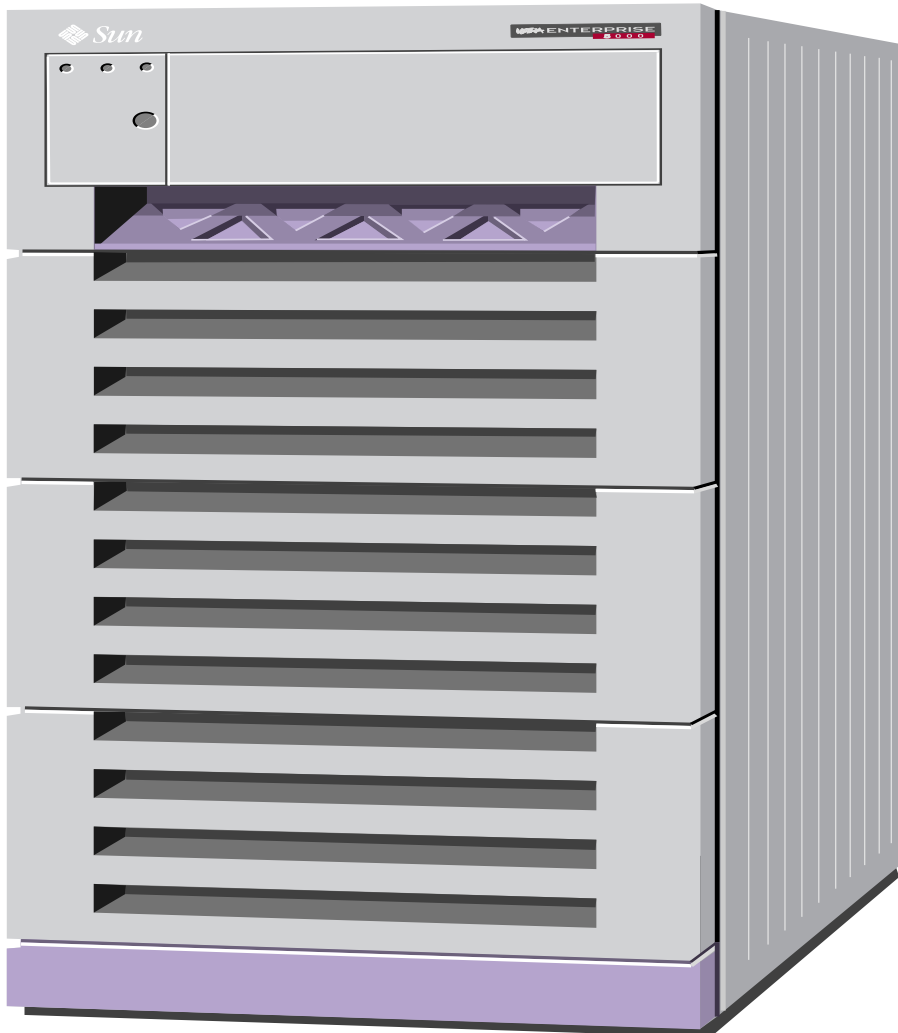
Cas du serveur de calcul

Les sujets que nous allons traiter recouvrent :

- la description de la machine : matériel disponible,
- les choix liés au système d'exploitation : les implémentations des services de base du système d'exploitation,
- les choix liés aux applicatifs : nous traiterons essentiellement le cas du service de temps CPU.

Cas du serveur de calcul

Description de la machine : Ultra™ Enterprise™ 5000



Cas du serveur de calcul

Description de la machine

Le matériel doit disposer de fortes capacités CPU (donc il est nécessaire de surveiller la quantité CPU pouvant être disponible).

Il est important de disposer de grandes quantités de disques et de mémoires centrales.

Ici, les zones caches vont prendre toutes leurs importances, on veillera à équilibrer les cartes disposant de CPU (autant de CPU, cache et mémoires par cartes systèmes).



Cas du serveur de calcul

Les choix liés au système d'exploitation

/tmp

/etc/norouter

Cas du serveur de calcul

Les choix liés au système d'exploitation

/tmp

Les calculs demandent souvent beaucoup d'espace sous /tmp. Si il est préférable de conserver la mémoire à l'usage du calcul, /tmp sera sauvegarder sur une partition classique d'un disque.

Si les fichiers « résultat » sont de taille importante, il sera alors nécessaire d'optimiser l'utilisation des disques (voir le paragraphe traitant des ressources NFS).

/etc/norouter

Il est conseillé d'invalider toute fonction annexe sur ce type de serveur, ainsi les fonctionnalités graphiques et de routage seront invalidées.



Cas du serveur de calcul

Les choix liés aux applicatifs

priocntl

```
montreal (sh) # ps -ec | grep essai
  488   IA  48 pts/5   0:00 essai
montreal (sh) #

montreal (sh) # timex essai

real      12.54
user      0.25
sys       4.48

montreal (sh) # timex priocntl -e -c RT -p 59 essai

real      8.67
user      0.25
sys       4.56
```

Cas du serveur de calcul

Les choix liés aux applicatifs

Les processus longs et non interactifs sont pénalisés par Unix. Pour privilégier leurs exécutions, il est alors nécessaire de leur changer leur priorité.



Cas du serveur NFS

Description de la machine

Les choix liés au système d'exploitation

Les choix liés aux applicatifs

Cas du serveur NFS

Les sujets que nous allons traiter recouvrent :

- la description de la machine : matériel disponible,
- les choix liés au système d'exploitation : les implémentations des services de base du système d'exploitation (gestion de l'espace disque),
- les choix liés aux applicatifs : nous traiterons essentiellement le cas du service NFS.

Cas du serveur NFS

Description de la machine : Ultra™ Enterprise™ 4000



Logiciel

Cas du serveur NFS

Description de la machine

Cette machine doit disposer d'un espace disque confortable, ainsi que d'un grand nombre de contrôleurs SCSI (ou liés aux périphériques disques utilisés).

Logiciel

L'installation initiale de la machine consistera à initialiser un disque en tant que disque système.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des processus

Gestion des ressources noyau

Gestion des disques : niveau de raid

Gestion des systèmes de fichiers

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des processus

Les processus importants sur cette machine sont ceux liés à NFS, donc toute connexion doit être exceptionnelle, et elle ne doit provenir que des administrateurs.

Gestion des ressources noyau

Ces serveurs sont sollicités énormément les buffers internes liés aux entrées/sorties, il est donc conseillé de modifier certaines ressources internes.

Gestion des disques : niveau de raid

Pour gérer au mieux les espaces disques, il est nécessaire d'utiliser les niveaux de raid fournis avec le système d'exploitation ou avec les périphériques.

Gestion des systèmes de fichiers

Une fois le niveau de raid choisi, il importe de prendre en compte le système de fichiers à installer pour gérer les partitions.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des processus

Pour chaque serveur	Nombre de nfsd
1 CPU	64
1 interface réseau 10 Mb/s	16
1 interface réseau 100 Mb/s	160
1 client actif	2

Temps CPU	Signification
SYSTEM > 60 %	aucun problème
SYSTEM < 60 %	trop de temps passé en mode USER (supprimer des applications)

Gestion des ressources noyau

ufs_ninode

ncsize

Surveillance de fsflush

Temps CPU	Signification
fsflush < 5 %	aucun problème
fsflush > 5 %	limiter le processus

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des processus

Le serveur NFS ne demande que peu de processus.

Ces processus ne nécessitent pas beaucoup de zone de swap. ils sont validés en priorité TS, il est donc important qu'aucune autre application ne prenne une priorité plus élevée.

Gestion des ressources noyau

Le noyau va être soumis à forte contribution concernant les caches des inodes et les DNLC. Il est donc conseillé de modifier les paramètres par défaut utilisés sur les machines.

Surveillance de fsflush

Il convient de surveiller ce processus, si le temps CPU qu'il utilise est supérieur à 5 %. Il est conseillé de modifier les paramètres qui lui sont liés.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Lecture séquentielle

Type de disque	100 k	1 M	10 M	100 M
normal	1,06	1,08	7,83	25,03
strip	1,07	0,75	3,13	19,50
miroir	0,84	1,12	5,86	39,20
raid 5	1,05	1,08	6,36	40,89

Ecriture séquentielle

Type de disque	100 k	1 M	10 M	100 M
normal	0,85	21,04	156,65	1248,28
strip	0,82	10,45	120,34	634,10
miroir	0,97	20,71	109,98	706,18
raid 5	1,03	26,88	252,61	2030,70

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

La première question à se poser est de savoir s'il est nécessaire d'un stripping, un miroir ou un raid 5, par rapport à un disque normal.

Puis, nous verrons l'impact du nombre de disques sur un stripping, l'impact de la taille du strip, et la différence entre un raid 5 avec ou sans log (ou un miroir avec ou sans log).

Résultat

Les résultats énoncés sont dûs à 10 processus travaillant en parallèle sur une machine. Les temps fournis sont ceux résultant d'une exécution où le temps `real` a été calculé.

Comme nous agissons sur une plate-forme implémentant du raid logiciel (SDS), les meilleurs temps système sont trouvés pour le disque dit « normal » (les écarts des temps système sont d'environ 10% entre un disque normal et un stripping ou un miroir, ou un raid 5).



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Lecture aléatoire

Type de disque	100 k	1 M	10 M	100 M
normal	1,52	2,14	3,11	5,46
strip	1,16	1,23	2,01	3,01
miroir	1,13	1,44	2,89	5,62
raid 5	1,00	1,16	3,79	6.75

Ecriture aléatoire

Type de disque	100 k	1 M	10 M	100 M
normal	0,97	2,18	3,38	11,24
strip	0,6	2,16	3,40	8,01
miroir	1,01	2,57	3,23	10,33
raid 5	0,96	4,10	15,20	22,62

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

D'après les résultats, les meilleures performances sont au compte du stripping, sachant que le raid 5 est fortement recommandé sur des accès en lecture seule.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Impact du log sur le miroir et le raid 5

Lecture

Type de disque	100 k	1 M	10 M	100 M
miroir sans log	1.11	2.07	4.42	7.05
miroir avec log	1.13	1.44	4.26	7.58
raid 5 sans log	1.06	1.16	3.79	5.20
raid 5 avec log	1.00	1.08	3.65	6.75

Ecriture

Type de disque	100 k	1 M	10 M	100 M
miroir sans log	1.01	2.57	9.48	11.24
miroir avec log	1.04	3.27	8.23	9.33
raid 5 sans log	0.96	4.51	15.20	19.64
raid 5 avec log	0.92	4.10	12.93	22.62

Cas de Volume Manager

Type de disque	100 k	1 M	10 M	100 M
raid 5 sans log	0.89	1.79	7.88	11.03
raid 5 avec log	0.89	1.14	7.15	13.06

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Impact du log sur le miroir et le raid 5

L'impact du log SDS et de la DRL de Volume manager est contrebalancé par le gain dû à l'absence de `fsck` qui aura lieu lors d'un arrêt brutal de la machine.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Impact du nombre de disques pour le stripping

Type de disque	100 k	1 M	10 M	100 M
strip 2 disques (ls)	0.90	1.17	5.81	31.06
strip 5 disques(ls)	0.90	1.11	4.71	26.45
strip 2 disques (la)	1.07	1.35	3,15	6,54
strip 5 disques(la)	1.06	1.10	2,45	4.22
strip 2 disques (es)	0.94	10.06	50.42	410.50
strip 5 disques(es)	1.01	7.50	36.17	331.85
strip 2 disques (ea)	1.01	1.80	2,92	4.10
strip 5 disques (ea)	1.08	1.40	1,88	2.46

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Impact du nombre de disques pour le stripping

Plus les transferts sont importants, plus le nombre de disques mis en oeuvre est intéressant dans une configuration strippée.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Impact de la taille du strip

Lecture séquentielle (sr)

Type de disque	100 k	1 M	10 M	100 M
strip 8k	0.88	1.09	5.39	27.91
strip 16k	0.90	1.17	5.37	26.43
strip 32 k	0.89	1.07	4.77	25.63
strip 128 k	0.90	1.11	4.10	22.45
strip 256 k	0.88	1.16	5.37	24.51

Ecriture séquentielle (sw)

Type de disque	100 k	1 M	10 M	100 M
strip 8k	0.90	8.21	53.71	404.11
strip 16k	0.90	6.27	41.23	340.10
strip 32 k	0.94	6.27	40.04	333.39
strip 128 k	1.01	6.50	36.17	331.85
strip 256 k	0.91	6.71	41.43	348.40

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

Pour les accès séquentiels, il est intéressant de dimensionner la taille du strip en fonction de la taille du transfert qui est prévu.

Les transferts aléatoires ne proposent pas de résultats significatifs.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

Natif

Les paramètres utilisés lors de la création

Les ordres de montage

VXFS

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

L'administrateur dispose de plusieurs techniques pour améliorer les performances du système de fichiers :

- utiliser le système de fichiers natif, en reprenant les paramètres de la commande `mkfs` comme nous lui avons indiqué dans le paragraphe précédent (nous vous rappelons que l'installation consiste à n'installer que le disque système),
- utiliser ce système de fichiers (en modifiant les paramètres de la commande `mkfs`) et en adaptant la commande de montage,
- utiliser un système de fichiers VXFS.



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

Natif

Les paramètres utilisés lors de la création

```
tadoussac# mkfs -F ufs -o cgsiz=200 free=2 nbpi=200000 rps=90 \
/dev/r....
```

Les ordres de montage

Type de disque	100 k	1 M	10 M	100 M
lecture N fichiers	0.062	0.046	0.418	5.31
lecture N fichiers (direct)	0.029	0.032	0.376	5.174
écriture N fichiers	0.09	0.032	0.554	6.706
écriture N fichiers (direct)	0.098	0.052	0.56	6.13
lecture 1 fichier	0.014	0.026	0.346	2.75
lecture 1 fichier (direct)	0.01	0.014	0.252	2.938
écriture 1 fichier	0.014	0.042	0.314	3.238
écriture 1 fichier (direct)	0.012	0.02	0.144	1.342

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

Natif

Les paramètres utilisés lors de la création

Les paramètres doivent s'adapter à la grandeur du système de fichiers, la commande `mkfs` vous est fournie au chapitre précédent.

Les ordres de montage

Solaris 2.6 dispose d'un système de fichiers modifié par rapport à Solaris 2.5, outre le changement de la taille maximum des fichier, il propose aussi une modification sur les ordres de montage. Il est possible de spécifier un paramètre :

- `forcedirectio` : les informations sont directement stockées dans la zone utilisateur (`mmap` intégré),
- `noforcedirectio` : les informations sont stockées dans la zone noyau, puis recopiées dans la zone utilisateur.

Par défaut, l'option utilisée est `noforcedirectio`.

Résultat

Nous utiliserons cette option pour les serveur de « data » qui proposent des fichiers de taille importante (ce type de montage s'adapte aussi très bien à un environnement base de données et permet de gagner des performances sur le système de fichiers sans implanter les raw devices).



Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

VXFS

Type de disque	100 k	1 M	10 M	100 M
lecture N fichiers	0.252	0.836	6.442	167.838
lecture N fichiers (VXFS)	0.336	1,472	11.368	159.394
écriture N fichiers	1.994	1.114	14.828	218.848
écriture N fichiers (VXFS)	1.53	0.618	13.116	195.704
lecture 1 fichier	0.128	0.974	3.214	31.682
lecture 1 fichier (VXFS)	0.07	0.218	2.904	28.5543
écriture 1 fichier	0.178	0.542	3.742	40.724
écriture 1 fichier (VXFS)	0.094	0.42	2.778	28.4256

Cas du serveur NFS

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

VXFS

Il est aussi possible d'utiliser un autre système de fichiers : VXFS. Ce dernier propose des facilités d'administrations non négligeables dans certains environnements.

Résultat

Ce système de fichiers est bien adapté pour les manipulations de fichiers de tailles importantes, sa justification est moins probante sur des fichiers dont les tailles ne dépassent pas les 10 M octets.



Cas du serveur NFS

Les choix liés aux applicatifs

Gestion du réseau

- tcp_close_wait_interval
- tcp_conn_req_max
- tcp_xmit_hiwat
- tcp_recv_hiwat

```
vancouver (root-sh) # iostat -xPn
                                extended device statistics
  r/s  w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
  0.0  0.0    0.0    0.0  0.0  0.0   0.0   5.7   0   0 c0t2d0s7
  0.0  7.4    0.0  146.3  0.2  0.6  28.9  78.2   3  33
tadoussac:/export/home/support/TEST
vancouver (root-sh) #
vancouver (root-sh) # iostat -xPn
                                extended device statistics
  r/s  w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
  0.0  0.0    0.1    0.0  0.0  0.0   0.0   5.7   0   0 c0t2d0s7
  0.0 10.8    0.0  214.4  0.3  0.8  29.8  77.7   4  47
tadoussac:/export/home/support/TEST
vancouver (root-sh) #
```

Gestion de NFS

Pour chaque serveur	Nombre de nfsd
1 CPU	64
1 interface réseau 10 Mb/s	16
1 interface réseau 100 Mb/s	160
1 client actif	2

Cas du serveur NFS

Les choix liés aux applicatifs

Gestion du réseau

Comme nous l'avons vu dans le chapitre précédent, certaines variables de TCP peuvent être positionnées :

- `tcp_close_wait_interval,`
- `tcp_conn_req_max,`
- `tcp_xmit_hiwat,`
- `tcp_recv_hiwat.`

Il est nécessaire de surveiller les connexions sur ce type de machine ainsi que les logins générant des processus plus prioritaires que les `nfsd`.

Gestion de NFS

La surveillance préconisée au chapitre 4 doit être mise en place. Dans la mesure du possible, il est préférable d'utiliser du `cacheFS`. Il en est de même avec `automount`. Cette méthode permet de ne pas garder de connexion établie en permanence sur le serveur. Elle doit se compléter d'une politique stricte de gestion des `PATH` et des processus sur le poste client (pour que le démontage puisse avoir lieu, et que le serveur ne soit pas trop sollicité).



Cas du serveur de base de données

Description de la machine

Les choix liés au système d'exploitation

Les choix liés aux applicatifs

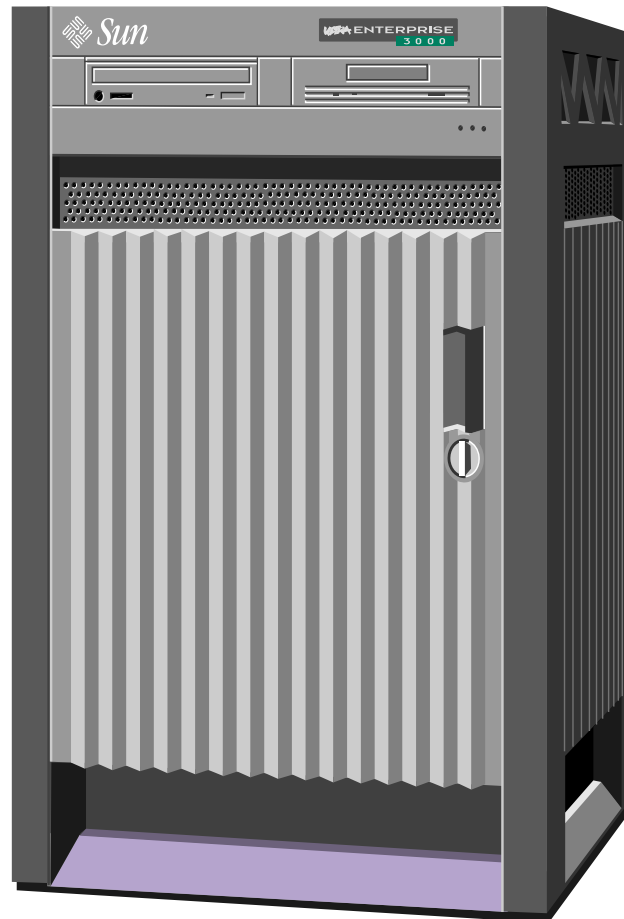
Cas du serveur de base de données

Les sujets que nous allons traiter recouvrent :

- la description de la machine : matériel disponible,
- les choix liés au système d'exploitation : les implémentations des services de base du système d'exploitation,
- les choix liés aux applicatifs : nous traiterons essentiellement le cas du service de base de données.

Cas du serveur de base de données

Description de la machine : Ultra™ Enterprise™ 3000



Logiciel

Cas du serveur de base de données

Description de la machine

Le matériel doit disposer de fortes capacités disques, et de fortes capacités RAM. Les SGBD utilisent énormément de mémoire pour travailler à une vitesse optimum.

Cette machine dispose souvent d'une redondance de périphériques pour qu'elle puisse continuer son exploitation sur une défaillance de disque. Nous disposerons donc de logiciels RAID.

Logiciel

Le disque système sera installé avec le cluster All. Le premier logiciel supplémentaire à installer est celui du SGBD.



Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des ressources noyau

Exploitation

`/tmp`

`/var`

Utilisateurs

`swap`

Gestion des processus

Gestion des disques : niveau de raid

Gestion des systèmes de fichiers

Gestion du réseau

Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des ressources noyau

Cette machine va solliciter les IPC (essentiellement la share memory), il est donc essentiel de surveiller l'utilisation de la mémoire centrale.

■ Exploitation

Les serveurs de bases de données rebootent rarement. Il est donc conseiller de forcer dans la crontab le nettoyage des fichiers messages, voire de vérifier la cohérence des systèmes de fichiers.

■ /tmp

Cette machine devrait mobiliser toute sa RAM pour la base de données (si du swapping est enregistré), il est conseillé de créer une partition spécifique pour /tmp.

■ Utilisateurs

En fonction du choix d'implémentation faite, il peut être demandé de devoir gérer beaucoup d'utilisateurs et beaucoup de sessions. Il peut être alors nécessaire de reprogrammer ces limites (voir le cas du serveur générique).

■ swap

Comme le mécanisme de swapping est perturbant, il sera particulièrement surveillé et l'administrateur pourra modifier les paramètres de type `lostfree`, `maxpgio`, `slowscan`, etc. pour limiter au plus les implications de ce mécanisme sur la base de données (et il veillera ... à ajouter de la RAM).



Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des processus

Les processus système

Les processus liés à la base de données

Temps CPU	Signification
USER > 60 %	aucun problème
USER < 60 %	trop de temps passé en mode SYSTEME (réorganisation)

Gestion des disques : niveau de raid

Gestion des systèmes de fichiers

Gestion du réseau

Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des processus

■ Les processus système

Comme dans le cas du serveur desktop, les processus non nécessaires seront supprimés. Si cette machine dispose de plusieurs interfaces de communications (et qu'elle n'a pas de fonctionnalité de routeur), l'administrateur créera un fichier `/etc/norouter (ip_forwarding 0)`.

Le processus `fsflush` sera particulièrement surveillé si la base de données est implantée sur du système de fichiers.

Pour ne pas perturber les processus de la base de données (qui sont validés par défaut en TS), les connexions graphiques seront interdites sur le serveur.

■ Les processus liés à la base de données

Ces processus peuvent être remontés en priorité, pour disposer de tout le temps CPU du serveur.

Gestion des disques : niveau de raid

La gestion des disques est très importante au niveau d'un serveur base de données et dépend des choix d'implémentation faits par l'administrateur de la base de données.



Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

stripping miroré

Raw device/systèmes de fichiers

Gestion des systèmes de fichiers

Gestion du réseau

Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des disques : niveau de raid

En fonction des résultats fournis précédemment, il est conseillé d'utiliser des volumes strippés/mirrorés (et ceci quelque soit le stockage effectué sur ces volumes).

Raw device/systèmes de fichiers

Les bases de données demandent de grandes capacités de stockage pour gérer leurs informations. Il est possible de choisir entre un stockage de type raw device ou système de fichiers :

- raw device
 - s'adapte à la structure de stockage de la base de données (en général, gestion de blocs de 2 K octets),
 - demande peu de buffers RAM (donc tout le reste peut être utilisé pour la share memory),
 - gestion non aisée de l'administration,
 - performance très liée aux SGBD.
- système de fichiers
 - impose une structure de stockage supplémentaire, qu'il set nécessaire d'adapter au SGBD (soit le SGBD utilise des blocs de données de 8 K, soit le système de fichiers utilise des blocs de 2 K),
 - mobilise plus de buffers RAM (asynchronisme) et demande une surveillance de `fsflush` (pour les inodes),
 - simplifie l'administration de l'espace.



Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

Création du système de fichiers

Montage du système de fichiers

VXFS

Gestion du réseau

Cas du serveur de base de données

Les choix liés au système d'exploitation

Gestion des systèmes de fichiers

- Création du système de fichiers

Il est fortement conseillé de modifier les paramètres du système de fichiers (nombre d'inodes, place laissée libre, etc.).

- Montage du système de fichiers

Le montage de type `forcedirectio` est préférable sur ce type de système de fichiers (on s'affranchit d'une partie du codage de l'application).

- VXFS

VXFS propose un système de fichiers dont le bloc initial est de 2 K octets, et une gestion des extents qui est celle utilisée en interne par les SGBD pour gérer leurs tables. Il est donc plus proche du monde de la base de données et donne des performances intéressantes pour les SGBD dont la taille du bloc ne peut être redimensionnée.

Gestion du réseau

Les interrogation des bases de données ont lieu via TCP, il est donc nécessaire de reprogrammer `tcp_close_wait_interval` (voire surveiller les SYN Attacks) et surveiller les connexions pendantes (imposer un `IDLE TIME` lors de la connexion).



Cas du serveur de base de données

Les choix liés aux applicatifs

Multi-bases de données

Privilégier des processus

Supervision de la zone de swap lors du multi-thread

Cas du serveur de base de données

Les choix liés aux applicatifs

Multi-bases de données

- Privilégier des processus

Si un choix multi-bases a été effectué (base de développement, base de test et base d'exploitation), il est possible de privilégier une base via les commandes de type `priocntl`, `pbind`, etc.

- Supervision de la zone de swap lors du multi-thread

Si le SGBD utilise le multi-thread, il est important de surveiller la zone de swap.



Cas du serveur WEB

Description de la machine

Les choix liés au système d'exploitation

Les choix liés aux applicatifs

Cas du serveur WEB

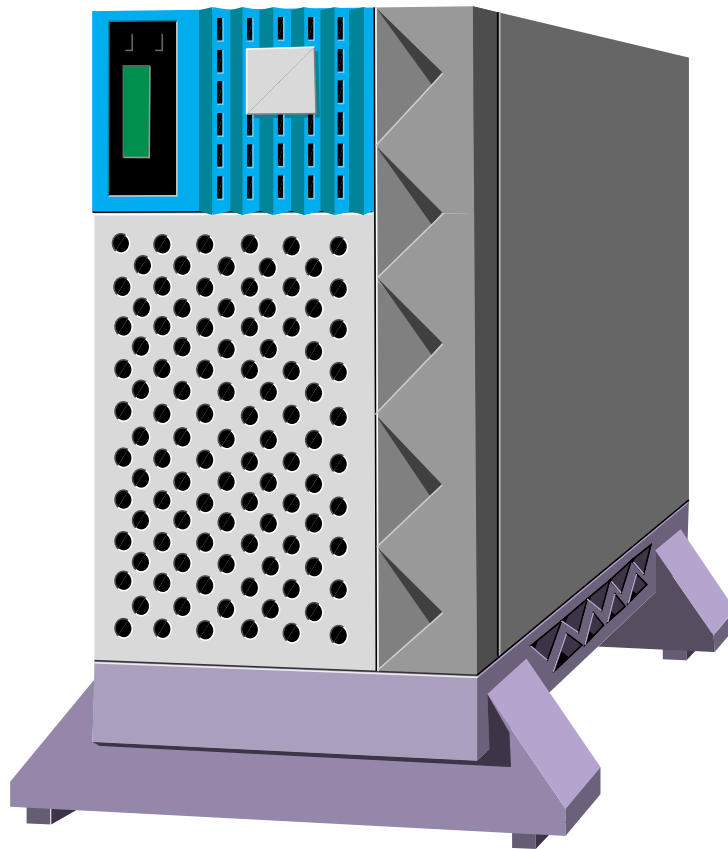
Les sujets que nous allons traiter recouvrent :

- la description de la machine : matériel disponible,
- les choix liés au système d'exploitation : les implémentations des services de base du système d'exploitation,
- les choix liés aux applicatifs : nous traiterons essentiellement le cas du service WEB.

Cas du serveur WEB

Description de la machine

Ultra™ Enterprise™ 150



Cas du serveur WEB

Description de la machine

Le matériel doit disposer de fortes capacités disques, et de bonnes capacités réseau.

Un logiciel de sécurité peut être adjoint à ce type d'installation. Ainsi certaines tâches réseau pourront être déchargées sur une autre machine.



Cas du serveur WEB

Les choix liés au système d'exploitation

Processus

Swap

Cas du serveur WEB

Les choix liés au système d'exploitation

Processus

La base de l'activité de la machine est de proposer un service concurrent. Un nombre de processus conséquent va donc être généré. Il est donc nécessaire de limiter les activités annexes.

Certains processus (liés au service d'annuaire) devront être surveillés (processus fuyants), ainsi que les mécanismes liés à SMTP.

Swap

Chaque processus va utiliser une quantité de mémoire non négligeable (environ 500 koctets), il est donc conseillé de surveiller les limites de type `maxpgio`, `lostfree`, etc.).



Cas du serveur WEB

Les choix liés aux applicatifs

Réseau

Nombre de sessions ouvertes

Syn Attack

Fin de connexion

Cas du serveur WEB

Les choix liés aux applicatifs

Réseau

Les machines fonctionnant en Solaris 2.5.1 doivent installer les patches 103630 et 103582 (au minimum version 15) pour pouvoir disposer de toutes les potentialités de programmation des ressources réseau.



Notes

Annexe A : Scripts et fichiers Réseau



- fichier S69inet
- snoop



Fichier S69inet

```
# This is the second phase of TCP/IP configuration.  The first part,
# run in the "/sbin/bcheckrc" script, does all configuration necessary
# to mount the "/usr" filesystem via NFS.  This includes configuring
# the interfaces and setting the machine's hostname.  The second part,
# run in this script, does all configuration that can be done before
# NIS or NIS+ is started.  This includes configuring IP routing,
# setting the NIS domainname and setting any tunable paramaters.  The
# third part, run in a subsequent startup script, does all
# configuration that may be dependent on NIS/NIS+ maps.  This includes
# a final re-configuration of the interfaces and starting all internet
# services.
#
#
# Set configurable parameters.
#
ndd -set /dev/tcp tcp_old_urp_interpretation 1
#
# Configure a default router, if there is one.  An empty
# /etc/defaultrouter file means that any default router added by the
# kernel during diskless boot is deleted.
#
if [ -f /etc/defaultrouter ]; then
    defroute="\`cat /etc/defaultrouter\`"
    if [ -n "$defroute" ]; then
        /usr/sbin/route -f add default $defroute 1
    else
        /usr/sbin/route -f
    fi
fi
#
# Set NIS domainname if locally configured.
#
if [ -f /etc/defaultdomain ]; then
    /usr/bin/domainname `cat /etc/defaultdomain`
    echo "NIS domainname is `/usr/bin/domainname`"
fi
#
# Run routed/router discovery only if we don't already have a default
# route installed.
#
if [ -z "$defroute" ]; then
    #
```

```

# No default route was setup by "route" command above - check the
# kernel routing table for any other default route.
#
defroute="`netstat -rn | grep default`"
fi

if [ -z "$defroute" ]; then
#
# Determine how many active interfaces there are and how many pt-pt
# interfaces. Act as a router if there are more than 2 interfaces
# (including the loopback interface) or one or more point-point
# interface. Also act as a router if /etc/gateways exists.
#
numifs=`ifconfig -au | grep inet | wc -l`
numpttifs=`ifconfig -au | grep inet | egrep -e '-->' | wc -l`
if [ $numifs -gt 2 -o $numpttifs -gt 0 -o -f /etc/gateways ]; then
# Machine is a router: turn on ip_forwarding, run routed,
# and advertise ourselves as a router using router discovery.
echo "machine is a router."
ndd -set /dev/ip ip_forwarding 1
if [ -f /usr/sbin/in.routed ]; then
/usr/sbin/in.routed -s
fi
if [ -f /usr/sbin/in.rdisc ]; then
/usr/sbin/in.rdisc -r
fi
else
# Machine is a host: if router discovery finds a router then
# we rely on router discovery. If there are not routers
# advertising themselves through router discovery
# run routed in space-saving mode.
# Turn off ip_forwarding
ndd -set /dev/ip ip_forwarding 0
if [ -f /usr/sbin/in.rdisc ] && /usr/sbin/in.rdisc -s; then
echo "starting router discovery."
elif [ -f /usr/sbin/in.routed ]; then
/usr/sbin/in.routed -q;
echo "starting routing daemon."
fi
fi
fi
fi

```



Snoop

snoop(1M) Maintenance Commands snoop(1M)

NAME

.....

Packet 101 Looks interesting. Take a look in more detail:
example\$ snoop -i pkts -v -p101

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 101 arrived at 16:09:53.59
ETHER:  Packet size = 210 bytes
ETHER:  Destination = 8:0:20:1:3d:94, Sun
ETHER:  Source      = 8:0:69:1:5f:e, Silicon Graphics
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:  ----- IP Header -----
IP:
IP:  Version = 4, header length = 20 bytes
IP:  Type of service = 00
IP:      ..0. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:  Total length = 196 bytes
IP:  Identification 19846
IP:  Flags = 0X
IP:  .0.. .... = may fragment
IP:  ..0. .... = more fragments
IP:  Fragment offset = 0 bytes
IP:  Time to live = 255 seconds/hops
IP:  Protocol = 17 (UDP)
IP:  Header checksum = 18DC
IP:  Source address = 129.144.40.222, boutique
IP:  Destination address = 129.144.40.200, sunroof
IP:
UDP:  ----- UDP Header -----
UDP:
UDP:  Source port = 1023
UDP:  Destination port = 2049 (Sun RPC)
UDP:  Length = 176
```

```
UDP: Checksum = 0
UDP:
RPC: ----- SUN RPC Header -----
RPC:
RPC: Transaction id = 665905
RPC: Type = 0 (Call)
RPC: RPC version = 2
RPC: Program = 100003 (NFS), version = 2, procedure = 1
RPC: Credentials: Flavor = 1 (Unix), len = 32 bytes
RPC: Time = 06-Mar-90 07:26:58
```

Sun Microsystems Last change: 14 Sep 1992 9

snoop(1M) Maintenance Commands snoop(1M)

```
RPC: Hostname = boutique
RPC: Uid = 0, Gid = 1
RPC: Groups = 1
RPC: Verifier : Flavor = 0 (None), len = 0 bytes
RPC:
NFS: ----- SUN NFS -----
NFS:
NFS: Proc = 11 (Rename)
NFS: File handle = 000016430000000100080000305A1C47
NFS: 597A0000000800002046314AFC450000
NFS: File name = MTra00192
NFS: File handle = 000016430000000100080000305A1C47
NFS: 597A0000000800002046314AFC450000
NFS: File name = .nfs08
NFS:
```

.....



Annexe B : Paramètres de configuration





Paramètres de configuration de l'environnement

Cas des processus

Limites des tables noyau

maxuprc = max_nprocs - 5

nproc = nombre de processus actifs

max_nprocs = 16 * maxusers + 10

Gestion de la mémoire

Variables	Valeurs	sun4c	sun4u	sun4d
maxpgio	40			
maxslp	20			
pagesize		4096	8192	4096
desfree	1/64 RAM	en page		
minfree	128 k	en page		
lotsfree	1/32 RAM	en page		
slowscan	100			
fastscan	nombre pages/2			

Gestion de la table des inodes

Valeur de ufs_ninode	2.5	2.6
ufs_ninode	max_nprocs + 16 + maxusers + 64	68 * maxusers + 360

Paramètres réseau

Nombre de connexions

tcpActiveOpens,

tcpPassiveOpens.

tcpListenDrop.,

tcpHalfOpenDrop

tcpListenDropQ0

tcp_conn_req_max_q

tcp_xmit_hiwat et tcp_recv_hiwat.

Temps de déconnexion

tcp_close_wait_interval.



Les IPC

File de messages

<code>msginfo_msgmap</code>	défaut	100
<code>msginfo_msgmax</code>	défaut	2048
<code>msginfo_msgmnb</code>	défaut	4096
<code>msginfo_msgmni</code>	défaut	50
<code>msginfo_msgssz</code>	défaut	8
<code>msginfo_msgtql</code>	défaut	40
<code>msginfo_msgseg</code>	défaut (<32768)	1024

Sémaphores

<code>seminfo_semmap</code>	défaut	10
<code>seminfo_semmni</code>	défaut	10
<code>seminfo_semmns</code>	défaut	60
<code>seminfo_semmsl</code>	défaut	25
<code>seminfo_semopm</code>	défaut	10
<code>seminfo_semvmx</code>	défaut	32767

Mémoire partagée

<code>shminfo_shmmax</code>	défaut	131072
<code>shminfo_shmmin</code>	défaut	1
<code>shminfo_shmseg</code>	défaut	6

Index

Symbols

/etc/system..... 4-7, 5-4,
..... 5-14
/var/adm/messages 4-7

A

accounting..... 3-53
acctcom 3-71
actimeo..... 2-111
adb..... 5-4,
..... 5-10
Adrian Monitor 3-81
at..... 5-44
automount 6-13,
..... 6-55
autoup..... 2-61

B

Bases de données 4-53
buffers 4-29
bufhwm 2-51

C

Cachefs..... 2-118,
..... 6-11,
..... 6-55
ckpacct 3-54
code..... 2-15
CPU..... 4-13
crash..... 5-4,
..... 5-12

cron..... 5-44

D

defaultrouter 6-9
DESFREE 2-39,
..... 2-44
développement 2-132
df 3-36
Disque 4-31
DNLC..... 4-25,
..... 6-37
dodisk..... 3-54

F

fastscan..... 2-44
FC-AL 1-58
forcedirectio 6-51
fsflush..... 2-61,
..... 6-37
fstyp..... 3-11,
..... 3-28,
..... 5-34

H

hardswap 2-42
heap 2-21
HTTP..... 2-130

I

IA 2-27



IDLE TIME.....	6-67	netstat	3-11,
image.....	2-15	3-13,
inodes	4-27,	3-33
.....	6-37	newfs	5-38
interruptions	2-11	NFS	2-105,
iostat.....	3-11,	2-126,
.....	3-26	4-41,
IPC	2-81,	4-43
.....	2-83,	nfsstat	3-13,
.....	4-7,	3-35
.....	5-20	nfswatch	3-79
L		NISplus.....	6-21
limit.....	2-21,	nm.....	5-6
.....	5-44,	noforcedirectio	6-51
.....	5-47	norouter	6-27,
LOSTFREE.....	2-39,	6-63
.....	2-44	nproc.....	2-21
		nscd	6-21
M		P	
max_nprocs	2-21,	page	2-17
.....	6-21	pagination.....	2-36,
maxpgio.....	2-44	2-39,
maxslp.....	2-44	2-41
maxuprc.....	2-21	pbind.....	6-69
maxusers	2-53	pcb.....	2-15
Mémoire.....	4-15	perfmeter	3-36
Mémoire partagée	5-24	pn_cnt.....	6-21
mémoire virtuelle.....	2-33	priocntl	5-26,
MINFREE.....	2-39,	6-69
.....	2-44	priorité.....	6-37,
mkfs.....	5-38,	2-27
.....	6-49	processeurs	1-18
monacct.....	3-54	Processus.....	4-19,
mpstat.....	2-9,	2-13
.....	3-11,	proctool	3-77
.....	3-20	prtconf	1-46
msginfo	5-20	prtdiag	1-46
N		ps.....	3-11,
ncsize.....	2-51,	3-23
.....	2-55	psrinfo	3-20
ndd.....	2-99,	R	
.....	5-40	RAID.....	1-74
		raid	6-39



Ultra™ Enterprise™ 150	
.....	1-31
Ultra™ Enterprise™ 3000	
.....	1-34
Ultra™ Enterprise™ 4000	
.....	1-36
Ultra™ Enterprise™ 450	
.....	1-33
Ultra™ Enterprise™ 5000	
.....	1-39
Ultra™ Enterprise™ 6000	
.....	1-41
uname.....	1-46
utlimit.....	5-47

V

var	6-19
VM	1-94
vmstat.....	2-11,
.....	3-11,
.....	3-15
VXFS	2-68,
.....	6-48,
.....	6-67

W

WEB.....	4-66
----------	------