Audit de la performance des systèmes Unix

Cédric Jacquot-Préaux

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ILE DE FRANCE

PARIS

MEMOIRE

présenté en vue d'obtenir

Le diplôme D'INGENIEUR CNAM

Spécialité : INFORMATIQUE

OPTION : ARCHITECTURE ET INTEGRATION DES SYSTEMES ET LOGICIELS

par Cédric JACQUOT-PREAUX

AUDIT DE LA PERFORMANCE DES SYSTEMES UNIX

Jury :

Yann Pollet (Président)
Yves Laloum
Patrice Lignelet

Membres :

Ghislaine Marchand
Jean-Marc Le Campion

Soutenu le mardi 24 octobre 2012

TABLE DES MATIERES

1 La performance des serveurs chez EDF – Généralités 11
1.1 Environnement des audits de performance 11
1.1.1 Organisation de l'entreprise 11
1.1.2 Présentation de la cellule audit de performance 13
1.1.3 Le périmètre technique 14
1.2 Contexte actuel des audits de performance au sein d'EDF 15
1.2.1 Contenu de l'audit de performance 15
1.2.2 Les commanditaires des audits de performance 16
1.2.3 Raisons principales de l'audit de performance à EDF 16
1.2.4 Les moyens de supposer un problème de performance 17
1.2.5 Causes génériques des problèmes de performance 17
1.2.6 Les attentes qualitatives du demandeur sur un audit de performance 19
1.2.7 Quelques remarques supplémentaires 20
1.3 Synoptique actuel pour le traitement des audits 22
1.3.1 Réception de la demande 22
1.3.2 Collecte des métriques sur le serveur 23
1.3.3 Analyse et recherche de la source du problème 24
1.3.4 Formulation des objectifs d'amélioration de performance et rédaction 24
1.3.5 Reformulation de recommandations 25
1.3.6 Fermeture de la demande de performance 25
1.4 Etude de quelques bonnes pratiques existantes à EDF 27
1.4.1 Journaliser l'activité de la machine 27
1.4.2 La gestion des capacités 27
1.4.3 Les Benchmarks 28
1.4.4 Mise en place des préconisations constructeur et veille technologique 28
1.4.5 ITIL et son intégration dans l'activité d'ASC 29
1.4.6 Résumé de l'existentiel en entreprise sur le thème de la performance 30
2 Diagnostic d'un problème de performance 33
2.1 Introduction 33
2.2 Plan succinct de chaque sous-parties 34
2.3 Les processeurs 36
2.3.1 Généralités 36
2.3.2 Différents outils triviaux pour superviser l'activité CPU 38
2.3.3 Les signes basiques d'une contention CPU 39
2.3.4 Types d'actions possibles sur les contentions liées à la CPU 41
2.4 Performance de la mémoire 43
2.4.1 Explication basique sur la gestion interne de la mémoire 43
2.4.2 Les problématiques mémoire en entreprise 45
2.4.3 Gestion de la mémoire, le cas de AIX 45
2.4.4 Gestion de la mémoire sous Solaris 49
2.4.5 Gestion de la mémoire sous Linux 52
2.4.6 La mémoire partagée et la SGA 55
2.4.7 Optimisations des caches 56
2.4.8 Surveillance et contrôle des processus 57
2.4.9 Préconisations mises en œuvre chez EDF pour l'optimisation mémoire 59
2.5 Performance et optimisation des disques et des systèmes de fichiers 60
2.5.1 Considérations générales de la problématique de la pile I/O 60
2.5.2 Performances des disques physiques 62

2.5.3 Sous-système de disques 64
2.5.4 Gestionnaire de volume et file system 65
2.5.5 Optimisation du système de fichier 65
2.5.6 Outils pour suivre les performances I/O disques et interprétation 69
2.5.7 Autres commandes moins usitées sous AIX 70
2.5.8 Cas particulier des disques logés en baie de stockage 70
2.5.9 Optimisation de la couche NFS 71
2.5.10 Préconisations générales sur les disques et couche I/O chez EDF 71
2.6 Analyse des performances réseaux 73
2.6.1 Généralités 73
2.6.2 Moyens d'observations du trafic réseau 74
2.6.3 Synoptique simple du traitement de la performance réseau 77
2.6.4 Mise en œuvre du paramétrage 78
2.6.5 Remarques importantes supplémentaires 80
2.6.6 Préconisations générales en terme de tuning réseau et mise en place chez EDF . 82
3 Considérations techniques supplémentaires et Outils. 83
3.1 Techniques de partionnement et micro-partionnement des CPU et virtualisation des processeurs sous AIX 83
3.1.1 Généralités sur la virtualisation 83
3.1.2 Le micropartionnement 83
3.1.3 Partage de CPU dans l'univers AIX : Mode de fonctionnement 84
3.1.4 Incidences sur les audits de performance : 85
3.2 Autres systèmes de Virtualisation utilisé pour la CPU 87
3.2.1 Allocations des CPU via ESX – VSPHERE pour les partitions Linux 87
3.2.2 Cas des plateformes SUN. 88
3.3 Virtualisation des cartes Fibre pour le SAN et Ethernet 88
3.3.1 Virtualisation des cartes fibres 89
3.3.2 Virtualisation des cartes Ethernet 89
3.4 Virtualisation de la Mémoire 89
3.4.1 Techniques sous VmWare 90
3.5 Outils d'aide aux audits de performance. 91
3.5.1 Etude des Outils existants au sein de la cellule. 92
3.5.2 Outils existants d'aide à la performance issus d'éditeurs de logiciel. 94
3.5.3 Propositions d'améliorations pour la collecte des données de performance dans le cadre d'une MCOI 98
3.5.4 OMNIVISION INVESTIGATION : 99
3.6 Améliorations fonctionnelles et organisationnelles des processus 102
3.6.1 Des points améliorants l'efficacité de ASC 102
3.6.2 Quelques points à améliorer 104
4 Bilan et Conclusion 107
4.1 Bilan de la mission 107
4.2 Conclusion 109

Note : Les annexes sont juxtaposées au document principal, elles contiennent pour la première annexe des mini-études de cas, la matrice de test pour VxVM , le script perfstat.sh (49 pages)et pour la deuxième annexe un audit complet d'une machine (26 pages). Elles ont leur propre table de matière.

Remerciements

Tout d'abord je tenais à remercier tout le corps enseignant du CNAM de Paris pour la qualité de l'enseignement dispensé. Je tiens particulièrement à remercier Yann Pollet, mon tuteur pour ce mémoire et professeur principal pour mon unité de Valeur C.

Je remercie également Eric Soudan-Gressier qui m'a aidé à trouver le sujet de mon mémoire d'ingénieur.

L'écriture de ce mémoire se veut un peu l'épitaphe d'une aventure commencée en 2004. Cette aventure n'avait pas initialement pour but de pousser mes efforts jusqu'au diplôme d'ingénieur : je participais seulement à quelques UE pour parfaire mes connaissances. Je fus au fur et à mesure convaincu par la qualité et l'approche des professeurs. Ainsi j'adresserai également mes remerciements à certains autres contributeurs de l'approfondissement de mes connaissances, qui m'auront marqué par leur qualité d'enseignement : Mesdames Delacroix, Wattiau et Bouzefrane; Messieurs Ranchin, Dememe, Arnaud, Servin, Florin, Farinone, Natkin, Pioch, Keryvel et Du Mouza.

Je voudrais également signaler la difficulté de compléter ce cursus ingénieur par un mémoire, car c'est un travail long et difficile. Je tiens absolument à remercier ma responsable côté EDF: Me Ghislaine Marchand, son support moral permanent et ses qualités managériales m'ont permis de me rendre à mes cours du soir sans problème. D'autres personnes ont été indirectement contributeurs à mon mémoire, dont Jacques Hulaux, ex-collègue érudit qui a pris le temps de relire mon mémoire pour le fond, de me formuler des remarques, mais aussi Jean-Marc Le Campion (agent EDF) pour son esprit critique et sa relecture.

Plus proche de moi, je tiens surtout dédié ce mémoire à mes parents, à mon père Daniel Trocmé décèdé en 2006 et à ma mère Elisabeth, toujours très présente pour me supporter moralement.

Convention typographique:

commande: désigne une commande système ou d'administration propre au système d'exploitation concerné comme exemple : ***echo***

paramètre : désigne un paramètre, d'un champ ou attribut d'une commande, également il peut s'agir d'un démon plus exceptionnellement **,** par exemple *fsflush*

`fichier :` Désigne un fichier de configuration au sein de l'arborescence du système d'exploitation comme `/etc/system`

Les résultats des commandes sont encadrées en type Courrier New : `Résultat`

Table des principaux acronymes.

Sigles Organisationnels.

EDF = Electricité de France
ASC = Audit Support Contrôle
DSP = Direction Des Services Partages
OI = Objet d'Infrastructure
CA = Chargé d'Application
RFC = Request For Change
ITIL = Information Technology Infrastructure Library
DSI = Direction du Système Informatique
OGD = Operateur Gestion des Données
MCOI = Maintien En Condition Opérationnelle Des Infrastructures
OI = Objet d'Infrastructure

Sigles Techniques

IHM = Interface Homme Machine
SGBD = Système De Gestion De Bases De Données
OS = Operating System
CPU = Central Process Unit
RAM = Random Access Memory
VMM = Virtual Memory Manager
VM = Virtual Machine
SR = Scan Rate
PI = Pagination In
PO = Pagination Out
CIO = Concurrent IO
DIO = Direct IO
FS = File System
RHEL = Redhat Entreprise Linux
I/O = Input/Output
CISC = Complex Instruction Set Computer
RISC = Reduced Instruction Set Computer
RSS = Resident Set Size
VMM = Virtual Memory Manager
JFS = Journalized Filesystem
UFS = Unix File system
NFS = Network File System
RAID = Redundant Area Inexpensive Disk
SDS = Solstice Disk Suite
SVM = Solaris Volume Manager
ODM = Object Data Manager
MTU = Maximum Transmitted Unit
LPAR = Logical Partition
VCPU = Virtual CPU
EC = Entitled Capacity
VIOS = Virtual I/O Server
NPIV = Network Port ID Virtual
LVM = Logical Volume Manager

Introduction

Dans un système informatique, les serveurs informatiques constituent un des éléments importants de l'architecture physique. Une entreprise comme EDF possède un parc de plusieurs milliers de serveurs, ces serveurs qui hébergent des applications se doivent en plus de la fiabilité de fonctionnement, de la disponibilité et fournir des temps de services ou de traitement avec des délais acceptables et conformes aux attentes.

Il est donc nécessaire pour toute entreprise soucieuse de ces dernières contraintes de disposer d'une cellule d'expertise dédiée au traitement de la problématique de la performance.

Les audits de performance de cette cellule sont donc un des moyens majeurs qui ont été mis en place mis par EDF pour la traiter. Si l'étude de la performance représente un investissement en termes de temps, de coûts, et de mise à niveau de connaissance; la contre-performance peut elle aussi engendrer des coûts bien plus importants aussi qu'en terme économique qu'en terme d'image de marque de la société.

L'étude de la performance revêt davantage d'une approche méthodologique, que d'une collecte d'actions et de règles visant à conformer ou normaliser un système en fonction de paramétrages ou de valeurs édictées dans diverses documentations ; c'est par ailleurs une notion fortement contextuelle, où l'obsolescence technique est un facteur omniprésent dans le domaine des systèmes d'exploitation.

Le but de ce mémoire est de présenter non seulement ce type de problématique dans un système informatique conséquent et hétérogène - comme celui d'EDF – mais aussi tel qu'il a été traité, par l'interprétation des informations pertinentes et par des actions concrètement mises en place. On se focalisera sur la partie Unix des systèmes d'exploitation des serveurs d'EDF, ce mémoire traitant en partie de la partie méthodologique et de compréhension du système : il est très facile de porter la méthode et la réflexion sur d'autres systèmes et contexte d'entreprise. De la même manière il ne peut se prétendre comme une référence et être entièrement exhaustif sur le sujet.

Ce mémoire s'articulera autour de trois parties, en premier lieu on se consacrera à présenter l'entreprise, son contexte et la cellule des auditeurs, à définir la problématique de la performance propre à cette entreprise, ainsi que les processus métiers attachés. Puis dans un deuxième temps, la méthode utilisée : des rappels techniques et synthétiques pour les différents éléments concernés par l'optimisation système, en fonction des systèmes d'exploitation, des outils ayant été utilisés, des métriques relatives et des paramètres principaux influençant la performance, notamment ceux que j'ai pu optimiser; et au final, ce qui peut être envisagé dans une approche généraliste et contextuelle.

Enfin la dernière partie abordera les évolutions techniques récentes des systèmes d'exploitation et du matériel telles que la virtualisation de celles-ci, impliquant d'ores et déjà de revoir certaines considérations pour l'audit de performance et les outils logiciels pour traiter le suivi de la performance avec leur forces et faiblesses et pour finir, quelques points d'améliorations possibles.

PREAMBULE

Problématique exposée

La question est la suivante : Comment aborder la problématique de performance des serveurs Unix/Linux au sein d'un système d'information vaste et hétérogène ?

Les raisons de ce choix de mémoire sont multiples :

- ✓ Réalisation d'une approche synthétique et transversale sur une problématique générique et permanente pour un système d'information

Ce mémoire se veut être la synthèse d'un travail effectué au cours de 5 années passé au sein d'une équipe dédiée à la performance des systèmes à EDF en valorisant ce qui est le plus souvent rencontré en terme de problématique de performance sur des serveurs de type Unix et des solutions techniques s'offrant à l'ingénieur système pour remédier à ce type de problème. Ces solutions se veulent assez simples à mettre en œuvre car les préconisations sont transmises aux équipes exploitantes des serveurs.

- ✓ Exposer la méthode de traitement des problèmes de performance utilisée en entreprise.
- ✓ Disposer pour EDF d'un document manuscrit inédit sur le traitement de cette problématique entreprise.

Ce mémoire permettra notamment aux équipes systèmes de disposer d'un guide synthétique qui permettra non seulement de comprendre prosaïquement les composants physiques et logiques concernés par la performance mais aussi de disposer des principaux outils et notions indispensables pour pouvoir effectuer un audit de performance.

- ✓ Evoquer l'incidence de la virtualisation et de la consolidation des composants pour la problématique de performance.

Bien que non exhaustif sur le sujet, ce mémoire évoque aussi les points qui sont à reconsidérer lors d'audit de performance sur des environnements virtuels.

La finalité de l'étude de la performance des serveurs est multiple, c'est l'amélioration :

- ✓ De l'emploi des ressources
- ✓ Des conditions d'utilisation pour les utilisateurs.
- ✓ Du comportement de la machine pour une montée en charge

- ✓ <u>Ouvrages de références au sujet de la performance et avertissement.</u>

Il existe un certain nombre d'ouvrages sur le sujet, de notes et des formations avec les supports de cours dispensés par les éditeurs et constructeurs. Par ailleurs je me suis servi et inspiré de certains de ces écrits, non seulement pour mieux comprendre, par exemple, certains mécanismes internes des systèmes d'exploitation par exemple la gestion de la mémoire centrale (tuner un système c'est avant tout le comprendre), pour connaître les principaux outils propres à chaque OS, et tous les moyens disponibles pour optimiser.

Les noms des ouvrages de référence utilisés pour ce mémoire sont en annexe bibliographique. La liste est non exhaustive surtout en ce qui concerne les ouvrages sur Linux. Ces ouvrages sont en langue anglaise et généralement écrits par des experts renommés chez les constructeurs (ingénieurs seniors – docteurs en informatique). Les plans de ces ouvrages destinés à mener à bien des résolutions sur la problématique de performances sont peu ou prou identiques. Ce synoptique de raisonnement est souvent le même pour des raisons logiques que j'expliquerai dans le présent document.

On peut formuler un certain nombre de reproches à ces sources:

- Nombre de ces livres sont obsolètes techniquement, car pour bon nombre d'entre eux ont plus de 5 ans. Or les améliorations techniques ont été nombreuses surtout en termes d'environnement virtuel.
- L'exhaustivité des paramétrages systèmes édictés dans ces ouvrages rend le tuning[1] complexe et illisible, et certaines propositions sont difficiles à mettre en œuvre.
- L'accent n'est pas assez mis sur le manque de pertinence de certains paramétrages (beaucoup de paramètres ont une influence peu discutée ou/et discutables) et de manquer de souligner les leviers les plus efficaces.
- Le manque, pour la plupart, d'un synoptique de raisonnement, d'accent sur les métriques critiques, et de propositions de quantification au lieu de simples descriptions des phénomènes.
- Peu abordent la performance dans un domaine de consolidation des ressources ou d'un système virtualisé, rendant par ailleurs certaines considérations erronées.
- Certains ouvrages manquent de signaler ou d'expliquer succinctement les mécanismes internes.
- Très orienté constructeur ou éditeur, certains ouvrages n'établissent pas de parallèle avec les autres systèmes d'exploitation et ne proposent pas une vision « inter-OS ».

De même, il n'est pas dans le cadre de ce mémoire de réécrire un livre sur ce thème, mais de présenter plus succinctement ce qui est réalisable ou a été fait dans le contexte d'EDF, au cours de mes quelques années d'étude de ce problème. D'autre part les solutions exposées se veulent en général relativement simples et rapides à mettre en œuvre, faisables et à la portée technique des équipes en charge de l'exploitation des serveurs. L'étude de ces problématiques de performance portait en majorité sur des serveurs en production, où l'expérimentation a peu sa place. Jouer « les apprentis sorciers » sur des machines exploitées avec des enjeux financiers conséquents est proscrit. Davantage de possibilités d'expérimentation étaient données sur les machines de recettes ou de développement mais pour certaines l'activité utilisateur n'est guère reproductible ou ne sont pas de même configuration.

Il est toujours possible d'améliorer les performances d'un serveur en allant plus en profondeur comme redévelopper un pilote ou aller modifier des paramètres dans les mécanismes internes de l'ordonnanceur … . Ce mémoire n'est donc pas un document pour « geek »[2] , il est loin d'être exhaustif sur le sujet et présente des solutions courantes à mettre en œuvre dans un milieu industriel avec ses contraintes inhérentes.

[1] Le tuning est un anglicisme désignant une activité d'optimisation pour des raisons d'aisance rédactionnelle, ce mot sera souvent employé pour désigner de manière équivalente l'activité d'optimisation.

[2] Terme anglophone désignant des férus d'informatique et de codage

Aussi il est impossible dans le cadre d'un mémoire d'exposer toutes les pistes techniques concourant à l'amélioration de la performance. Le tuning système est une activité qui a comme sous-jacent des mécanismes complexes et est souvent affaire de compromis.

Ce mémoire se limite donc à mon expérience personnelle en termes de tuning sur des systèmes Unix/Linux et se veut relativement pédagogique dans la mesure où il est indispensable, avant toute optimisation et de traitement de la performance de comprendre la manière dont un système fonctionne. Ce document aborde et répond à la problématique donc de manière relativement générique et contextuelle à EDF.

1La performance des serveurs chez EDF – Généralités

1.1Environnement des audits de performance

1.1.1 Organisation de l'entreprise

✓ Présentation générale de l'entreprise

EDF, acronyme pour Électricité de France, est le premier producteur européen d'énergie électrique et se classe parmi les premiers acteurs mondiaux. EDF a une intégration verticale de la filière Énergie puisqu'il est à la fois producteur, transporteur, distributeur et revendeur d'énergie. Elle produit l'électricité essentiellement à partir de la fission nucléaire et des centrales hydrauliques, ainsi que de quelques centrales thermiques. EDF est le principal producteur et distributeur d'électricité de France, des outsiders sont cependant sur les rangs (dans la distribution ou la production comme GDF-Suez ou Direct Énergie…).

Le secteur de l'énergie fait partie d'un des secteurs majeurs puisqu'on estime que les besoins énergétiques dans le monde doubleront d'ici 2050. C'est un enjeu énorme notamment dans les marchés émergents, ainsi EDF est déjà présent dans certains pays d'Europe centrale et orientale comme la Pologne ou la Slovaquie, en Afrique, au Vietnam, et dans le cadre de la concentration des acteurs, a pris des participations majoritaires dans British Energy en Grande Bretagne.

EDF est mondialement reconnu dans sa maîtrise de l'énergie nucléaire et a une recherche active (projet EPR) ; EDF investit plus de 350 Millions d'euros par an en R&D avec 2000 chercheurs et 390 innovations par an. Le chiffre d'affaire du Groupe est de 64 Milliards d'Euros en 2008 dont 47 % réalisé à l'international, le réseau de l'énergie est fortement interconnecté entre les acteurs européens puisque l'énergie électrique est difficilement stockable à grande échelle.

Pour finir, EDF est l'un des grands employeurs français (105 000 en France en 2008) et 160 000 environ dans le monde, la population est essentiellement constituée de 54% d'agents de maîtrise, de 25% de cadres et de 21% d'ouvriers.

✓ Présentation du SI d'EDF

Les deux grands centres de productions EDF des systèmes informatiques sont les sites de Clamart et de Pacy/Eure en Normandie avec des Centres Régionaux situés à Lyon, Nantes, Lille… et des antennes reparties sur d'autres sites ; un autre site de production est en construction dans l'Eure, des exercices de PRA (Plan de Reprise d'Activité) sont d'ailleurs effectués semestriellement.

L'essentiel des effectifs de la DSP[3]-IT se situe soit à Clamart ou à Nanterre et gère:

- 90 000 postes de travail
- 300 applications majeures

[3] Acronyme désignant la Direction des Services partagés

➢ 6000 serveurs dont 2800 exploités par EDF directement

La DSP-IT est composée de 4 centres de Services Partagés (CSP): AOA [4]& services (Achat), comptabilité, ressources humaines (RH) et informatique (IT).

Le CSP IT (Centre des Services Partagés) est principalement constitué de 3 branches :

➢ La branche développement Projet Maintenance, en charge de développement et la maintenance en conditions opérationnelles des SI EDF.

➢ La branche Informatique Télécom et Services, en charge de l'intégration des services à l'utilisateur autour du poste de travail plus communément appelé Helpdesk.

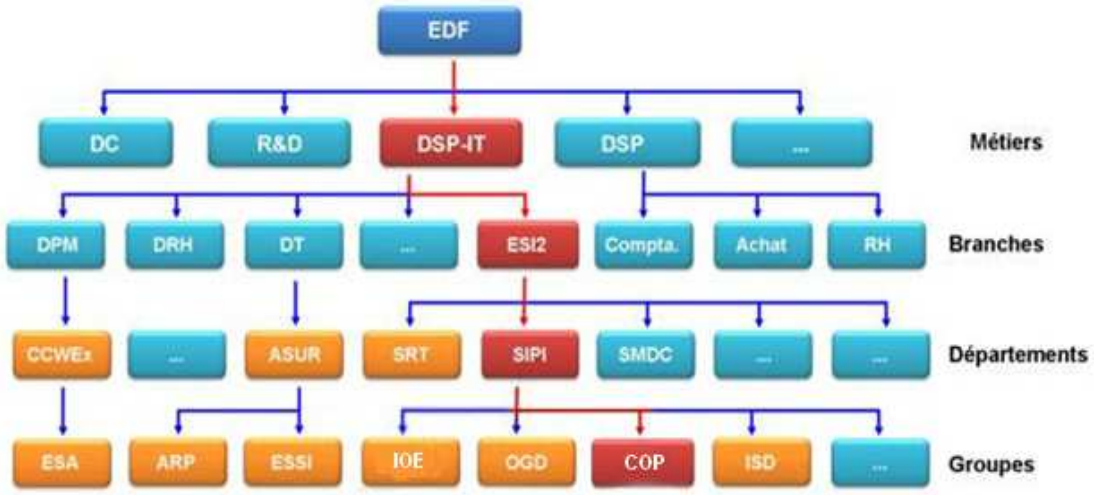
➢ La branche des SI Métiers et des Services d'Infrastructures (ESI2) en charge d'assurer la maîtrise de l'exploitation et de ces services.

Cette dernière branche est garante du fonctionnement des services d'infrastructures informatiques de l'entreprise, de l'exploitation des applications des SI-Métiers et leur intégration dans l'environnement de production.

SIPI (Service d'Ingénierie de Production et d'Infrastructures) a pour mission de proposer des services dans le domaine des serveurs, de l'outillage et de la gestion de données. Il fournit également une expertise dans et une ingénierie d'exploitation sur le périmètre du système informatique EDF.

Dans ce groupe on trouve le contrôle de production (COP) chargé du contrôle du parc informatiques constitué de machine gérées par des sociétés de service externes[5]. Ce groupe COP est constitué de 3 entités; un pôle de fournisseur de services (START et Omnivision) et un pôle de fournisseur d'outils (DEV) et de développement et enfin ASC[6].

Schéma hiérarchique des entités, notamment le positionnement de COP



[4] Administration des Obligations d'Achat

[5] A ce jour 3 sociétés informatiques ont la charge de l'infogérance les serveurs du SI EDF. On les dénommera infogérants par commodité de langage.

[6] Audit Support Conformité

1.1.2 Présentation de la cellule audit de performance

La cellule auquel j'appartiens est en charge de plusieurs tâches, l'équipe est composée d'une dizaine de personnes d'une société de prestation informatique managée par des agents EDF et un manager côté prestataire de service.

Cette équipe est constituée de trois binômes (pour des raisons de couverture) classés par OS[7] : 1 binôme AIX (OS d'IBM), 1 binôme (Solaris et Linux – Redhat) et un Binôme MS Windows. Cette prestation a été mise en place il y a 6 ans.

Un ingénieur système au sein de la cellule ASC effectue 3 types de taches :

✓ Une fonction de support Niveau 2 et 3 à la résolution d'incident

Cette fonction appelle l'ingénieur système en tant qu'acteur à la résolution d'incidents d'un haut niveau technique. En tant que support dit de Niveau 2 et 3, notre cellule doit être sollicitée après étude de la problématique par un administrateur système ou un exploitant de la machine.

Ce support vise à apporter des réponses ou des solutions techniques relatives à un problème rencontré par un infogérant ou administrateur des centres régionaux, voire pour des projets sur des plateformes d'intégration ou de développement, ou des machines de production. Ce support peut aussi prendre la forme d'une recherche documentaire : notices constructeurs disponibles ou surtout des procédures mise à disposition par l'ingénierie des systèmes EDF.

Certaines demandes de support nécessitent une intervention rapide sur la machine via une cellule de crise activée pour l'occasion, dans ce cas on doit indiquer à l'administrateur système les actions à effectuer afin de rétablir au plus vite l'opérabilité de la machine en premier lieu et des temps de réponse adéquats. Les activités de support liées en partie à des problèmes de fonctionnalités et de performances sont donc parfois très fortement couplées.

Dans le cadre du support nous avons également pour mission de proposer des plans d'action ou d'être le référant avec le constructeur pour la mise en place ou d'une escalade technique vers le constructeur, notamment pour des mises à jour ou de correctifs ou tout simplement pour des questions plus pointues d' anomalies non connues : ces plans d'actions sont proposés à la cellule initiatrice de l'incident.

La gestion de ces incidents dans sa globalité suit une suite de processus tel qu'ITIL le conçoit.

Les incidents sont qualifiés, enregistrés, priorisés, diagnostiqués, résolus si la solution est relativement immédiate et acheminés vers les personnes les plus compétentes ; les utilisateurs sont informés et au final l'appel est clos. 80-85% des incidents proviennent d'un changement.

✓ La fonction d'audit de performance

Énoncés précédemment certains supports donnent aussi lieu à des audits de performances. Ces audits de performances peuvent être dus à plusieurs causes. Ils reposent sur une collecte de données et métriques ou bien sur des outils de monitoring temps réels.

[7] Operating System ou Système d'exploitation, nous utiliserons par facilité de langage et d'écriture souvent OS.

A partir de cette collecte d'informations, on proposera des optimisations systèmes : modifications de paramétrage ou implémentations architecturales possibles. Ces audits sont formalisés par un document écrit, ou bien lorsqu'il s'agit de situations de crise, par des préconisations données à la volée de manière orale et suivies par un mail.

✓ Tests des solutions ou des objets d'infrastructures

Il peut être demandé de participer aux tests de solution mise en place (installation d'OS ou d'OI[8]), ceci ayant pour but d'augmenter notre efficience sur le support.

Les tests sur les objets d'infrastructures et les souches n'étant jamais totalement exhaustifs, notre cellule peut aussi remonter les alertes et anomalies décelés lors de leur mise en place ou exploitation quotidienne et de faire vivre la base de connaissance.

✓ Différenciation des fonctions d'audits et de support

Des fonctions énumérées ci-dessus, il convient de différencier donc la fonction de support et celle d'auditeur.

La problématique performance a trait à une dégradation graduelle, spontanée ou intermittente de la machine d'une application ou d'un service. On parlera davantage de service dégradé que d'indisponibilité de service, qui est en soit un problème de performance « extrême ».

Le support technique ne nécessite pas en général la surveillance ou l'observation de l'activité de la machine via des métriques, les données observées seront en général statiques, des fichiers de configuration par exemple ; contrairement à la performance, où la notion de ce qui est observé revêt d'un caractère plus dynamique et spatio-temporel.

Le support technique se réfère à un problème clairement identifié : par exemple l'impossibilité d'effectuer une tache, l'indisponibilité d'une application ou fonction, la mise en exergue d'erreurs logicielles ou matérielles flagrantes ou plus simplement répondre à des questions techniques.

1.1.3 Le périmètre technique

La cellule intervient sur un parc de 4000 machines tout système confondu. L'essentiel des machines de production / intégration / tests sont des machines de type UNIX. Les 3 grands constructeurs représentés au sein du PARC EDF sont :

- **SUN MICROSYSTEMS – ORACLE** avec son OS : Solaris
- **IBM** avec son OS : AIX via BULL qui revend les solutions IBM
- **Helwett Packard,** principal constructeur intégrant les systèmes Windows de Microsoft et supportant via sa gamme Proliant la distribution Linux Redhat

La distribution Linux Redhat étant donc présente sur les serveurs HP (gamme Proliant) et IBM via sa gamme X3650.

[8] Objets d'infrastructure, il s'agit de modules additionnels ou de progiciels

Répartition des OS dans le parc EDF fin 2010 :

Système d'exploitation	Nombre de serveurs	ratio
AIX (IBM)	1344	35.7%
Linux (Redhat)	364	9.67%
Solaris (SUN)	1205	32.0%
WINDOWS (MICROSOFT)	847	22.5%
Autres (HP-UX)	3	0.3%
Total	3763	100%

On note la montée en puissance de système Linux (Redhat) et des systèmes virtualisés depuis quelques années. La décroissance de Solaris est due à des raisons d'obsolescence technique de la gamme matérielle et logicielle, au manque de solutions de virtualisation et à l'aspect trop intégré avec la solution Oracle et surtout d'un choix politique du DSI.

L'hétérogénéité des versions de OS et du matériel sur les audits de performance complexifient l'étude de la performance, ceci étant lié aux :

- Différences matérielles, les systèmes d'exploitation dans le catalogue produit mis à disposition dans le catalogue EDF.
- Centres régionaux qui ne dépendent pas des mêmes services achat et donc peuvent avoir du matériel plus «exotique» voire obsolètes à traiter.
- Outils de supervision qui ne sont pas toujours à jour par rapport aux dernières implémentations matérielles ou logicielles, notamment sur les processeurs multi-cœurs avec le multithreading ou des machines virtuelles.
- Obsolescence des données de collectes avec les nouvelles versions, les scripts doivent évoluer en fonction des nouveautés (éviction de certaines métriques au profit d'autres ou tout simplement suppressions de certaines mesures rendues obsolètes par les évolutions logicielles ou matérielles).
- Difficulté d'expertiser des machines/technologies dépassées ou tout simplement des machines trop récentes, il faut être qualifié et formé aux dernières technologies.

1.2 Contexte actuel des audits de performance au sein d'EDF

1.2.1 Contenu de l'audit de performance

Un audit de performance est un rapport dactylographié par un auditeur de la cellule ASC pour une personne ayant sollicité cette expertise par une demande explicite. Ce rapport est un document rapportant :

- La finalité de l'audit et les anomalies constatées

- Un résumé de la configuration
- Des mises en forme graphiques des mesures collectées par les sondes
- La synthèse et conclusions afférentes aux observations des métriques
- Les préconisations éventuelles sur le paramétrage ou sur les améliorations architecturales.

Ce rapport ne doit pas être non plus un cours de performance ou une explication en termes trop techniques et abscons, mais une lecture simple, précise et intelligible pour les destinataires qui ne sont pas souvent très techniciens.

1.2.2 Les commanditaires des audits de performance

Les demandes d'audits de performances émanent généralement d'un seul type de personnes au sein du SI d' EDF.

Les Charges d'applications sont en général les personnes commanditaires des audits, elles sont chargées de la «vie» de l'application en général, elles s'assurent que les fonctionnalités des applications, leur opérabilité, et leur qualité de service soient respectées. Elles doivent être l'interface entre les études, les architectes, les administrateurs systèmes, les intégrateurs et l'Exploitation.

Enfin les architectes du SI ou les Études et Projets dans le cadre de test ou de tirs peuvent aussi demander des audits sur une machine afin de mesurer et d'apprécier si elle n'est pas trop chargée, ou bien si elle peut accepter une nouvelle application.

1.2.3 Raisons principales de l'audit de performance à EDF

Il peut y avoir plusieurs raisons pour un chargé d'affaire de demander un audit:

- Contre-performance avérée sur alerte d'un tiers (problème dans les temps de réponses d'une application, comportement anormal de l'application).
- Demande d'un aperçu de l'activité de la machine après ou avant l'installation d'une application, ou montée de version d'un logiciel (ces audits se déroulent donc en général en deux temps et se rapprochent parfois à du Capacity Planning).
- Pour des campagnes de tests ou de tirs (parfois appelées Benchmark)
- Audit accompagnant la mise à jour matérielle d'un serveur ou servant de justificatif à sa mise à jour ou d'upgrade matérielle.
- Enfin les demandes faites à mauvaise escient. Le demandeur suppose que la machine a un problème alors que les suppositions sont infondées ou que cela peut être lié à un dysfonctionnement exogène au système (problème de SAN par exemple ou de dysfonctionnement réseau).

1.2.4 Les moyens de supposer un problème de performance

Ce paragraphe reprend et détaille la première cause avancée pour la demande d'un audit à savoir un problème de performance supposé, dont les signes les plus communs sont :

- Un ressenti de lenteur applicative ou de temps de réponse du système,
- Une exécution de taches trop longues (via Ordonnanceur ou exécution programmée)
- Alerte répétée d'un logiciel de monitoring (concernant ces logiciel, voir en 3eme partie).

A noter que ces dégradations peuvent être ressenties de manière graduelle. Parfois l'application est en état de dysfonctionnement complet (blocage, crash ou bien temps de réponse extrêmement élevé).

Ces blocages applicatifs en production donnent souvent lieu à des conférences de crise qui consistent à mettre "autour de la table" divers experts, en vue de trouver une solution curative dans l'urgence et de rétablir le bon fonctionnement de l'application. Il est à noter que ce qui importe aux "métiers" est d'avoir une application opérante, le serveur faisant uniquement office de socle physique.

1.2.5 Causes génériques des problèmes de performance

Les causes d'un problème de performance de manière générique peuvent être classées en deux types :

Celles qui sont **endogènes** au système :

- Mauvais paramétrage initial du système, mis en exergue par une montée en charge opérée par une application plus consommatrice de ressources.
- Dysfonctionnement matériel inopiné.
- Lacune capacitive sur un ou plusieurs composants systèmes (exemple : manque de mémoire vive).
- Anomalie de la souche système révélée de manière fortuite.
- Incompatibilité entre des modes de fonctionnement dans le système lié aux applications (paramétrages antagonistes).
- Suite au passage de correctifs sur la souche système.

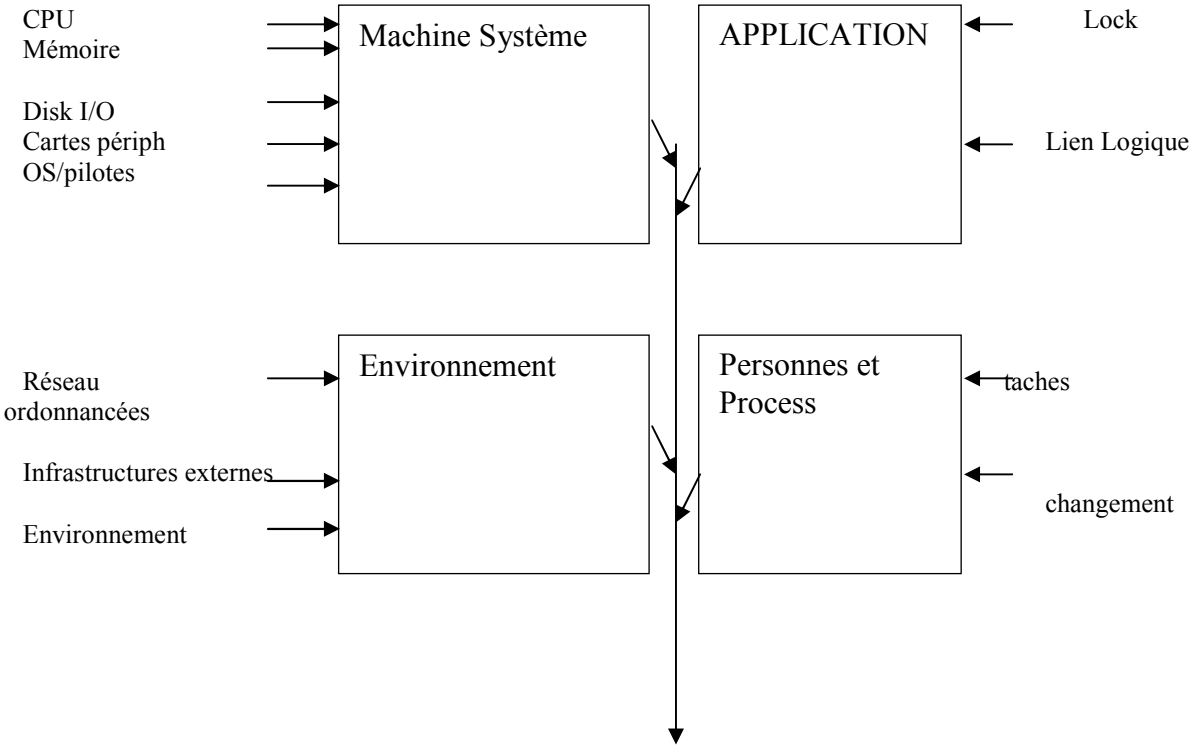
Et celles **exogènes** au système :

- Mise à jour de progiciels, ou montée de version occasionnant des impacts négatifs sur le système.
- Montée en charge du nombre d'utilisateurs sur l'application mettant en exergue le sous-dimensionnement.

- Dysfonctionnement sur le réseau (engorgement, routeur défectueux), nouvelle topologie réseau ou changement/contre-performance de matériel dans le SI impactant la machine (exemple : anomalie sur une baie de disque).
- Mauvais paramétrage des applications (c'est une cause majeure de la sous-performance des machines, cause qui par ailleurs doit être démontrée).

Au final, avoir connaissance des changements passés (matériel et logiciel) sur une machine revient parfois à trouver une des causes probables de la perte de performance.

✓ Origines des problèmes par un Diagramme d'Ishikawa :



1.2.6 Les attentes qualitatives du demandeur sur un audit de performance

De prime abord les attentes dépendent du type de demande, de ce fait la finalité et le contenu du rapport peuvent légèrement différer ; ce rapport nécessite plusieurs jours de rédaction après la collecte des données quand le diagnostic est complexe.

En fonction du type de la demande certains points peuvent être mis plus ou moins en avant:

- Dans le cadre de la résolution d'une dégradation de performances, les remarques et recommandations sont primordiales avec l'appui des journées symptomatiques, dans un suivi de tirs l'interprétation des métriques est à mettre plus avant.
- Dans une validation d'une montée applicative, les pré et post installation de l'application sont à mettre en relief.

Dans les recommandations produites à l'issue de l'audit certaines nécessiteront des prérequis ou seront à effectuer en concomitance avec d'autres recommandations, certaines peuvent renvoyer à des remarques à faire valider par d'autres cellules techniques.

Les propositions faites peuvent être des solutions de type palliatif, curatif ou préventif. Des remarques peuvent être faites à la marge de ce qui est demandé même si aucun problème n'est à priori décelé lors de l'audit - par exemple, une configuration présentant un risque potentiel pour la performance.

Dans certains cas il est envisageable de faire un plan de tests sur les paramètres proposés afin de trouver la meilleure adéquation, ces plans de test sont donc abondants en terme de métriques à faire figurer sur le rapport.

La problématique de la performance est fortement liée à celles des applications et de leurs contraintes en termes de qualité de service. C'est par ailleurs dans la formulation de la demande que l'on comprend que le demandeur a une dégradation de performance non pas explicitement sur un serveur mais, le plus souvent, sur le fonctionnement d'une application.

Or une application est souvent la coopération de différents modules se trouvant sur plusieurs machines. Lors des demandes, il est assez fréquent de demander simultanément l'étude de plusieurs machines participant au fonctionnement de l'application. Le point de contention peut se trouver en effet sur des machines limitées en termes de ressources. L'application repose donc sur un ou plusieurs serveurs ; de même, un serveur peut héberger une ou plusieurs applications. Un problème au niveau de l'application n'a pas forcement pour origine un problème de dimensionnement sur le système.

Traiter un problème de performance au niveau de l'application revient à mesurer et juger si le système est la source du problème ou non, dans le cas contraire à demander à étudier les paramétrages de l'application ou du logiciel.

Si le système est hors de cause, les conclusions de l'audit peuvent comporter des optimisations à la marge mais doivent aussi orienter le demandeur sur l'origine probable, qui peut être hors de notre périmètre de compétences. Une machine peut comporter plusieurs

applications en cohabitation, l'audit est aussi un moyen de vérifier qu'il n'existe pas d'externalités négatives entre les applications. Une application trop consommatrice sur une machine mutualisée, abritant plusieurs applications indépendantes ou non, peut gêner une autre application.

Une vision globale du fonctionnement de l'application est nécessaire, les documents d'architecture à jour doivent être fournis dans cette optique et sont indispensables à une étude complète (ces documents ne sont fort heureusement pas toujours nécessaires à l'étude de la performance).

1.2.7 Quelques remarques supplémentaires

Un tuning de serveur ne peut pas tout résoudre, notamment les problèmes liés aux pannes matérielles. On rentre alors dans une logique de remplacement et de contrat de maintenance matérielle. Un tuning ne peut s'opérer que sur du matériel fonctionnel.

Estimer l'impact d'une dégradation de performance revient à évaluer des types de solutions à mettre en face de ce problème et le temps que l'auditeur doit accorder pour balayer les solutions d'optimisation qui s'offrent à lui : il faut être vigilant sur le rendement entre gain et temps passé à optimiser le système.

En effet une dégradation de performance constatée de 5 % ou de 80 % ne suscite pas les mêmes interrogations, les mêmes solutions, le même investissement : ne pas faire du surinvestissement intellectuel ou de temps pour un gain marginal faible.

Dans certains cas l'audit peut déboucher sur une revue d'architecture (non traitée par la cellule).

Dernier aspect et non des moindres, l'audit de performance a aussi pour finalité d'éviter d'avoir à acheter en supplément du matériel onéreux, d'utiliser au mieux les ressources du système et de limiter l'impact des indisponibilités des applications métiers. Le tuning peut donc éviter de se lancer dans une logique d'investissement matériel trop prématurée et dispendieuse inutilement, *in fine* il suffira d'acheter que ce qui est nécessaire et ce que le tuning ne peut résoudre totalement.

Parmi les attentes liées aux audits de performance, il faut notamment préciser une différence sémantique entre un système dit performant et un système optimisé ou optimal.

La problématique de performance introduit la notion d'attente des utilisateurs/équipes par rapport à un cahier des charges, notamment en terme de temps de réponse (différence entre la soumission d'une requête et la réception de la réponse), de débit (le débit étant lui-même corrélé au temps: quantité rapportée au temps Io/secondes; transaction/sec; Mo/sec)…

Ces attentes seront dépendantes des spécifications matérielles et des paramétrages logiciels.

La notion de performance est aussi très subjective puisse qu'elle renvoie à un dimensionnement qui se fait par rapport à des attentes humaines (parfois basée sur le vécu des équipes projets ou métiers) ou bien très pragmatique si basée sur le différentiel entre ce qui est observé et promu par le constructeur. Au final les utilisateurs jugeront un système performant comme étant principalement des conditions d'utilisation normales de leur serveur ou de leur application.

Ces évaluations de performance sont dépendantes même des contraintes métier du client (transactionnel d'ordres à haute fréquence dans le milieu bancaire par exemple ou la milliseconde a son importance)

Un système optimisé relève de la notion de la meilleure allocation possible en fonction de besoins et du profil de la machine. Un système performant n'est pas forcément un système correctement optimisé et *vice versa*. A partir du moment où le système est vu comme performant, puisqu'il se conforme aux attentes des personnes l'utilisant, il est de leur point de vue « correctement optimisé », ce qui n'est parfois pas celui de l'ingénieur système en charge de l'audit.

Un système correctement optimisé serait un système pour lequel il n'y aurait pas de gaspillage de ressources (surdimensionnement) ou de carence (sous-dimensionnement). Un système optimal est un système pour lequel il y a une maximisation forte des contraintes afin de trouver la configuration la plus adéquate : d'où un amalgame possible entre les notions de système performant et de système optimisé, où l'on a tendance à admettre qu'un système est performant s'il est optimal.

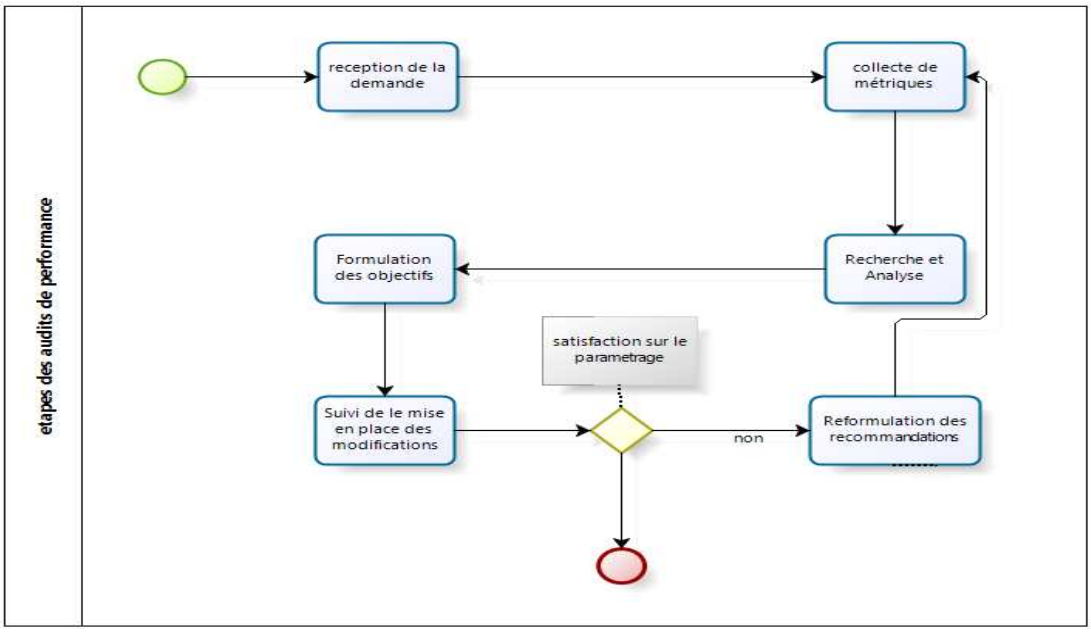
Chez EDF on associera davantage la notion de performance à la notion de respect des temps de réponse ou de traitement et de service rendu qu'*a contrario* de savoir si le système est réellement optimisé ou optimal ; bien souvent, pour être dit performant, le système sera volontairement surdimensionné ce qui indique que le travail de recherche de l'optimum est une notion davantage délaissée au détriment de la notion de performance.

Parvenir à un système optimal suppose également, en dehors du travail du tuning de la cellule, que la conception architecturale et le capacity planning aient été menés à bien.

1.3Synoptique actuel pour le traitement des audits

A l'instar de la fonction de support, le circuit de processus des audits est très semblable à ce que peut préconiser ITIL. Il s'agit du circuit existant dans le traitement des audits.

✓ Schéma simple du processus de traitement des audits



1.3.1 Réception de la demande

Le CA [9], *via* un formulaire, instancie une demande d'audit de performance ; cette demande doit être la plus complète possible.

Ce document a pour but de :

- Décrire les symptômes rencontrés,
- Dénommer les serveurs concernés,
- Indiquer une période d'étude,
- Définir les attentes relatives à cet audit,
- Et éventuellement de réceptionner en pièce jointe les documents liés à l'architecture de l'application.

[9] C.A = Chargé d'Application

Il s'en suit une prise en compte logicielle (traçabilité des demandes) du côté de la cellule et de s'entretenir éventuellement avec le demandeur, verbalement, afin de bien cibler ses besoins et d'obtenir les éléments les plus factuels possibles; il faut se méfier des liens de causalité évidents émis par le demandeur, notamment en assertions rapides ou interprétations trompeuses. Nous n'observons bien souvent que l'une des conséquences visibles d'un dysfonctionnement qui peut être plus profond et pernicieux.

1.3.2 Collecte des métriques sur le serveur

Il s'agit des métriques et des fichiers journaux. Cette étape vise à collecter en premier lieu toutes les données nécessaires à une expertise sur le serveur.

Cela passe par une analyse de deux types de fichiers, les fichiers enregistrant l'activité de la machine, les transactions et les journaux du système, mais également les métriques systèmes mesurant les activités disques, CPU, mémoire et réseau (script ou sonde). Cette collecte doit être faite de manière préventive et systématique.

La collecte des métriques systèmes doit avoir un historique suffisamment conséquent (environ 10 j) afin de pouvoir «profiler» le fonctionnement de la machine de manière nominale et standard[10], mais aussi prélever la période sur laquelle le début du dysfonctionnement a été relevé.

A la marge, l'auditeur peut implémenter une sonde plus spécifique à l'étude du problème : par exemple les processus les plus consommateurs et des métriques de retransmission de paquets ou autre script *Shell* pouvant journaliser des informations ne figurant pas sur les sondes par défaut.

Enfin il faut prendre connaissance du dimensionnement-paramétrage de la machine : taille de la Swap, RAM, des paramétrages TCP, des configurations des cartes réseaux…).

✓ Conduite à tenir si la collecte des métriques systèmes n'a pas été possible

- Essayer de voir si les données des sondes d'outils métiers de supervision ou de reporting autre sont disponibles ; il s'agit chez EDF des logiciels Patrol, Omnivision et Sysload.

- Mettre en place les sondes sur la machine si la dégradation de performance peut de nouveau être observée, en effet il est assez délicat de diagnostiquer une dégradation de performance si celle-ci ne s'est produite qu'une seule fois en absence de métrique, est- elle reproductible? Ce n'est parfois pas le cas mais en général les problèmes de performance impliquant le paramétrage système ne se résolvent pas d'eux-mêmes. Il y a une forte prédiction à pouvoir ré-observer le problème à certaines périodes bien identifiées (Ndt : tâches planifiées).

[10] On appelle aussi ce type de profil standard : la Baseline

Si aucun des points précités ne peut être remplis, on peut analyser les fichiers journaux du système (journaux d'erreurs, de messages) et tenter de déceler ce qui semble anormal.

Ne pas oublier également que les journaux d'informations sont «périssables[11]» et ne sont présents que durant une période : ces derniers peuvent être volumineux (représenter parfois plusieurs centaines de mégaoctets) et de ce fait doivent être filtrés/parsés.

1.3.3 Analyse et recherche de la source du problème

Pour un fichier journal, il suffit de l'éditer et de traquer les erreurs ou messages anormaux ; pour les fichiers des métriques systèmes, de les mettre en forme graphiquement si ces fichiers sont des relevés de points, afin de mieux pouvoir juger visuellement le comportement des différents composants.

L'analyse doit se faire notamment en ayant la possibilité de comparer le comportement avec une situation dite nominale et a priori normale («Baseline»). Cette notion de normalité est une des clés de l'analyse de la performance, une situation de normalité n'est pas forcément une situation optimale. Cette analyse doit aussi tenir compte de l'historique de mise à jour de la machine (passage de correctifs, ajout de matériel…) et de la configuration.

1.3.4 Formulation des objectifs d'amélioration de performance et rédaction

Le rapport d'audit fait suite à l'analyse et doit être suffisamment détaillé, agrémenté de graphiques et de métriques. Lorsqu'un audit est effectué, un certain nombre de métriques ou de remarques sont écrites et rapportées. L'auditeur par ailleurs ne fait pas figurer nécessairement tous les points qui ont été vérifiés ou analysés, n'est écrit que ce qui est pertinent ou nécessaire. L'audit de performance est au final une synthèse intelligible pour le lecteur qui ne dispose pas forcement de tout le bagage technique pour comprendre tous les aspects, de ce qui a été vu, remarqué et à préconiser pour améliorer la performance.

La complexité sous-jacente en termes d'observation et de recherche est en quelque sorte masquée au lecteur et commanditaire de l'audit.

La thématique de la problématique de performance est souvent une affaire de compromis car elle nécessite de modifier certains paramètres systèmes qui peuvent influencer d'autres éléments de manière parfois antagoniste et pénalisante. Ceci est particulièrement vrai lorsque la machine héberge plusieurs progiciels et/ou applications.

De l'analyse faite précédemment, on peut aussi proposer d'autres pistes d'améliorations de manière marginale et complémentaire, ce qui laisse encore des optimisations possibles. Cependant il ne faut pas tomber dans le piège de l'optimisation à tout prix, en clair ne pas optimiser ce qui fonctionne *a priori* correctement comme le spécifie l'adage "le mieux est souvent l'ennemi du bien.".

Il est extrêmement fastidieux de rendre une machine complètement optimale. A l'instar de ce qui se passe dans d'autres domaines, on subit souvent la loi des rendements décroissants

[11] Les fichiers journaux ne gardent les données que une sur une fenêtre de temps glissant, on appelle cette technique la rotation de log. Des sauvegardes régulières peuvent résoudre ce problème

surtout dans le cas de surdimensionnement de mémoire ou processeurs et optimiser (et donc passer du temps) ce qui n'apporte un gain minime, est à proscrire.

✓ Test des solutions proposées avant la mise en application

Dans certains cas les recommandations doivent de préférence et si possible être testées sur une autre machine si celles-ci concernent une machine de production avec un environnement similaire. Souvent les applications mises en production sont constituées de triplet de machines (test/pré-production/production).

Généralement la machine de pré-production est identique architecturalement à la machine de production ou se doit de l'être et donc constitue idéalement un bon « bac à sable ».

1.3.5 Reformulation de recommandations

Cette étape optionnelle est effectuée si un suivi post recommandation est demandé ce qui peut être le cas si le tuning se fait par étapes ou par changements séquentiels de certains paramètres. Le processus d'optimisation est parfois itératif, ce qui nécessite de repasser l'Etape 2 (cf :1.3.2).

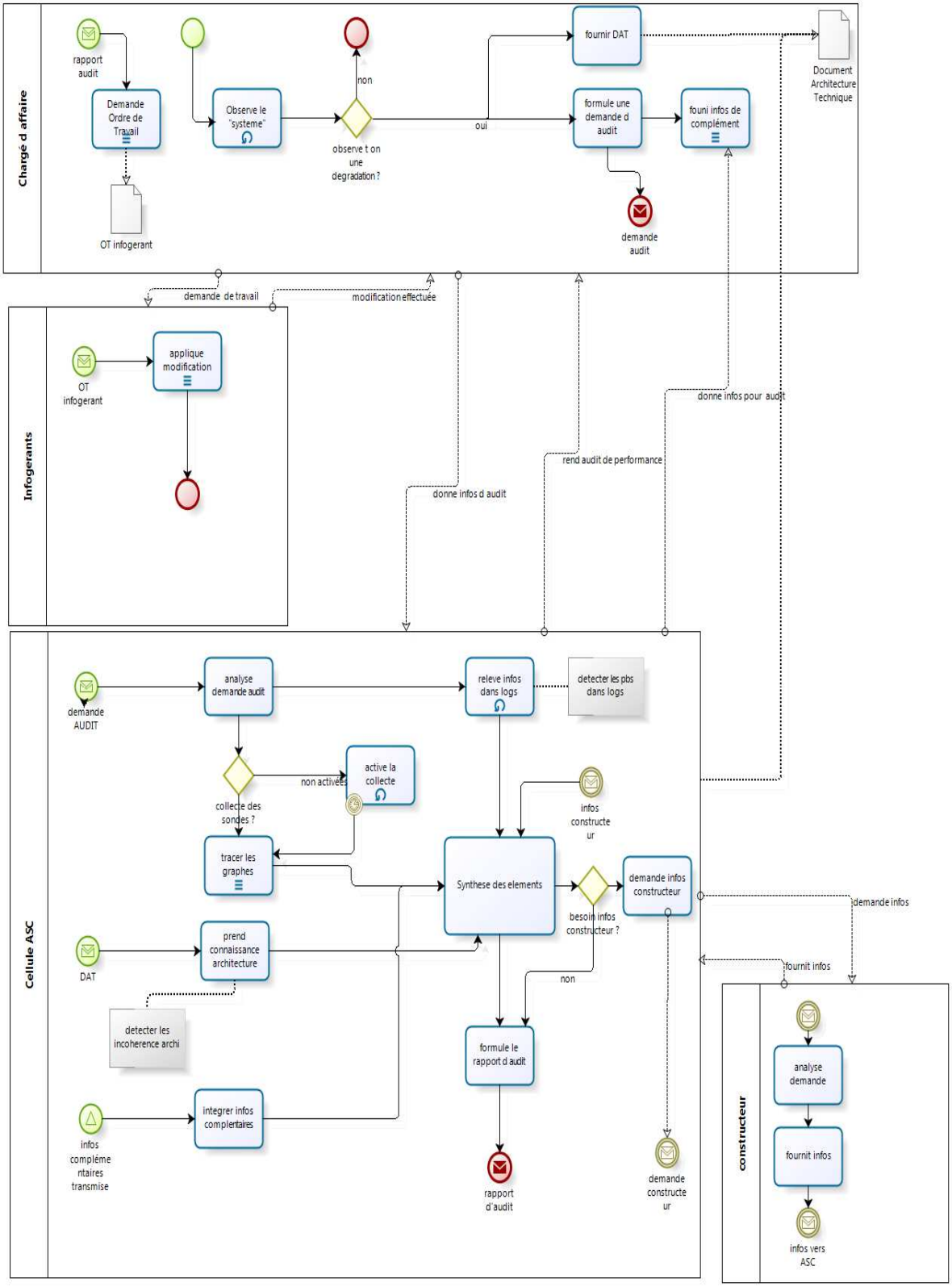
1.3.6 Fermeture de la demande de performance

Lorsque le document d'audit de performance est remis, un certain nombre de graphiques et de résultats de commandes sont émis : il se peut effectivement que les recommandations indiquées soient adéquates à la première itération et ne nécessitent pas un nouveau tour de boucle.

Le document est archivé au sein d'une base de document ainsi que toute la correspondance d'emails (principe de traçabilité).

Le problème peut être considéré comme résolu lorsque le demandeur ne donne pas de retour négatif ou communique que les préconisations ont pu résoudre les problèmes.

✓ Schéma BPMN complet du processus des audits de performances



1.4Etude de quelques bonnes pratiques existantes à EDF

Nombre de bonnes pratiques chez EDF se rapprochent d'ITIL, nous allons en détailler quelques-unes, certaines ont déjà été évoquées.

1.4.1 Journaliser l'activité de la machine

Ne sachant pas à priori quand le problème de performances peut survenir, il est nécessaire de journaliser l'activité de la machine afin de fournir une analyse *a posteriori* du problème de performance si celui-ci n'a pu être observé une seule fois ou quelques fois dans le passé. D'autre part il est préférable de connaître le profil de charge type et normal (la Baseline). Avoir une Baseline permet de discriminer les éléments de données qui ne semblent pas cohérents et normaux et de savoir ce qu'est le « bruit » sur une période ou des anomalies ont été constatées.

Dans le cadre d'EDF, l'historisation des métriques systèmes a été rendue possible par des scripts en exécution journalière sur les machines UNIX.

1.4.2 La gestion des capacités

Parler de la gestion des capacités ou du capacity planning n'est pas hors de propos puisqu'en définitive c'est une approche similaire au processus d'optimisation à savoir : définir et connaître les composants systèmes, le dimensionnement adéquat des ressources, afin de répondre aux besoins présents ou futurs pour une machine en rapport avec des exigences et contraintes métiers. Cette architecture devra donc répondre à des besoins de temps de réponse et de débit ou de capacité de traitement en adéquation avec les attentes du projet. C'est défini dans ITIL comme le sous-processus de la Gestion de la capacité des Services (SCM).

Une différence notable est que l'optimisation de performances est un moyen de pallier à un sous dimensionnement flagrant ou à une mauvaise allocation des ressources par le moyen de modifications sur le système, c'est en ce sens une des résultantes d'un « mauvais » capacity planning. Le capacity planning est en ce sens plus prospectif, et plutôt une tache d'observation et de constatations des besoins : de ce fait, une approche plus architecturale.

Une des différences réside dans le moment où les mener, à savoir que la capacity planning se fait avant ou à l'assemblage et la définition du système lorsque les besoins sont connus (NB : processus ITIL de la Gestion de la capacité des ressources – RCM).

L'audit de performance du système se fait lorsque le système est déjà défini, assemblé et *a priori* instancié par une dégradation de performance dans une majeure partie des cas. Ainsi un bon capacity planning initial réduit les besoins ultérieurs de tuning.

Les optimisations sont donc mises en place par des actions correctives et l'une des conclusions dans les audits peut être la remise en cause du matériel, et de proposer éventuellement le passage sur un serveur avec des capacités plus importantes.

Faire du capacity planning peut se révéler être une tâche ardue et introduit plusieurs paramètres :

- Variance dans la définition des besoins.
- Difficulté d'évaluer les besoins futurs.
- Besoins des éditeurs éloignés des réalités.
- Difficulté d'extrapoler,
- Modélisation imparfaite.

L'expérience du concepteur dans ce type de problématique est également un facteur important dans la recherche de l'architecture finale appropriée.

1.4.3 Les Benchmarks

L'objet des Benchmarks est de pouvoir servir de base de comparaison et d'évaluation du matériel et des logiciels afin de mesurer la performance de l'application ou de la machine, et si le dimensionnement et le paramétrage peut respecter les critères attendus. Ce sont des batteries de test spécifiques ou non s'appuyant sur des programmes et des données, et pouvant servir en tant qu'outil de décision. A ce titre les Benchmarks ont la propriété d'être standardisés et répétables, propriété que n'ont pas forcément les cas pratiques. Les Benchmarks peuvent servir dans le développement logiciel afin d'identifier la régression de performance après des changements de code ou des améliorations.

Des points énoncés précédemment il s'avère qu'un Benchmark peut raisonnablement différer de l'environnement de production. Cela peut toutefois donner un ordre d'idée. Les Benchmarks sont réalisés conjointement avec le constructeur et des consultants chez EDF, tel a été le cas lors de la mise en place des machines virtualisées sous AIX. Cela a pour but de se rapprocher du cas nominal d'exploitation de la machine dans un cadre attendu.

Certains des Benchmarks métiers sont réalisés avec l'aide de la cellule ASC, où l'on peut être amené à commenter le comportement de la machine.

1.4.4 Mise en place des préconisations constructeur et veille technologique

Lors de la mise en place des architectures un certain nombre de recommandations peuvent être émises par les constructeurs *via* les documents de référence (les plus connus sont les RedPapers pour IBM, Blueprint pour SUN), des paramétrages spécifiques en fonction du matériel sous-jacent, des souches ou des logiciels partie prenant de la machine ; ces recommandations sont émises par les cellules de veille des constructeurs et dans le cadre de contrat de maintenance entre EDF et les constructeurs. Des représentants des constructeurs, experts techniques interviennent régulièrement chez EDF. Ils peuvent être sollicités lors de problèmes techniques généraux et anomalies et inciter à mettre en place un certain nombre de préconisations suite aux dysfonctionnements.

Au-delà des interventions des experts des constructeurs, les ingénieurs systèmes chez EDF sont également attentifs aux informations circulant. Ces implémentations doivent cependant suivre un circuit de changement de configuration (circuit RFC).

La gestion des changements se fait dans l'esprit de la démarche ITIL.

1.4.5 ITIL et son intégration dans l'activité d'ASC

EDF, afin d'optimiser le fonctionnement de ces processus métiers informatiques et de ses circuits, utilise ITIL : ITIL étant un code de bonnes pratiques pour la fourniture de services informatiques (aide, support, exploitation…) et non d'une méthode à proprement parler.

Enumérons rapidement les différents domaines où des bonnes pratiques sont déjà en place dans le périmètre d'ASC, la liste est bien sur non exhaustive.

- ✓ Gestion des incidents
  - ➢ Escalade technique vers le constructeur
  - ➢ Suivi du cycle de vie des incidents, géré chez EDF par l'outil Peregrine [12].

- ✓ Gestion des changements
  - ➢ RFC (Request for Change) soumis à l'approbation d'un comité.
  - ➢ Vérifier que les objectifs ont été atteints.

- ✓ Gestion des configurations
  - ➢ Mise en place d'archivage des audits.
  - ➢ Base des bonnes pratiques pour l'administration des machines et le développement d'objets d'infrastructures (RDE pour Référentiel de Développement et d'Exploitation).
  - ➢ Les informations relatives aux équipements informatiques en exploitation ou en stock sont inventoriées par une IHM intranet (cf:Inventiv pour Inventaire Tivoli)
  - ➢ Les configurations des machines sont normalement identiques aux prescriptions du DAT [13] lorsque ceux-ci sont à jour. Les DAT intègrent notamment les schémas des flux et des choix architecturaux.

- ✓ Gestion des problèmes
  - ➢ Prévention de la récurrence des incidents, ces incidents sont débattus en CTO [14] avec des personnes de la cellule.
  - ➢ Capitalisation par le retour d'expérience.

- ✓ Gestion de la capacité

[12] Progiciel de HP AssetCenter de prise d'appel d'incident, progiciel payant
[13] Document d'Architecture Technique
[14] Comité technique opérationnel.

- Au quotidien le capacity planning (surveillance et anticipation des performances globale du système) est opéré par les chargés d'applications par le biais du progiciel Omnivision et les infogérants plus marginalement.

✓ Gestion Financière des services informatiques

- Visibilité sur les coûts informatiques

Sur la prestation de l'équipe, cela s'est traduit par une refacturation en interne des audits de performances aux équipes demandeuses, un des effets de cette mesure a été la disparition partielles des demandes faites à mauvaises escient, ce ticket d'entrée étant rédhibitoire pour une supputation des commanditaires d'audits non avérée et non critique.

La finalité est de pouvoir estimer et maîtriser les coûts liés à un service notamment en temps jour/homme pour la fourniture d'audit et de support.

L'équipe ASC fournit un service d'audit à plusieurs entités; il est donc normal de pouvoir marquer les flux financiers entre les entités et d'éviter des comportements abusifs de certaines entités découlant de la gratuité.

In fine le mode de fonctionnement de la cellule d'audit support est intégré dans les divers domaines ITIL et permet d'en accroître l'efficacité opérationnelle et fonctionnelle, l'inconvénient essentiel de ITIL est un alourdissement bureaucratique et la multiplicité des couches d'acteurs qui donnent une certaine inertie au système de gestion de traitement des incidents et de la gestion des changements, ceci fragilise le traitement des cas les plus urgents.

1.4.6 Résumé de l'existentiel en entreprise sur le thème de la performance

Les bonnes pratiques vues précédemment n'ont pas pour but de traiter ce thème mais d'être une politique d'évitement et d'une meilleure gestion du processus entreprise de la problématique de performance. Énumérons rapidement ce qui était en place pour le traitement de la performance au sein d'EDF lors de mon arrivée :

- Logiciel de capacity planning Omnivision [15].
- Sonde de collecte de données présentes sur les machines et macro Excel de traitement.
- Quelques documents d'études de Benchmark réalisés par les études
- Document de mise en forme des audits.

Au final on s'aperçoit que même si un document mettant en forme les audits de performance existe, il est à l'initiative de l'expert système de compléter ce document avec sa propre méthode d'analyse et connaissances ce qui peut varier en fonction des personnes.

Les connaissances et méthodes d'analyse liées au traitement de la problématique de la dégradation de performance se diffusaient essentiellement jusqu'à présent oralement entre les différents experts. L'archivage des audits de performance reste la seule sortie écrite sur le traitement de performance pour un système.

L'ambition donc de ce mémoire est aussi d'apporter une réponse à ce besoin de l'entreprise, de permettre une transmission de connaissances écrite au sein de l'équipe, ce qui explique

[15] Une description de ce produit y est faite dans la 3eme partie

aussi son fort contenu technique dans la deuxième partie et de plus d'apporter une réflexion liée aux nouvelles implémentations technologiques mises en place chez EDF depuis quelques années.

2 Diagnostic d'un problème de performance

2.1Introduction

Analyser un problème de performance suppose de prendre en considération un certain nombre d'éléments et ne se résume pas uniquement à l'observation de graphiques afférents à l'activité système et faire des supputations uniquement à partir de cela.

Pour rappel un problème de performance s'analyse entre autres à partir :

- Des symptômes perçus au niveau applicatif énoncés et du «feedback» du C.A et des utilisateurs.
- Des problèmes d'origine fonctionnelle.
- De l'analyse de tous les fichiers journaux et d'erreurs.
- Des remontées d'information éventuelles d'outils tierces.
- De la connaissance des derniers changements opérés sur la machine et de la configuration tels que des mises à jour logicielles …
- Des notes des constructeurs ou d'éditeurs sur des problèmes afférents à la performance diffusés régulièrement.
- De la connaissance sommaire des différentes briques logicielles présentes sur le serveur et de leur paramétrage «système».
- Des nuisances potentielles exogènes au système dans le SI, notamment la partie réseau et baies de disque fortement impactant le fonctionnement de la machine.

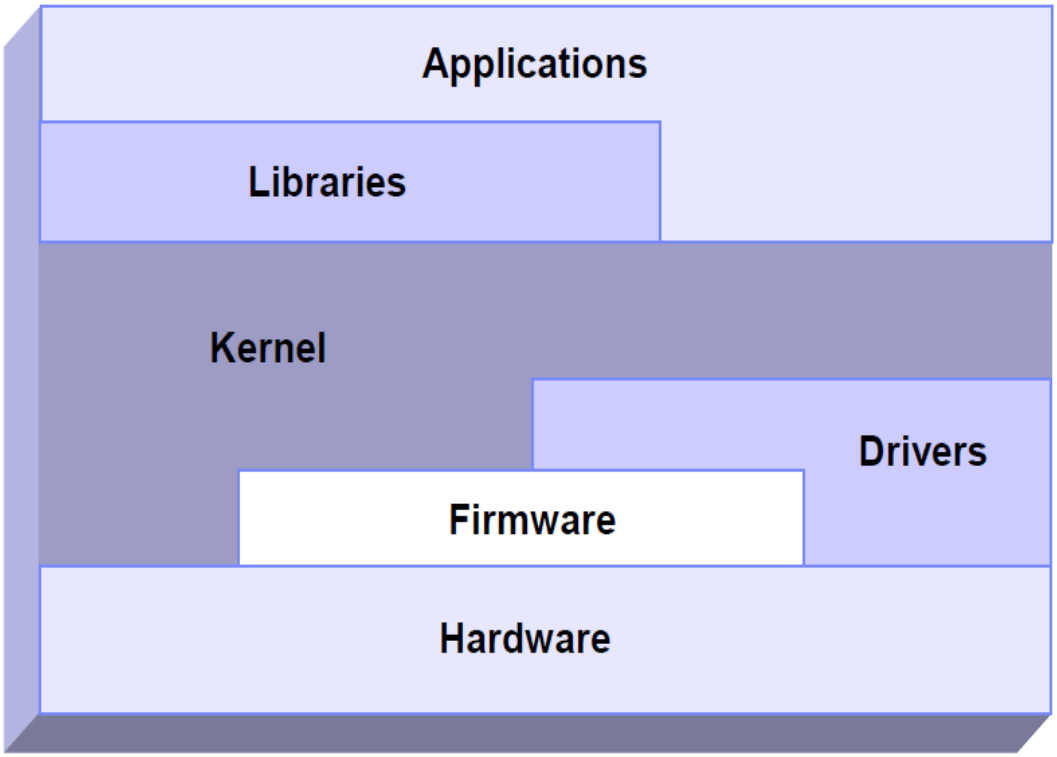
Pour résumer l'analyse des données provenant des éléments systèmes n'est qu'un des éléments d'analyse et la considération des éléments précités est nécessaire.

Ce chapitre concernera donc de cette analyse de données provenant des éléments réseau, mémoire, CPU et disque, et d'expliquer de manière sommaire le fonctionnement et les fondamentaux inhérents à chacun de ces éléments sur les systèmes UNIX présents chez EDF.

Etant donné la complexité et l'exhaustivité des paramétrages propre à chacun, le présent document sera volontairement simplificateur. Il faudrait en effet plusieurs centaines de pages pour discuter de ce sujet amplement. Optimiser le système suppose le paramétrage logiciel de ce qui se situe entre l'application (du ressort des équipes projets ou outillage) et la partie proprement matérielle. On peut aussi proposer une amélioration éventuelle des éléments matériels qui se concrétisent par un achat, d'un changement ou une revue architecturale matérielle.

Le système d'exploitation s'interface plus ou moins directement avec le noyau (kernel), qui se fait l'interface avec toutes les autres « briques » du système.

Schéma d'interaction des différentes couches dans la problématique de performance[16]



Si le système d'exploitation interagit sur les différents éléments matériels, c'est le noyau qui en a le contrôle; cela concerne quatre familles d'éléments: les processeurs, la mémoire, la gestion des disques, les cartes réseaux qui représentent ce que l'on appelle l'activité du serveur.

Cette activité peut être mesurée par un certain nombre de données et statistiques provenant donc du noyau. On dispose donc d'un grand nombre de métriques dont il faut pouvoir tirer les informations les plus intéressantes et pertinentes, chaque objet physique appartenant à l'une des classes ci-dessus fourni individuellement de nombreuses métriques variantes dans le temps.

2.2 Plan succinct de chaque sous-parties

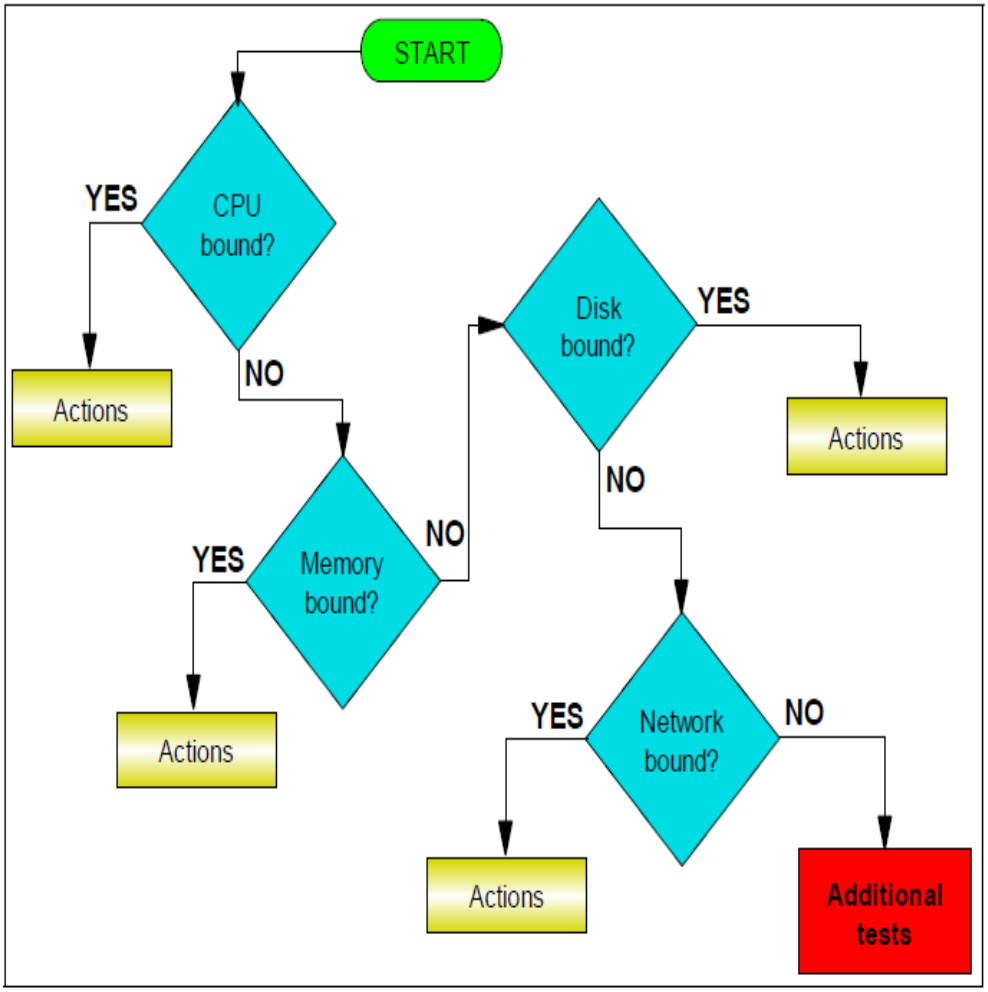
Cette partie se découpera en 4 sous-parties afférentes aux éléments physiques précités.

Une explication simplifiée de l'élément sera faite en introduisant le lexique propre à l'élément, ensuite la mise en œuvre pour observer la performance les métriques et les seuils importants.

Enfin ce qui est préconisé de manière générale en terme d'optimisation, et ce qui a été fait au sein d'EDF (toutes les optimisations possibles ne sont pas forcément mises en œuvre).

[16] Schéma tiré du RedPaper IBM Tuning des systèmes Linux

✓ Synoptique général de l'analyse de performance au niveau des éléments du système [17]:



[17] Schéma tiré du RedBook IBM tuning AIX 5 L

2.3Les processeurs

2.3.1 Généralités

Analyser les processeurs en premier lieu n'est pas un choix aléatoire. Les processeurs sont effectivement au cœur du fonctionnement de la machine et aussi des moins influencés par des contentions exogènes au serveur, on entend souvent dire que la machine «rame»; cela signifie souvent implicitement que les problèmes proviennent des processeurs de la machine: cette vulgarisation de langage est certes quelque peu erronée mais dénote un fait que j'ai moi-même pu constater: une grande partie des problèmes de performance sont bien liés à l'activité des processeurs bien qu'en moyenne un serveur utilise 12% du temps processeur.

Par ailleurs c'est un élément qui malgré la complexité sous-jacente est plus facile à comprendre pour l'analyse de la performance et dont les métriques sont les plus faciles à interpréter.

Le processeur est l'élément le plus rapide du système, une contention y est toujours significative. De même, régler une contention au niveau de la CPU peut créer une nouvelle contention portant sur un autre élément (effet de déverrouillage de la contrainte la plus forte).

Le processeur est aussi une des parties les plus onéreuses des éléments avec la mémoire vive: ainsi, pouvoir diagnostiquer un problème de sous-dimensionnement CPU assez rapidement peut être pertinent si on doit faire une commande de processeurs additionnels auprès du fabriquant; optimiser l'utilisation de la CPU peut aussi se révéler judicieux d'autant plus si elles ne sont plus fabriquées et donc rares à obtenir pour des matériels anciens.

✓ Points préalables à la compréhension des processeurs et processus

Le rôle du processeur est de manipuler des données et des instructions provenant de la mémoire centrale vive par le biais de ses propres registres et caches; le processeur applique un certain nombre d'opérations élémentaires à ces données par le traitement des instructions (notamment contenues dans un programme) converties en langage machine puis réaffecte ces données résultantes vers les registres mémoires ou la mémoire centrale.

La métrique la plus triviale désignée pour les capacités de traitement du processeur est exprimée en Mégahertz (Mhz) ou Gigahertz, 1 Mhz est égal à 1 million de cycle de CPU par seconde. Une instruction est ainsi traitée en un ou plusieurs cycles d'horloge processeur.

Cependant, l'architecture et la technologie du processeur sont aussi importantes que cette vitesse de calcul. Notamment si la technologie est de type CISC ou RISC, et des types d'implémentations matérielles ...

Le temps d'exécution d'un programme peut être estimé aux nombres d'instructions qu'il contient, pondéré par le nombre de cycles utiles à chaque instruction rapporté à la fréquence du CPU. Le nombre élémentaire d'opérations requises pour exécuter un programme est une résultante de la compilation. Le nombre de cycles nécessaires à une instruction est lié à sa complexité sous-jacente.

Un processus est une entité contrôlée par l'OS pour son attribution de ressources systèmes; un processus est initié par une commande, un programme ou un autre processus, et possède un environnement d'exécution, un espace d'adressage, et un identifiant unique attribué par l'OS lors de son exécution.

Les threads sont des files d'exécutions séquentielles et un processus peut être composé de un ou plusieurs threads. Les threads, ou processus légers, ont la particularité de partager leurs espaces d'adressage : ils sont donc moins consommateurs en terme d'espace mémoire, et facilitent la commutation de contexte.

Un processus a, vis-à-vis du système, plusieurs états possibles :

✓ Etats d'un processus via à vis de l'ordonnanceur

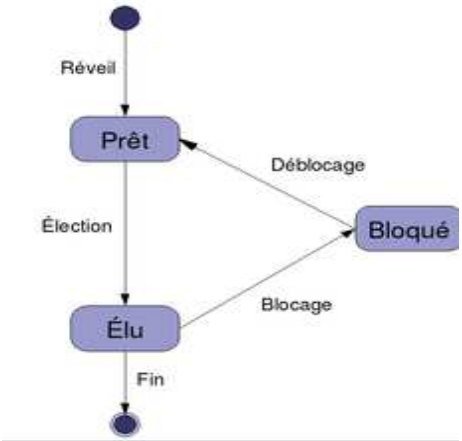


Diagramme d'état d'un processus simple.

Les CPU vont pouvoir exécuter séquentiellement et itérativement les processus, durant un quantum de temps, *via* des files d'attentes. L'ordre d'exécution sera géré par l'ordonnanceur [18]du système en fonction de la priorité et de l'état d'éligibilité à l'exécution. Les processus ayant un niveau d'exécution bas [19] sont traités en priorité. Cette priorité dépend d'un certain nombre d'éléments: la police d'exécution, la priorité affectée à l'initialisation du processus, la pénalité d'usage à la CPU; autrement dit, les priorités d'exécution sont dynamiques et favorisent les processus les plus critiques.

Il est possible de modifier la politique et les tables de l'ordonnanceur grâce à des commandes système.[20.] La priorité des processus se modifie en général *via* les commandes ***nice*** et ***renice*** sous Unix.

De nos jours, un processeur a souvent plusieurs cœurs d'exécution, et implémente des technologies de parallélisme d'exécution modifiant la terminologie d'appellation entre processeurs physiques, logiques, et virtuels.

[18] L'ordonnanceur est aussi appelé scheduler.
[19] Sous Unix, plus le niveau priorité est bas, plus le processus est prioritaire : ce n'est pas le cas sous d'autres type d'OS.
[20] Sous AIX ,il s'agit des commandes ***schedo*** et ***schedtune*** ; sous SOLARIS : ***dispadm*** *;* voir implémenter le FSS (Fair share scheduler) qui permet notamment dans le cas des zones de pondéré les ressources attribuées aux zones.

✓ Connaître la version d'OS et les informations relatives à la CPU

Pour répondre à cette question, voici quelques commandes les plus utilisées :

Pour connaître le modèle de la machine et type de système d'exploitation :

Solaris/Linux : ***#uname –a*** pour le modèle ; et/ou ***oslevel –r*** sous AIX

Pour obtenir les informations sur les processeurs :

Solaris : ***/usr/sbin/psrinfo –vp***

AIX : ***lsattr –El proc*** ou ***lsdev –Cc processor***

Linux : ***cat*** `/proc/cpuinfo`

2.3.2 Différents outils triviaux pour superviser l'activité CPU

La supervision sommaire de l'activité CPU et mémoire sous Unix se fait en général par les commandes ***vmstat***, ou ***sar*** et ***mpstat***. Ces commandes prennent notamment en argument la fréquence d'exécution de la commande[21].

Exemple sous AIX de retour de la commande ***vmstat*** :

```
# vmstat 5
System configuration: lcpu=2 mem=3920MB

kthr    memory              page              faults        cpu
----- ----------- ------------------------ ------------  -----------
r  b   avm   fre   re  pi  po  fr   sr  cy  in   sy  cs   us sy id wa
9  0  4200  2746   0   0   0   0    0   0   3  198  69   70 30  0  0      0
4  7  4200  2746   0   0   0   0    0   0   3   33  66   67 31  2  0      0
2  6  4200  2746   0   0   0   0    0   0   2   33  68   65 34  1  0      0
3  9  4200  2746   0   0   0   0    0   0  80  306 100   80 20  0  1      0
2  7  4200  2746   0   0   0   0    0   0   1   20  68   80 20  0  0      0
```

Les champs les plus scrutés sont :

- *r*: processus éligibles à l'exécution ou en cours d'exécution (appelé aussi « run queue »)

- *b*: processus à l'état bloqué (en attente de ressources)

- *us* et *sys* : consommation utilisateur (exécution du propre code de l'application ou des librairies partagées) et système

- *cs* (context switch) : nombre de permutations de processus à la seconde

- *idle* : temps libre et *wa* : en attente d'I/O (parfois appelé *wait i/o ou wio*), il est à présent considéré comme un pseudo temps libre du processeur. Sous Solaris, le *wio* n'est plus rapporté car il manque de pertinence.

[21] Attention de ne pas mettre une valeur d'échantillonnage trop basse (< 5 sec) à cause des risques d'Overhead dans le cadre de la collecte des sondes , l'intervalle est de 60 secondes.

En ce qui concerne la supervision en temps réel des processus en exécution, il existe des commandes plus spécifiques au suivi des activités des processus les plus consommateurs : ***prstat*** et ***top*** sous Solaris, ***topas*** sous AIX, et ***top*** sous Linux.

Exemple de sortie de ***prstat*** sous une machine Solaris 8 :

```
  PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
   67 root     3968K 2912K sleep   33    0   0:00.00 0.1% picld/7
16568 oracle   1115M 8496K sleep   53    2   0:01.10 0.1% oracle/15
22494 admsip   1920K 1592K cpu0    58    0   0:00.00 0.0% prstat/1
 7842 oracle   1104M   12M sleep   52    2   0:02.48 0.0% oracle/1
16570 oracle   1109M   10M sleep   53    2   0:00.17 0.0% oracle/19
    1 root      856K  248K sleep   58    0   0:11.43 0.0% init/1
  219 root     2008K 1256K sleep   58    0   0:01.40 0.0% cron/1
16566 oracle   1114M   14M sleep   53    2   0:00.53 0.0% oracle/258
  384 root     2824K 2016K sleep   58    0   0:00.00 0.0% mountd/4
  319 root     4328K 1656K sleep   59    0   0:00.00 0.0% sendmail/1

Total: 82 processes, 513 lwps, load averages: 0.13, 0.05, 0.05
```

On y remarque un certain nombre d'informations intéressantes, notamment l'état du processus; son identifiant (*PID*), son occupation en mémoire centrale (*RSS*) et mémoire virtuelle (*SIZE*), sa priorité, sa durée d'exécution, son occupation processeur, la commande liée au processus et le nombre de processus légers (*NLWP*). Certains processus sont mono-thread, comme par exemple : `sendmail/1`.

Nous pouvons obtenir les informations sur l'activité de chaque processeur par la commande ***mpstat*** (***vmstat*** offrant une vision agrégée de l'activité CPU).

2.3.3 Les signes basiques d'une contention CPU

✓ Avec la commande vmstat :

- Les processus en état d'exécution (r) doivent être normalement inférieurs à ceux en état bloqués (b) auquel cas on peut suspecter un blocage de processus
- Ratio du nombre de processus en exécution sur le nombre de CPU virtuelles (Cœur) supérieur à 2 (système chargé), si supérieur à 4 (système surchargé)
- Occupation du processeur à plus de 80 % (système très chargé) supérieure à 15 min
- Occupation de la CPU à 99% (saturation du système) supérieure à 2-3 min

Le critère de l'occupation de la CPU est surtout pertinent sur des machines physiques sans partage inter-partition de CPU. Un ralentissement peut être ressenti à partir en général de 80% pour 1 CPU. L'activité système représente idéalement moins de la moitié de l'activité utilisateur *(% sys/% us*).

Une activité système peut être importante dans le cadre de longues files d'attente de threads, ce qui a pour effet de provoquer une importante commutation de threads, pouvant provoquer un cercle pernicieux puisque l'activité système, en mode noyau, est protégé et prioritaire. De ce fait, ce type d'activité ne doit pas être prépondérant et être surveillé: une forte consommation système est en général de mauvais augures pour la performance du système.

Pour éviter la commutation de contexte, les threads s'exécutent avec une affinité de processeur.

✓ Avec la commande sar

La commande ***sar*** est utile à un plus d'un titre, car elle permet de relever des activités sur tous les éléments matériels et logiques à intervalles réguliers (par défaut, toutes les 10 min) et de les conserver dans un répertoire de l'arborescence système. Cette commande est installée à partir de paquetages. Les données sont souvent présentes sous les souches Linux d'EDF.

✓ Avec le commande mpstat

Cette commande est utile pour déterminer les asymétries d'utilisations instantanées. Elle représente l'activité par processeur.

Exemple d'une capture instantanée avec ***mpstat***

CPU	minf	mjf	xcal	intr	ithr	**csw**	**icsw**	**migr**	**smtx**	srw	syscl	usr	sys	wt	idl
0	0	0	7	10	6	38	0	4	0	0	23	0	0	0	100
1	0	0	7	11	6	8	0	0	0	0	12	0	0	0	100
2	0	0	19	11	6	44	0	3	0	0	168	0	0	0	100
3	0	0	58	85	81	43	0	1	0	0	32	0	0	0	100
4	**431**	**0**	**2**	**21**	**7**	**0**	**10**	**0**	**0**	**0**	**362449**	**28**	**72**	**0**	**0**
5	0	0	14	11	7	105	0	11	7	0	40	0	0	0	99
6	0	0	64	49	6	74	0	22	4	0	90	1	0	0	99
7	0	0	35	307	92	82	0	9	5	0	37	0	0	0	100
16	0	0	7	10	7	66	0	1	0	0	81	0	0	0	100
17	0	0	13	10	5	60	0	1	0	0	33	0	0	0	100
18	0	0	2	10	5	44	0	1	0	0	33	0	0	0	100
19	0	0	2	15	10	14	0	0	0	0	35	0	0	0	100
20	0	0	1	10	5	43	0	20	4	0	75	0	1	0	99
21	0	0	7	17	12	22	0	3	1	0	14	0	0	0	100
22	0	0	1	10	5	54	0	2	5	0	49	0	0	0	99
23	0	0	7	10	5	63	0	6	1	0	26	0	0	0	100

Une CPU a sa propre file d'attente, une seule CPU saturée par une application mono-threadée peut montrer des contentions systèmes alors que globalement l'activité CPU semble acceptable (effet dilutif des autres CPU "libres"): c'est notamment le cas pour certaines « vieilles » applications, et des tâches de compression souvent mono-threadées gourmandes en utilisation processeur.

Une contention au niveau d'une CPU peut être aussi le fait d'un bind [22]entre un processus et une unité d'exécution. Des commandes sous AIX comme ***topas***, ***nmon*** ou ***lparstat*** remontent quant à elles des informations un peu plus prolixes, notamment dans des environnements partitionnés.

✓ Problème de la saturation CPU pour des processus monothréadé

[22] Le terme bind *(an)* consiste à attacher un processus directement à un processeur nommé pour son exécution. Ce qui peut être une forme de limitation en terme d'appropriation de ressources CPU, mais gêne aussi les politiques d'ordonnancement du système, c'est à manipuler avec précaution.

Il est assez délicat de distinguer *a posteriori* par ***mpstat*** des saturations CPU liées à un processus monothreadé, puisque l'exécution de ce processus se fera à tour de rôle sur différents processeurs: ce qui rendra l'observation graphique par processeur ou globale moins pertinente.

Le diagnostic de ce type de saturation se fait *via* une supervision en temps réel, afin de déceler l'exécution vers une autre CPU qui sera saturée durant un peu de temps. Graphiquement, pour la mise en forme de ***vmstat***, cela s'interprète essentiellement par des paliers correspondant au ratio 1 CPU sur l'ensemble des CPU (un exemple plus illustratif est proposé en Annexe).

Il faut donc être extrêmement attentif à l'aspect monothread/multlithreadé sur des systèmes multiprocesseurs: aussi le processus peut être attaché à un processeur (bind) de manière volontaire. Lorsqu'une CPU est saturée par un processus monothread la seule solution est de redévelopper le processus en mode multi-threading ou de redévelopper le code , ce qui n'est parfois pas possible . Dans ce cas, il est toujours possible de migrer vers des processeurs plus performants, ce qui est dispendieux.

2.3.4 Types d'actions possibles sur les contentions liées à la CPU

- Repartir la charge sur différentes périodes d'exécution, en évitant de faire tourner simultanément des applications très fortement consommatrices
- Changer les paramètres du scheduler : on peut également utiliser les commandes de priorisation de processus ***nice***/***renice***
- Utiliser les technologies les plus appropriées en fonction des applications : certaines apportent un gain de performance, notamment dans le multithreading
- Déterminer et analyser les processus les plus consommateurs, arrêter[23] les processus qui n'ont aucune pertinence (ceci englobe également les services)
- Redémarrer régulièrement la machine afin notamment de supprimer les processus zombies
- Optimiser les applications avec leur propre outil interne et si besoin réduire la charge
- Sérialiser les traitements s'ils ne peuvent s'exécuter de manière simultanée
- Proposer un accroissement des ressources CPU lorsque tous les points évoqués ci-dessus auront été examinés.
- Brider la consommation des processus par des fichiers systèmes[24]

Dans la pratique, chez EDF

Nous préconisons généralement :

- Un ré-ordonnancement ou une sérialisation de certaines tâches.

[23] Sous le shell Unix, la commande ***kill*** est utilisée pour interrompre volontairement un processus ou par l'arrêt de services liés aux démons.

[24] Nous verrons ces fichiers un peu plus tard dans le document (il s'agit des fichiers limits ou projects)

➢ Des propositions pour alléger les sollicitations des processus applicatifs, et tuner l'application consommatrice (ce qui est rendu possible par les équipes études).

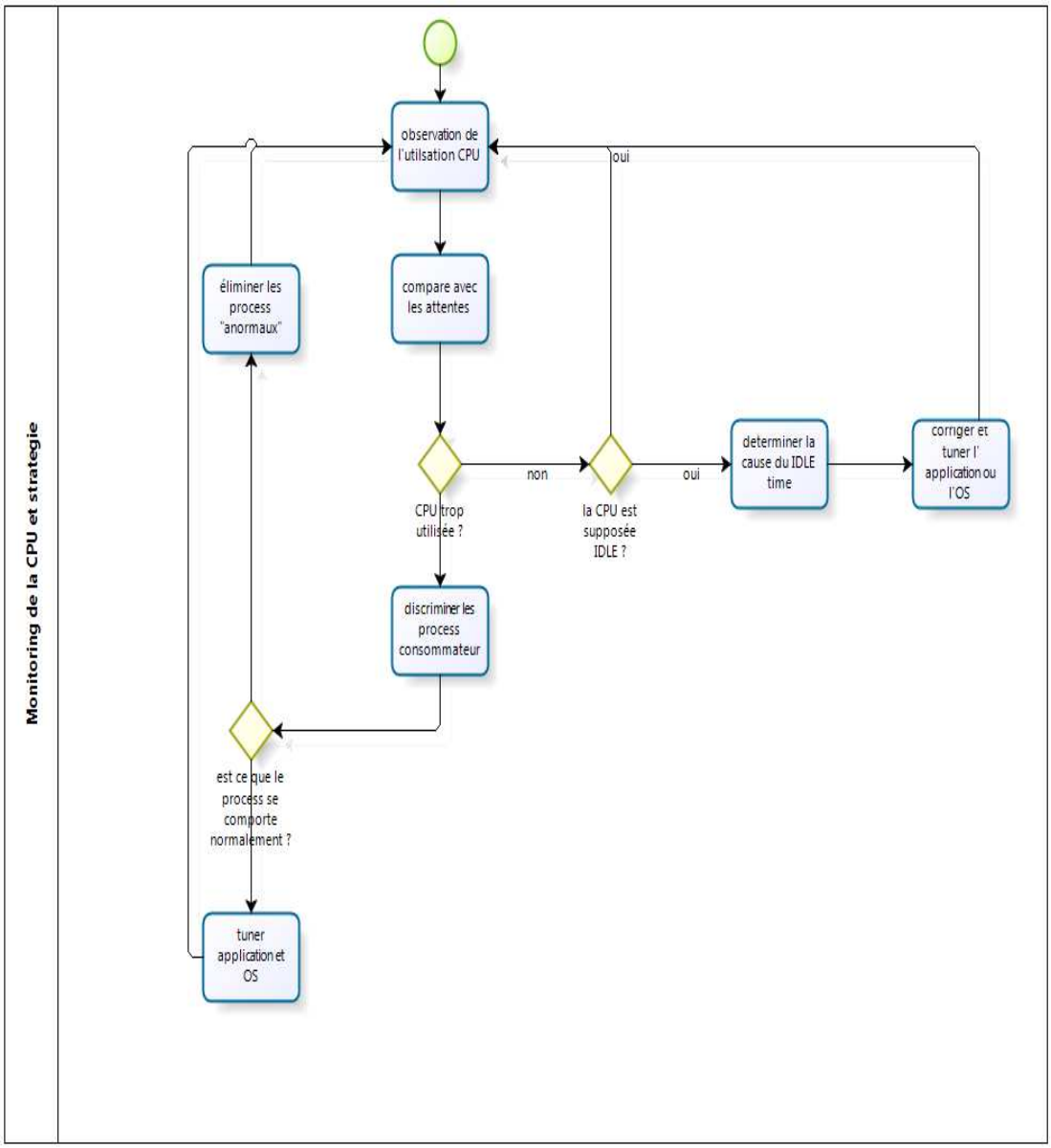
➢ Une migration vers des modèles CPU plus puissants, voire augmentation du nombre de CPU attribuables ;

➢ Opter vers une version multithreadée de l'application, si possible

➢ Arrêt des services inutiles (on allège la machine en termes de processus).

➢ Limiter la consommation CPU des processus par des fichiers systèmes

Synoptique de traitement basique pour la performance des CPU



2.4 Performance de la mémoire

2.4.1 Explication basique sur la gestion interne de la mémoire

Les saturations mémoires ont des effets notables sur les performances globales du système, et sa supervision est parfois plus essentielle que les performances CPU.

Cette contrainte sur la mémoire s'est notamment accentuée sur les quinze dernières années ou les cycles d'horloges des CPU, augmentant exponentiellement, ont largement supplanté la performance (les temps d'accès) de la mémoire. Au début des années 1980, le temps d'accès de la RAM était de 200 ns alors que le cycle d'horloge de la CPU était de 5 Mhz (200 ns). Dorénavant, l'horloge de la CPU est au moins de 2-3 GHz sur les modèles actuels alors que la RAM a un accès de quelques dizaines de nanosecondes. Plus clairement, la mémoire est apparue au fil du temps comme un goulot d'étranglement de plus en plus apparent.

Un système se doit, pour fonctionner, de disposer d'une quantité de mémoire suffisante : non seulement pour exécuter les applications métiers mais également pour faire tourner les routines du système.
Ainsi il convient de prendre en compte ce dont le système a besoin pour faire exécuter ces scripts/taches/applications métiers, et pour son fonctionnement nominal.

Quand on évoque la mémoire, on sous-entend deux types de mémoire : la mémoire dite physique qui utilise les capacités en barrettes de RAM de la machine, et la mémoire dite virtuelle qui permet en autre d'excéder la capacité d'adressage de la machine au-delà de sa limite physique RAM. Pour cela le système utilise la pagination et l'échange, moyens par lesquels il va distribuer la mémoire disponible.

La pagination est décrite de manière triviale comme un déplacement des segments mémoire entre la mémoire centrale vers un espace de stockage disque temporaire, ceci afin de libérer de la mémoire physique demandée par un autre processus.

De ce fait la pagination revêt une connotation péjorative ; mais elle est en définitive un mécanisme intéressant, et indispensable au bon fonctionnement du système d'exploitation ; car l'accès aux disques (lecture/écritures) est infiniment plus long que les accès mémoire.

La définition propre de la pagination est un découpage en granule de la mémoire adressable en quelques kilos octets, qui facilite la gestion des processus en mémoire virtuelle. Un processus est donc adressé sur des segments découpés par un nombre de pages.

Un processus a comme contrainte mémoire la taille de son image exécutable et la quantité de mémoire utilisée pour ses données.

L'espace d'adressage d'un processus (code, données et pile d'exécution) est constitué de l'ensemble des adresses auxquelles le processus a accès au cours de son exécution, et est une notion totalement indépendante de la mémoire physique sous-jacente. Cet ensemble d'adresses s'appelle espace d'adresses virtuelles.

La mémoire RAM est incontestablement coûteuse (même si les prix ont été beaucoup baissés ces derniers années), généralement tous les processus ne s'exécutent pas en même temps ; si ceux-ci demandaient une allocation mémoire de manière simultanée, la machine ne pourrait fournir en général suffisamment d'espace en mémoire centrale ; à l'instar des compagnies de transport, on va utiliser une politique de surbooking. Le système va donc tirer

parti de ce fait, en exploitant de l'espace disque sur lesquels un certain nombre de pages moins utilisées seront transférées si besoin.

Du point de vue de l'administrateur système, la mémoire virtuelle disponible est donc la concaténation de la mémoire physique centrale et des espaces d'échange (également appelés disque de swap). L'espace de pagination disque a une structure de fichier différente des autres espaces disques.

Les morceaux de l'image du processus sur le disque sont lus uniquement lorsqu'on en a besoin. Si celui-ci ne réside pas en mémoire centrale, il y a défaut de pages, il va donc être chargé en mémoire centrale depuis les disques[25].

Le système va donc opérer, par des algorithmes évolués, la gestion des pages qui devront être présentes en RAM afin que tous les processus puissent s'exécuter de manière correcte et concurrente, puisque la mémoire virtuelle donne l'impression que la machine dispose d'une capacité d'adressage supérieure à la capacité physique de la RAM.

Sans rentrer dans les détails, les techniques les plus courantes pour gérer les ressources mémoires sont :

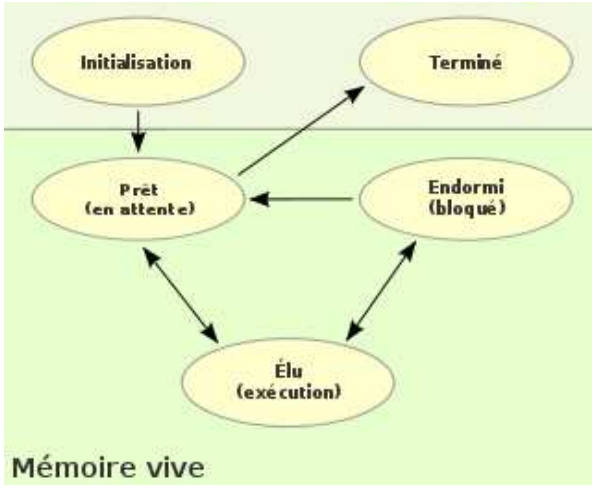
- la pagination à la demande,
- la protection de la page sur la copie sur écriture.
- la récupération de page.

La supervision des défauts de page est importante qui consiste à charger depuis les disques vers la mémoire centrale les pages manquantes. En effet, si le système passe son temps à gérer des défauts de page, c'est autant moins de temps disponible pour gérer l'exécution des processus.

D'une certaine manière la mémoire physique (RAM) essaie de contenir le maximum de pages liées aux processus actifs, et ne sont transférées sur l'espace de pagination disque que des pages mémoires bien particulières que nous verrons par la suite.

Un processus possède différents états en mémoire. Ces états sont retranscrits au sein des commandes de supervision de la mémoire :

✓ <u>États d'un processus en mémoire</u>



[25] Ce chargement en mémoire centrale est caractérisé physiquement par de la pagination en entrée (page-in ou pi).

2.4.2 Les problématiques mémoire en entreprise

Sur les trois systèmes UNIX exploités au sein du parc EDF (Solaris, Linux, AIX), les problématiques mémoire sont monnaie courante sous AIX en version inférieure à AIX 6.

Sous AIX, l'administrateur système a davantage de paramètres optimisables pour sa gestion de mémoire ; en outre, par défaut, les serveurs ne sont pas installés toujours avec les paramètres mémoires adéquats sur les versions anciennes.

Sur les 5 années effectuées au sein d'ASC chez EDF, je n'ai jamais changé un seul paramétrage mémoire sous Solaris (hormis des préconisations liées au dimensionnement de la taille mémoire RAM ou de swap disque), Solaris est un système relativement bien optimisé dans la gestion de mémoire; et très peu de fois pour Linux.

L'essentiel de l'optimisation de la mémoire sera donc consacrée à AIX et les deux autres OS seront évoqués de manière plus synthétique pour complétude.

2.4.3 Gestion de la mémoire, le cas de AIX

✓ Explications génériques et observations de l'activité mémoire

Pour AIX, la Virtual Memory est gérée par le VMM (Virtual Memory Manager) dont le but est de gérer les pages en mémoire, l'allocation des pages, et de résoudre les défauts de page : pour cela il doit maintenir une liste de pages libres allouables (cf :free list).

Tout processus qui se loge pour la première fois en mémoire centrale créé un défaut de page. Le VMM doit garantir un fonctionnement mémoire acceptable et de ce fait, il doit également minimiser le trafic I/O avec les espaces de pagination et de disques.

On surveille essentiellement la mémoire virtuelle via la commande ***vmstat*** [26] :

```
CJ007D3M@serveurA> vmstat -I 5

System configuration: lcpu=8 mem=8192MB ent=2.00

  kthr     memory              page              faults              cpu
-------- ----------- ------------------------ ------------ -----------------------
 r  b  p   avm   fre  fi  fo  pi  po  fr   sr  in   sy  cs us sy id wa    pc    ec
 7  2  0 1406680 17904 334 1250   0   0   0    0 807 73777 3403 57 16 25  2  1.49  74.4
 4  2  0 1406091 16037 343 984   0   0   0    0 1002 108125 3885 78 19  3  0  1.98  99.1
 3  1  0 1406145 12463 355 1099   0   0   0    0 1093 14760 4189 44 11 39  6  1.15  57.6
 2  1  0 1406150  8885 385 390   0   0   0    0 669 14050 3027 38 10 48  5  0.99  49.5
 6  1  0 1406195 100772 334 819   0   0 18765 86418 687 16699 3069 45 16 35  4  1.26  63.2
 5  1  0 1406194 94938 524 709   0   0   0    0 523 12603 2856 41 10 43  6  1.06  53.0
```

Interprétations et explications sommaires des différentes métriques ci-dessus:

Cartouche memory :

- *Avm* = active virtual memory (nombre de pages de mémoire virtuelle en cours d'utilisation ; les pages virtuelles sont assignées en mémoire)

[26] Pour Virtual Memory Statistique

- *Free* = Taille de la free List, ou nombre de pages allouables en mémoire centrale pour des nouveaux processus et pour résoudre des défauts de pages ; il est normal de voir cette valeur chuter lorsque les processus se chargent en mémoire: ces pages seront les premières à être évincées si besoin.

Cartouche pages :

Très communément la taille d'une page est de 4 Ko ou 8 Ko.

- *Pi* ou page-in représente un transfert de page de l'espace disque de pagination vers la mémoire physique.
- *Po* ou page-out, le mouvement inverse du *pi*: cette valeur est souvent proche de 0
- *Fi* et *Fo* sont quant à eux les pages in/out transitant entre la mémoire centrale et les file Systems. C'est notamment le cas pour les pages dites persistantes - c'est à dire adossées à un emplacement physique disque.
- *fr* = freed : pages libérées par seconde, et *sr* = scan rate : nombre de pages scannées par seconde

✓ Description des différentes pages résidentes en mémoire centrale

Avant d'entamer la description des principaux paramètres qui permettent de gérer la mémoire sous AIX, il faut définir les différents types de pages résidents au sein de la mémoire virtuelle[27].

La mémoire centrale est peuplée de 4 types d'espace mémoire :

Pinned Memory : pages toujours résidentes en mémoire (cf.« pinnée »), elle sont inéligibles pour le page out. Ces pages sont relatives au noyau, au processus Swapper, aux tampons des périphériques et des réseaux…

Persistentes pages : pages relatives au code du programme typiquement qui ont trait aux instructions ou des données. Ces pages sont persistantes car elles bénéficient d'un emplacement permanent sur un disque de données. Les pages de code d'un programme d'exécution ne sont conceptuellement qu'en mode lecture, ce qui explique également qu'elles ne peuvent être modifiées : elles ne sont donc pas en page-out vers le disque de swap.

Pages de travail (ou working pages) : pages utilisées temporairement par les processus : piles des processus, et régions de données liées aux processus en exécution telles que les variables, segment code binaire du noyau… Elles ne sont pas adossées à un espace permanent de stockage. Le disque de pagination constitue ainsi le seul emplacement disque où elles peuvent être recopiées/modifiées au besoin. Elles sont éligibles à la pagination de sortie vers le disque de swap si elles ont été modifiées.

Pages Client : pages de client NFS, pages compressées : ces pages sont sauvées dans les disques standards locaux, CDROM pages, JFS amélioré, VxFS pages (tout type de file cache en dehors de JFS et GPFS qui utilise son propre mécanisme…).

En parallèle on distingue : les pages de "données" et les pages de calcul (traduction de computentional).

[27] Sous une dénomination différente, nous retrouvons ce même type de segmentation mémoire et d'implémentation sous Linux et Solaris. Je profite ainsi du cas d'AIX pour expliquer brièvement la problématique de mémoire et d'utilisation de la pagination.

Les fichiers de pages de données (cf. de non-calcul) regroupent les pages de données : elles sont appelées : Data file page, et se trouvent en tant que client page ou en persistent pages (donc présentes sur les disques). Elles sont en partie modifiables et peuvent donc être retournées vers leur emplacement d'origine en cas de modification, car elles proviennent d'un espace de stockage sur disque.

Les pages de calcul sont des pages de working segments ou des fichiers d'exécutions (code de programme). Une page qui n'a pas été modifiée n'a pas besoin d'être en page out (le page-out implique une recopie). Les pages disposées à être modifiées sont donc des pages de travail et des fichiers de données.

Ce qu'il faut retenir est que lorsqu'on évoque les phénomènes de pagination entre la RAM et l'espace disque de pagination, ceci ne concerne généralement qu'un type de page dans un contexte particulier : les pages dites de calcul sont celles qui ne peuvent être pas transférées *a priori* sur les systèmes de fichiers normaux.

Le principe du tuning de la mémoire sous AIX va donc consister à modifier la proportion de différents types de pages existantes en mémoire centrale, au profit de celles qui sont coûteuses en temps de transfert vers les espaces disques (pagination et autres), et donc pénalisantes pour les performances systèmes.

✓ Principaux paramètres pour l'optimisation de la mémoire sous AIX

Afin d'optimiser le fonctionnement de la mémoire centrale, on agit essentiellement sur quelques doublets de paramètres (des seuils ou des booléens). Il y a des dizaines de paramètres inhérents à l'optimisation de la mémoire ; dans la pratique, on résout principalement les problèmes mémoires uniquement *via* quelques doublets de paramètres.

Les mécanismes internes à la gestion de la RAM sont compliqués, je me limiterai donc à expliquer très brièvement les effets de ces paramètres.

- Les valeurs de seuil pour la gestion sont : *minfree*, *maxfree*, *maxperm*, *minperm*, *strict_maxperm*, *strict_maxclient* (les deux derniers paramètres sont par défaut adéquats).

- La valeur *%Numperm* correspond à la taille instantanée utilisée de la mémoire dédiée au cache FS. Ces valeurs sont visualisées en compte administrateur par la commande ***vmo***.

- Le *minperm* définit la taille limite inférieure en pourcentage de la RAM allouable au cache mémoire des fichiers. Le *maxperm* pour le type JFS en est sa borne maximale.

- *Minperm* / *maxperm* ou *maxclient* sont les seuils qui permettent d'agir sur les politiques d'éviction de page du cache I/O au profit des pages de calcul. Par défaut le VMM favorise les pages de calcul au détriment des pages de fichiers (c'est à dire le cache IO ou cache FS).

Il faut effectivement, dans des versions d'AIX inférieures à la version 6, souvent agir sur ces paramètres notamment en présence d'application utilisant leur propre segment de mémoire partagée telle qu'Oracle. Le comportement du système sera influencé par la manière dont les types de pages seront répartis.

Le *maxclient* est le pendant de *maxperm* avec un système de fichier de type JFS2 ou montage NFS. La disparition progressive du type de file system JFS rend *maxperm* obsolète, aussi on fixe en général: *maxperm* = *maxclient*.

Pour modifier les valeurs sous VMM s'effectue par la commande ***vmo***. Il existe également un moyen de surveiller l'allocation de la RAM à instantanément avec la commande ***svmon***, où nous pouvons diagnostiquer si la mémoire est sur-allouée.

Exemple de sortie de svmon:

```
svmon -G
               size       inuse        free         pin     virtual
memory      8388608     8333839       54769      674647     3805471
pg space    4194304       11167

               work        pers        clnt       other
pin          420647           0           0      254000
in use      3805471           0     4528368

PageSize   PoolSize       inuse        pgsp         pin     virtual
s     4 KB         -     7740335       11167      334119     3211967
m    64 KB         -       37094           0       21283       37094
```

Dans le cas ci-dessus, elle n'est pas sur-allouée (la *memory/size* étant supérieure à la somme des 3 autres valeurs en surgras (mémoire virtuelle , pages client en utilisation , pages persistantes en utilisation).

✓ Signes de contention mémoire sous AIX

S'il y a une mauvaise gestion de la RAM, le VMM passe son temps à référencer les pages et à opérer des entrées-sorties avec le disque de pagination.

Les signes indicatifs sont :

- Suractivité du scruteur de pages (*sr*) > 200 pages/sec fréquente
- Valeurs élevées du *po* (pagination de sortie avec le disque d' échange)
- Ratio sr/fr > 4 (pages scrutées / pages libérées)
- Suractivité du disque de swap [28] et occupation permanente élevée
- Certains processus ne peuvent plus montés en mémoire (alerte dans les fichiers journaux)

Il faut bien se rendre compte que les activités de pagination vers le swap disque doivent être corrélées avec le taux de scrutation (scan rate) et doivent être appréhendées d'un point de vue global. La pagination vers le swap disque n'est pas en soi une aberration puisqu'elle est inhérente au fonctionnement d'une machine.

Lorsqu'il reste trop peu de RAM libre, des interruptions et arrêts de processus peuvent survenir, d'où une application susceptible de s'arrêter de manière involontaire.

✓ Paramétrage VMM lié à l'utilisation d'Oracle

[28] Espace généralement dénommé hd6 pour AIX. C'est la désignation du disque d'échange.

Pour l'utilisation d'Oracle, dans les versions antérieures à AIX 5.3, il est nécessaire de réduire la taille du cache I/O si une pagination excessive est détectée ; notamment pour éviter une double "bufférisation" des données, car Oracle utilise son propre cache. Il est parallèlement nécessaire de s'assurer d'un mode d'accès aux données sur disques optimal. En effet les disques de données en accès (raw devices[29]), les I/O concurrents (CIO) et les filesystems avec l'option direct i/o contournent le cache mémoire du système.

Une autre préconisation [30] consiste à positionner le paramètre vmo *lru_file_repage* [31] à 0 (par défaut à 1), ce positionnement visant à favoriser les pages de calcul, au détriment du cache FS.

Les dernières versions d'AIX implémentent cette préconisation et ne nécessitent pas le repositionnement de ce booléen.

2.4.4 Gestion de la mémoire sous Solaris

De la même manière, Solaris propose un système de pagination basé lui aussi sur un disque de stockage. A l'instar de AIX, la mémoire centrale sera consommée par les différents caches (notamment le filepage cache appelé aussi cache des fichiers, le buffer cache), la mémoire partagée, le noyau et l'espace consommé par les processus. Aussi tous les processus en exécution sur la machine n'ont pas besoin de tous leurs segments mémoire montés en RAM simultanément.

Ne seront transférées en disque de pagination uniquement les pages pour lesquelles une modification ne peut être portée sur les emplacements de données originels et des processus en état swappés; cela est vrai pour la mémoire dite anonyme, qui sert pour des données qui sont appelées à être modifiées. Elles sont anonymes en ce sens qu'elles ne correspondent pas à un fichier nommé.

✓ Le tuning de la mémoire sous Solaris

La mémoire se paramètre essentiellement par des seuils de déclenchement du démon *pagger* (processus de scrutation de la mémoire) et par des modifications du fichier `/etc/system`. C'est le démon *swapper* qui migre certaines pages de la RAM vers le disque.

Les seuils [32]du démon de pagination à considérer (processus qui vise à maintenir une disponibilité de pages libres) restent en réalité quasiment inchangés[33].

Si la politique et les seuils ne sont pas adaptés, le système peut passer du temps à faire des transferts entre le swap et la RAM.

---

[29] Implémentation accédant à la donnée de manière brute (mode caractère) est non utilisée chez EDF.
[30] A partir des versions AIX 5.2 ML04 et 5.3 ML01.
[31] LRU (Last Recent Used) est un algorithme optimisant l'utilisation du cache FS et la mémoire centrale. Ce paramétrage indique notamment que VMM doit voler en priorité les filepages du cache I/O et laisser les pages de calcul en mémoire.
[32] Modifiable par le commande ***mdb*** (pour mémory débugger); il s'agit de *Lostfree, Slowscan, Desfree, Fastcan, Minfree, Throttlefree.*
[33] Je n'ai jamais eu à changer ces seuils au courant de ma présence chez EDF.

✓ Espace de pagination et de mémoire virtuelle

De la même manière que sur AIX, Solaris propose un espace disque dédié afin de compléter la mémoire centrale dans sa gestion de la mémoire virtuelle, c'est un pseudo file système avec une structure de données particulière. Ce pseudo-filesystem est aussi accessible par le point de montage `/tmp`, on peut y copier des fichiers (ce qui est en général déconseillé car cet espace est vidé lors des redémarrages de serveur; d'autre part, il est dédié à l'implémentation de la mémoire virtuelle). Un fichier déposé sous `/tmp` est logé en premier lieu en RAM. L'utilisation à bon escient de `/tmp` est donc primordiale.

L'occupation de ce pseudo [34] FS peut être vue par la commande ***df –k***, et sa taille disque configurée par la commande ***/usr/sbin/swap –l***.

✓ Interprétation de l'activité mémoire et signes de contentions

L'activité de la mémoire sous Solaris est observée essentiellement par la commande ***vmstat***.

```
vmstat 5 5
 procs     memory            page            disk          faults      cpu
 r b w   swap  free  re  mf pi po fr de sr m0 m1 m2 m1   in   sy   cs us sy id
 0 0 71 9534576 7039984 53 373 9 13 14 0 4  2  1  1  0  256 2275  931  1  1 98
 0 0 29 9541248 7072696 10 78 0 10 10 0  0  0  0  0  0  265 5690  655  1  1 98
 0 0 29 9541248 7072704 78 439 0 0 0  0  0  0  0  0  0  230 2381  655  0  1 99
 0 0 29 9541248 7072704 0 0  0  0  0  0  0  0  0  0  0  235  847  606  0  1 98
 0 0 29 9541336 7072792 0 1  0  0  0  0  0  0  0  0  0  224  531  598  0  0 100
```

La signification des métriques diffère peu de celle d'AIX. Par contre, la comptabilité des pages de la free list (pages disponibles) comporte une partie des pages cachées provenant des systèmes de fichier, elle est généralement de l'ordre de quelques Giga-octets[35].

Il est intéressant de remarquer que la colonne w (processus entiers déplacés en paging space) représente des processus qui, par manque de mémoire, ont été déplacés en swap comme c'est le cas ci-dessus, c'est donc *a posteriori* le symptôme d'un problème mémoire ; la colonne est remise à zéro si les processus sont exécutés de nouveau pour être terminés ou bien lors d'un redémarrage de la machine.

La métrique *sr* (scan-rate) est sans doute la métrique la plus à considérer sous Solaris : l'activation régulière du scruteur de pages indique que nous atteignons des niveaux de pages disponibles bas (cf free list[36])

La colonne *po* (page out) est également synonyme d'une activité d'écriture RAM vers le disque de swap.

La commande ***vmstat –p*** est quant à elle utile pour distinguer ce qu'on appelle le "bon" paging (celui des filesystems) du "mauvais" paging propre aux pages anonymes. Ce type de pagination anonyme indique que certaines pages doivent aller se loger en swap disque. En dernier recours les pages liées au noyau peuvent être déchargées de la mémoire centrale.

[34] Cet espace doit forcément être contigu sur le même disque.
[35] Sous AIX et LINUX, cette colonne laisse apparaître un espace de quelques centaines de pages.
[36] En l'occurrence à 1/64 eme de la RAM physique

Il est possible de connaître la répartition de la mémoire par des commandes comme ***Mdb*** surtout utilisée sous Solaris 8-9 et ***kstat*** plus usitée sous Solaris 10. Ces commandes sont rarement invoquées lors de nos audits.

✓ Paramétrages possibles sous Solaris

Il est possible de paramétrer les seuils du démon de pagination (ce qui en pratique n'est jamais opéré) et d'activer le *priority_page*.

Le démon *fsflush* est celui qui permet d'écrire les modifications effectuées sur les données dans les caches de la mémoire physique vers les disques.

Il est possible de paramétrer dans `/etc/system` son occurrence par la directive *tune_t_fsflushr* ; le paramètre *autoup* est la portion [37]de mémoire physique qui sera scrutée. Les machines EDF disposent d'une optimisation et configuration particulières au démarrage via le fichier `/etc/system`

En pratique

On agit davantage sur des aspects volumétriques de la RAM ou de la SWAP. Pour ma part les recommandations se sont limitées à ce type de remarques, ou à faire en sorte que les applications soient moins gourmandes en réservation de RAM et de bien dimensionner le système en terme de mémoire partagée et sémaphores[38] notamment les applications utilisant des segments partagés telles que les SGDB.

✓ Signes d'une contention mémoire sous Solaris dans un cas pratique

- Colonne Free list continuellement basse, forçant le scan-rate à se déclencher périodiquement.
- Activités *po/pi* élevées, couplées avec un *sr* élevé ;
- Pagination Anonyme (*Api/Apo*) élevées *via* le visu de la commande ***vmstat –p***
- Ratio pages scrutées / pages libérées (sr/fr) supérieur à un ratio de 4/1 de manière prolongée.
- Messages d'alerte de type "*no swap space available*" dans les fichiers journaux comme `/var/adm/messages` ;
- Nombreux process swappés (colonne *w* par **vmstat**) en mémoire centrale. (signes à posteriori d'une carence mémoire) ;
- Activité du process *fsflush* supérieure à 5% ;
- Défaut de pages majeures élevé (cf. ***vmstat –s***).
- Certains processus ne peuvent plus s'exécuter (vérifier `/var/adm/messages`)

[37] Attention : dans le cas d'importantes quantités de RAM, l'activité du *fsflush* peut être plus importante puisque la quantité de mémoire à scruter sera plus conséquente.
[38] Il s'agit de directives dans /etc/system sous Solaris 8 et le fichier /etc/project sous Solaris 10.

2.4.5 Gestion de la mémoire sous Linux

De nombreuses évolutions sont également à noter sous Linux depuis les noyaux 2.4.X jusqu'aux noyaux les plus récents du parc EDF (les Linux Redhat à base de noyaux 2.6.X[39]). A l'instar d'autres Unix, un certain nombre de paramètres sont modifiables avec des effets plus ou moins décelables. On retrouve des mécanismes de gestion de la mémoire assez voisin d'AIX ou de Solaris.

Lors de mon expérience chez EDF, le tuning de la mémoire sous Linux a été peu fréquent, c'est une situation en passe de changer du fait de la montée en puissance de Linux en guise de remplacement de Solaris et du nombre important de machine virtuelle exploitant Linux Redhat.

✓ Surveillance de la mémoire sous Redhat

Comme sous Solaris et AIX, ***vmstat*** et ***sar*** peuvent être utilisées. Les champs ***vmstat*** concernant la mémoire pour la RHEL sont un peu différents. Ci-dessous la sortie écran d'une ancienne version RHEL.

```
[cdq@dbacncq1 net]$ vmstat 5
procs                  memory      swap          io     system         cpu
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy wa id
 0  0 462036  24076 157144 106412    1    3   188   131   98   122  3  1  2 94
 0  0 462036  24076 157160 106404    0    0     0   132  166   202  3  0  0 97
 0  0 462036  24076 157176 106396    0    0     0    74  137   201  4  1  0 95
 0  0 462036  24076 157188 106400   21    0    22    27  159   205  3  0  2 95
```

swpd : Quantité de mémoire virtuelle utilisée (ko) partie RAM + partie disque.
free : Quantité de mémoire physique libre (ko).
buff : Quantité de mémoire physique utilisée comme tampons d'E/S (ko).
cache : Taille du page cache en utilisation dans la mémoire physique.
si : Quantité de mémoire paginée lue depuis un disque en ko/s.
so : Quantité de mémoire paginée transférée sur disque en ko/s.
bi : Blocs écrits par seconde sur des périphériques orientés bloc.
bo : Blocs lus par seconde sur des périphériques orientés bloc.

L'estimation de la mémoire disponible peut se faire entre autre par la commande ***free –m***.

✓ Configuration des paramètres mémoires

Les fichiers propres à la configuration mémoire se trouvent sous `/proc/sys/vm/`. Il est possible de paramétrer le comportement du noyau et aussi de la mémoire virtuelle *via* trois méthodes : par la commande ***systcl***, de modifier le ficher `/etc/systcl.conf` ou bien *via* des modifications sous `/proc/sys/vm/`, grâce à la commande ***echo***.

[39] Sous Linux les noyaux avec une terminaison impaire sont des noyaux dits de développement, de ce fait les noyaux 2.3 et 2.5 ne sont pas utilisés dans le cadre d'une exploitation.

Le développement sur les techniques du management de la mémoire pour Linux a été largement inspiré par les autres Unix, il n'est donc pas étonnant de retrouver des mécanismes similaires. Il y a une kyrielle de paramètres potentiellement modifiables à des fins d'optimisation mémoire.

✓ Pour les noyaux 2.4 (par exemple le noyau 2.4.21 pour la RHEL 3) - 2003 à 2010 (fin de support Redhat)

Les améliorations ont été portées sur l'algorithme de remplacement de page, sur la prédictibilité de l'algorithme, sa politique d'éviction des pages, et sur l'unification du page cache et du buffer cache.

Paramètres optimisables sous une 2.4.21 sous `/proc/sys/vm` (NB :les plus importants sont indiqués en sur-gras) :

bdflush	**kswapd**	overcommit_memory	pagetable_cache
dcache_priority	max_map_count	overcommit_ratio	skip_mapped_pages
hugetlb_pool	**max-readahead**	**pagecache**	stack_defer_threshold
inactive_clean_percent	min-readahead	page-cluster	

Il est important de noter qu'un certain nombre de paramètres ont assez peu d'effets significatifs.

Le démon *Bdflush* (*alter ego* du démon *fsflush* sous Solaris) détermine le taux de libération et de retour des pages contenues dans le buffer cache vers le disque, il est optimisable [40] et peut permettre d'éviter des contentions mémoires et disque ; il comporte nombre d'arguments. Ce démon permet la copie de pages dites sales (cf. "*dirty*") ou modifiées du page cache vers les disques.

Par exemple voici un paramétrage possible pour un serveur chargé au niveau I/O disques :

```
vm.bdflush="100 5000 640 2560 150 30000 5000 1884 2"
```

Le *kswapd* est quant à lui activé lorsque des seuils bas de free pages sont atteints ; son paramétrage indique le nombre de pages (*swap_cluster*) qu'il va tenter de libérer et de mettre en swap out. C'est une routine fonctionnant par intervalle de temps régulier et dépendant des seuils de déclenchement. Inversement, il permet de replacer des pages en mémoire depuis le disque de swap.

Le *pagecache* peut être ajusté par 3 valeurs (le maximum, le minimum et une valeur médiane) ces valeurs vont déterminer le comportement du démon *kswapd* vis-à-vis du cache.

✓ Noyau 2.6.9 (RHEL 4) – Noyau 2.6.18 (RHEL 5.4)

Les améliorations par rapport au noyau 2.4 sont nombreuses et permettent une plus grande stabilité de la machine *via* une meilleure gestion de la mémoire : optimisation liée à la gestion de la mémoire partagée (*reverse-file-page*), gestion des espaces d'adressage supérieurs, amélioration du cache TLB (Translation Lookaside Buffer)

[40] Il est conseillé de le modifier *via* la commande *sysctl.*

block_dump	laptop_mode	mmap_min_addr	percpu_pagelist_fraction
dirty_background_ratio	legacy_va_layout	nr_hugepages	swap_token_timeout
dirty_expire_centisecs	lowmem_reserve_ratio	nr_pdflush_threads	**swappiness**
dirty_ratio	max_map_count	overcommit_memory	vfs_cache_pressure
dirty_writeback_centisecs	max_writeback_pages	overcommit_ratio	zone_reclaim_mode
drop_caches	min_free_kbytes	page-cluster	
flush_mmap_pages	min_slab_ratio	**pagecache**	
hugetlb_shm_group	min_unmapped_ratio	panic_on_oom	

Notons l'absence du *bdflush*, remplacé par le *pdflush* (chargé des écritures mémoire du cachefile des pages modifiées vers les disques pour opérer aussi des synchronisations). Quatre paramètres permettent de modifier significativement le comportement du *pdflush* : *dirty_background_ratio, dirty_ratio, dirty_expire_centisecs,* et *dirty_writeback_centisecs.*

dirty_background_ratio indique le seuil pour lequel les processus *pdflush* commencent à synchroniser les pages modifiées du page cache avec les disques (par défaut à positionné à 10).

A noter que le *vm.pagecache* n'est plus réellement paramétrable sur des RHEL 4-5. Il est dynamiquement ajusté.

On va davantage utiliser la notion de *swapiness* qui est la propension du système à swapper. Ce nombre adimensionnel s'établit entre 0-100 (basse tendance à swapper – forte tendance). Ce paramètre est non déterministe et disons, quelque peu abscons.

Le démon *kswapd* va réduire le nombre de pages physiques utilisées par le système en cas de demande importante de page et va tenter de :

- réduire la taille du tampon et du page cache ;
- déplacer sur disque les pages de mémoire partagées ;
- déplacer sur disque et déréférencer des pages.

En supplément des données remontées par *vmstat*, il est possible de visualiser le fichier `/proc/meminfo` qui permet de disposer d'informations instantanées sur l'occupation de la mémoire.

✓ Les signes de contentions de la mémoire sous Linux :

A l'instar d'AIX, la mémoire libre disponible affichée est basse (cf. f*ree*) ; en effet on cherche à maximiser l'utilisation du cache I/O.

- La métrique *free* n'est pas en soi un élément très informatif, cependant sa variation peut être pertinente pour comprendre le comportement de la machine ;
- Les activités avec le swap disque (colonnes *si/so* du ***vmstat***) ou *via* un ***sar –W*** ;
- Sous Linux il n'y a pas de notion de métrique scan rate (NDR : colonne sr sous AIX/Solaris) ;
- Une augmentation de l'occupation du swap disque, au-delà de 60% d'occupation de manière continue ;

- L'activité de *bdflush/pdflush/kswapd* anormalement élevée (> 3-4 %) ;
- Nombre élevé de fautes majeures indiquant une mauvaise utilisation du page cache, ceci peut aussi être symptomatique d'un problème de dimensionnement swap et mémoire (cf. commande ***sar –B***).

✓ Quelques remarques supplémentaires

On notera les changements assez importants entre les différentes versions de noyau Linux non seulement sur les paramètres liés à la gestion de la mémoire virtuelle mais également sur les implémentations mises en place au fil des versions, les stratégies de gestion de la mémoire étant plus ou moins entièrement revues. Il faut donc être très attentif à la version de noyau sur laquelle on souhaite opérer des modifications. De plus on a la possibilité avec Linux de pouvoir facilement recompiler le noyau , ce qui peut être intéressant pour certains industriels , de réécrire ou d'adapter certains pilotes.

2.4.6 La mémoire partagée et la SGA

Lorsqu'on applique des paramétrages mémoire, il faut garder à l'esprit la taille allouée pour la mémoire partagée pour certains types d'applications (progiciels). Cette mémoire particulière est rattachée par plusieurs processus.

De ce fait lorsqu'on surveille la RSS [41] (mémoire physique réelle allouée au processus) des processus sur le serveur, le cumul des RSS de chaque processus concernant la même application ne peut excéder les capacités de mémoire physique de la machine, ce qui est bien sûr erroné. Ce biais est dû justement à l'attachement multiple de cet espace partagé par un nombre de processus. La mémoire partagée doit être le moins possible en espace de disque d'échange ; Oracle et autres SGBD sont une application qui fait appel à de la mémoire partagée en grande quantité.

Aussi il faut tenir compte de cette mémoire lorsqu'on établit les seuils du cache file I/O, notamment sous les anciennes versions d'AIX. Oracle a, dans son propre système de fonctionnement, un espace de mémoire partagé pour chaque instance de base, ainsi le cache file I/O (mécanisme propre à l'OS) n'a pas besoin de mettre en doublon les mêmes pages. Pour voir les espaces partagés, il faut que les instances soient démarrées.

On considère en général que l'ensemble des mémoires partagées ne doivent excéder 50-60 % de la mémoire physique totale. Il est possible de voir les espaces de mémoire partagées et les sémaphores par la commande ***ipcs*** [42]. Cette limitation par des directives sur le système d'exploitation de l'espace partagé est possible sous plusieurs systèmes :

Sous Solaris 8-9, il est possible de dimensionner la limite d'un segment d'un espace de mémoire partagée *via* le fichier `/etc/system` [43] et `/etc/project` sous Solaris 10 pour la limite de l'ensemble des espaces mémoires partagés. Pour Linux il s'agit de la valeur `/proc/sys/kernel/shmmax`. Sous AIX les segments mémoires partagés sont alloués à

[41] Resident Set Size, espace mémoire résident dans la mémoire physique
[42] Pour interprocess communication show ***(ipcs -mbo*** pour la partie mémoire partagée)
[43] Directive *shmsys:shminfo_shmmax*="valeur"

la demande par des primitives systèmes du type *shmmat*, en outre cela dépend du noyau [32-64 bits] et de la version d'AIX[44] .

2.4.7 Optimisations des caches

✓ Généralités

La problématique des caches se situe au confluent des optimisations mémoires et au niveau des optimisations I/O disques.

Leur but est de fournir à la mémoire centrale une facilité d'accès à l'information demandée et donc de diminuer les accès susceptibles d'être fait vers les disques physiques qui sont coûteux en temps d'accès et temps CPU.

Il y a un nombre important de caches dans un "système" et qui se situent à différents niveaux, nous avons par ailleurs abondamment évoqué le cache fichier de la RAM, généralement la taille des caches est antagoniste à sa vitesse d'accès (donc de leur performance intrinsèque).

Les caches les plus rapides se trouvent entre les processeurs et la mémoire centrale, ces tailles ne peuvent être définies et dépendent uniquement des implémentations matérielles. Ces optimisations de cache visent à améliorer le fonctionnement de la mémoire virtuelle/physique mais aussi des entrées/sorties opérées sur les disques. Un bon fonctionnement de la mémoire virtuelle/physique de la machine créé une externalité positive sur les performances des disques physiques.

Parmi les caches dénommés plus ou moins de manière différente en fonction des OS, on y trouve ceux que nous avons déjà évoqués :

- Le buffer cache qui contient les données inhérentes aux pilotes des périphériques en mode bloc. Les périphériques ne sont accédés qu'au travers du buffer cache.
- Le page cache qui comprend des données propres aux images et aux données sur disque ; lorsque des pages sont lues depuis les disques, elles sont stockées dans cet espace afin d'améliorer l'accès ultérieur à la même donnée. C'est le cache le plus volumineux du système (cache dont nous avons déjà parlé).
- Le swap cache qui contient les données modifiées qui ne peuvent être réécrites sur les disques de données y sont habituellement transférées.

Mais il y a encore d'autres type de cache améliorant les performances disques / mémoire par exemple :

- Cache des inodes.
- Cache DNLC [45];

[44] Pour plus d'information : http://publib.boulder.ibm.com/infocenter/AIX/v6r1/index.jsp?topic=/com.ibm.AIX.genprogc/doc/genprogc/ipc_limits.htm

[45] Directories Name Loop Cache (cache pour la résolution des nom de répertoire avec les inodes)

- Cache TLB [46];
- Cache des méta-datas (pour l'utilisation de gestionnaires de volume)

✓ Supervision des principaux caches

Quelques commandes usuelles pour estimer les caches les plus courants.

AIX :

Pour rappel, le cache des fichiers est borné par les valeurs *min_perm /max_perm* ou *min_client , max_client* et la taille est visualisable par le *num_perm* sous ***vmo –a*** ou via la commande ***svmon –G***.

Solaris :

La taille du page cache file ne peut être déterminée, seul le *segmap cache* (qui est une sorte d'anté-cache au sein du cache des fichiers) peut avoir une borne haute [47](en général 12 % de la RAM).

Il est possible également de déterminer le buffer cache (*bufhwm* sous `/etc/system`) : cette taille peut être exprimée en volume ou en portion de la RAM (2-5 %). L'efficacité d'un cache est exprimée en hitratio (on considère qu'elle doit être > 90% en lecture et > 65 % en écriture et mesurée par un ***sar –b*** par exemple). L'Estimation courante de la taille des caches se fait avec les commandes ***mdk –k :: memstat*** ou ***netstat –n system_pages***

Expérience personnelle: Je n'ai jamais eu à changer de taille de cache fichier (c'est par ailleurs un paramètre très peu surveillé).

Linux :

Sur les anciennes RHEL, la taille du cache fichier était paramétrable, dorénavant celui-ci s'ajuste de manière automatique. Nous n'avons pas de moyen de tuner les autres caches sur les dernières versions Linux Redhat. Pour voir la taille des caches : ***cat*** `/proc/meminfo`

2.4.8 Surveillance et contrôle des processus

Ce court paragraphe expose les moyen de surveillance des processus de la machine, un processus étant consommateur de deux éléments systèmes principalement de ressources CPU et de ressources mémoire qu'elles soient virtuelles ou physiques.

On ne peut pas réellement parler de tuning au niveau des processus puisque les processus émanent du code qui est interprété ; il faudrait donc être capable de redévelopper le code mais

[46] Translation Lookaside Buffer (cache utilisé par le gestionnaire mémoire pour la translation d'adresse virtuelle en adresse physique)

[47] Cf. /etc/system avec la valeur segmap_percent

d'un point de vue de l'ingénieur système : être capable de les suivre, d'en connaître les besoins en ressources physiques et les implications en terme d'I/O. Il est certes possible de modifier les priorités d'exécution des processus mais surtout d'apporter des contraintes systèmes à ces processus afin que ces derniers ne soient pas à l'origine de famine de ressources au niveau global de la machine.

✓ Suivi des processus par les commandes

Commune à toutes les plateformes UNIX, la commande ***ps*** (process show) permet d'avoir une visualisation instantanée des processus présents sur la machine. Dans une étude sur les utilisations des processeurs, ces informations sont particulièrement intéressantes car on peut déterminer les processus les plus consommateurs de ressources (avec les filtres appropriés). Ces filtres peuvent faire ressortir un classement par exemple en fonction du compte propriétaire du processus, de sa consommation mémoire ou plus communément de sa consommation CPU.

Connaître les processus consommateurs permet de cibler la recherche sur les causes de la déplétion des ressources du système et de rapidement demander une investigation du projet et de l'éditeur ou des équipes middleware si besoin.

Cette commande ***ps*** doit être itérée plusieurs fois pour être significative car cela représente une activité des processus instantanée, comme le souligne l'adage "un point ne fait pas une tendance". Des commandes temps réels peuvent nous renseigner également de l'activité, comme ***topas*** pour AIX, et ***prstat*** pour Solaris.

Des outils assez puissants et verbeux existent pour tracer l'activité au niveau du processus et des primitives systèmes appelées : il s'agit de ***dtrace*** (Solaris 10), ***strace*** sous Linux, ***truss*** pour Unix en général. On peut ajouter à cela une kyrielle d'outils propre aux éditeurs logiciels.

Les outils comme ***truss*** ou ***dtrace*** sont rarement utilisés car cela nécessite d'avoir une bonne connaissance du fonctionnement de l'application, des notions de programmations système et d'excellentes notions sur les appels systèmes.

✓ Moyens à disposition pour limiter la consommation des processus

Il y a des moyens de brider la consommation de processus : il faut noter que l'allocation mémoire pour un processus maximal possible est liée à l'architecture du processeur (32 ou 64 bits). Un processus 32 bits ne peut adresser que 4 Go de mémoire maximum. Ainsi sous Linux un processus ne peut pas prendre plus de 1 Go d'espace noyau et 3 Go d'espace utilisateur.

Les systèmes récents fonctionnent en général en mode 64 bits donc peuvent adressés en mémoire presque de manière infinie. Des limitations processus sous Solaris sont possibles par le fichier `/etc/system` ou `/etc/project` (sous Solaris 10) ; la commande ***prctl*** peut également limiter les ressources liées à un process. Notons également le process *rcapd* qui a pour but de contraindre la consommation de processus à certaines limites et la commande ***plimit*** *et* ***ulimit*** pour la "session" courante. Sous Linux, la limitation se fait par

exemple sous */etc/security/limits.conf* sous AIX, par le fichier */etc/security/limits*.

2.4.9 Préconisations mises en œuvre chez EDF pour l'optimisation mémoire

Dans un ordre non indicatif, pour le tuning de la mémoire voici les actions que j'ai appliquées :

- Discrimination des processus fortement consommateurs : demander leur optimisations aux équipes projets (Oracle, Java, Sybase, Informatica…) ;
- Limitation du cache I/O ; favoriser les pages de calculs ou anonyme si besoin ;
- Mettre en œuvre les mécanismes contournant la double mise en tampon des données.
- Contrôler les ressources allouées aux processus via les fichiers
- Augmentation de l'espace de swap disque ;
- Augmentation de la RAM ;
- Paramétrage des sémaphores et espace partagé au travers des fichiers de configuration ;
- Activer et utiliser les Huges/Larges pages pour les applications pouvant en bénéficier notamment pour Oracle.

Les résultats et gains de performance ont été parfois importants et ont pu améliorer très significativement la réponse des applications ou résoudre des problèmes de fonctionnement plus pénalisants pour le système.

2.5Performance et optimisation des disques et des systèmes de fichiers

Plus communément appelé tuning I/O[48], c'est sans aucun doute la partie la plus délicate à traiter étant donné la multiplicité des mécanismes, des couches logicielles et physiques parties prenantes et des effets possible négatives pour des serveurs utilisant une baie de disque. C'est par ailleurs une partie souvent prise en charge par les équipes du stockage tant les impacts peuvent être important pour un ensemble de serveurs.

Les liens sont forts entre le tuning I/O et celui de la mémoire. De même, paramétrer ou concevoir un système optimisé au niveau des I/O nécessite de comprendre la mécanique de gestion de la mémoire.

Cette partie s'articulera sur :

- Une présentation générale de la problématique,
- Une explication du rôle de chaque couche de la pile[49] I/O (les variantes entre systèmes d'exploitation seront évoquées sans rentrer dans les détails),
- Une présentation du jargon lié à la performance des "disques",
- Les outils de supervision des activités d'entrées/sorties avec les disques
- Un diagnostic générique des problèmes de performances I/O
- L'énumération de ce qui peut être fait pour rendre un système plus optimisé dans la gestion des I/O.

2.5.1 Considérations générales de la problématique de la pile I/O

A l'instar des problèmes mémoire et CPU, les problèmes de I/O sont généralement supposés lorsque les applications "ralentissent"; les chargés d'application observent très rarement les outils de supervision et il n'y a en général aucune alerte d'outil tierce à proprement parler. Les mécanismes liés aux entrées/sorties disques sont complexes.

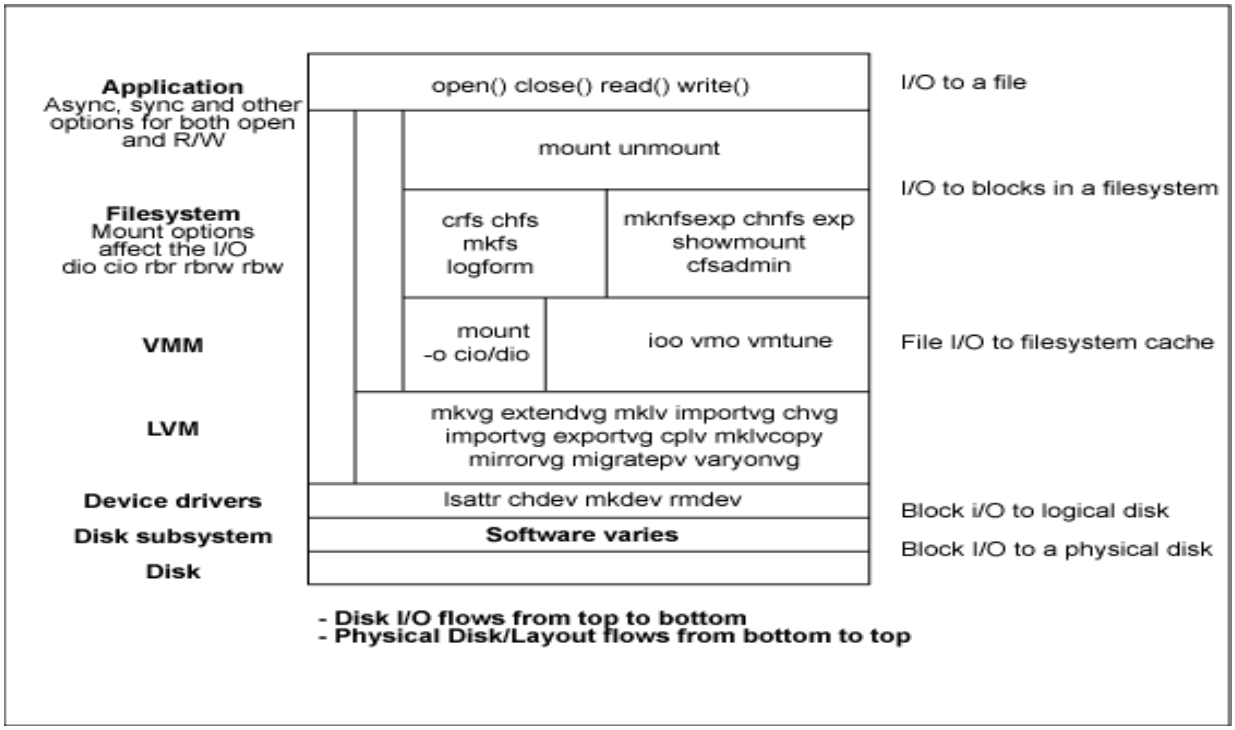
Chaque sous-système I/O utilise ses propres tampons et mécanisme de cache et certaines applications elles-mêmes maintiennent leur propre cache à l'instar d'Oracle ou de Java. Au-delà de ces facilités mémoire, chaque couche essaie d'implémenter un ordonnancement des I/O demandées afin de minimiser les sollicitations faites aux disques physiques, par le biais de contrôle d'échéance, ou par exemple par agrégation d'I/O.

Une mauvaise intégration de la chaîne I/O peut non seulement amener à des latences énormes, mais aussi à une déplétion des ressources mémoires et de la suractivité processeur.

[48] I/O signifiant Input/Output ou E/S (entrée/sortie) en français : les deux acronymes seront utilisés indifféremment.
[49] Egalement appelée Stack I/O

De manière très synthétique, voici schématisée la pile I/O sous un système AIX[50].

NB : La partie droite est commune à tous les systèmes.



Commentaire

L'application utilise des primitives de lecture et d'écriture qui sont effectuées sur la couche système de fichiers ; ces demandes d'I/O sont ensuite prises en charge par la partie mémoire et différents caches associés (caches inodes, buffer cache, cache I/O file) ; si les demandes ne sont pas satisfaites par les caches mémoires, c'est la couche gestionnaire de volumes (LVM pour AIX) qui accède aux buffers des drivers de disques logiques et enfin jusqu'aux disques physiques de données. En synthèse, nous avons la chaîne en approche "*UP to BOTTOM*": (attention cela ne représente pas forcément le chainage physique pour le traitement des E/S) :

APPLICATION - FILESYSTEM – MÉMOIRE – VOLUME MANAGER – DRIVERS DISQUES – CONTROLEURS - DISQUE PHYSIQUE

Enumérer la problématique de performances d'I/O n'est pas uniquement une question de performances des disques physiques : il s'agit de considérer tout l'ensemble de la chaîne I/O. Nous avons vu que la CPU est l'élément le plus rapide d'un serveur, que la mémoire centrale est plusieurs milliers de fois moins rapide que cette dernière, de même les accès disques sont très supérieurs en latence à l'accès mémoire (qui est de l'ordre de la dizaine de

[50] Schéma tiré du Redbook IBM traitant de la performance sur les AIX 5L

millisecondes). De ce fait les accès disques peuvent se retrouver comme goulot d'étranglement.

La bonne performance de la chaîne d'I/O tient avant tout énormément à un travail de conception architecturale: il faut effectivement choisir des disques performants, voir la répartition des données sur la partie disques physique, décider de la stratégie de répartition des informations entre disques (combien de disques physiques? Quelle stratégie RAID [51] ? Quel type de Baie? Quel type de file system ? Quelles options d'accès à la donnée : direct I/O, concurrent I/O? Comment seront accédées les données : séquentiel, aléatoire?) et par conséquent trouver le paramétrage approprié au niveau du gestionnaire de volume. Nous allons brièvement passer en revue les éléments de la pile I/O.

2.5.2 Performances des disques physiques

Un disque est l'espace de stockage de données par excellence, les données sont dites permanentes. Un disque, physiquement, est un empilement de "galettes" accédées en écriture ou lecture par un des bras de lecture/écriture. Si la capacité de stockage suit une progression exponentielle, il n'en n'est pas de même pour la vitesse de rotation des disques et de la mécanique, liée à la gestion des disques physiques. Les disques les plus performants ont une vitesse rotative de 10 000 rpm [52]. L'accès à la donnée sur le disque reste fonction de la vitesse de rotation du disque et du positionnement sur la bonne piste du bras qu'on appelle *seek_time*[53]

✓ Pour avoir accès aux informations de configuration des disques :

#***prtconf*** pour Solaris

#***hdparm*** pour Linux (disque IDE) – ou ***dmidecode***

#***lsattr*** ; ***lscfg*** pour AIX

Lorsqu'on veut appréhender l'analyse de disques physiques, il faut évoquer :

- L'environnement (disque physiques interne ou baie de stockage) ;
- L'utilisation du disque ;
- La saturation ;
- Le débit ;
- Le seek time (ou temps de positionnement);
- Le temps de service ;
- L'accès à la donnée.

[51] RAID : *Redundant Area Inexpensive Disk* : désigne les techniques permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit la tolérance aux pannes, soit la sécurité, soit les performances de l'ensemble - ou une répartition de tout cela.

[52] rpm pour "*round per minute"* (tours par minute).

[53] Anglicisme pour "temps de rencontre" : ici, il s'agit du temps nécessaire au bras de lecture pour se positionner au début de la lecture de la donnée.

✓ Accès à la donnée

Il est important de différencier si les demandes d'I/O sont séquentielles ou aléatoires, autrement dit si on a une lecture des blocs consécutifs d'un point de vue spatial. Les outils pour déterminer ce comportement peuvent le faire en traçant le processus par la commande ***truss***.

Exemple de capture ***truss*** pour un accès aléatoire

```
22975/3:        lseek(19, 0x6A0E2000, SEEK_SET)                 = 0x6A0E2000
22975/3:        read(19, " >DCE8 i\0\0\0\0\0\0\0".., 65536)     = 65536
22975/3:        lseek(19, 0x6AA52000, SEEK_SET)                 = 0x6AA52000
22975/3:        read(19, " >DCE8C0\0\0\0\0\0\0\0".., 65536)     = 65536
```

Bref commentaire : Après la primitive *lseek*, on observe une primitive *read.*

Dans un accès séquentiel, une seule primitive *lseek* est nécessaire à plusieurs primitives *read.*

Intuitivement on en déduit que le débit sur un accès séquentiel est plus important que sur un accès aléatoire car on ne perd pas le temps de positionnement pour accéder à la donnée. De manière expérimentale on peut connaître le type d'I/O que font certains progiciels, Oracle par exemple lit ces données le plus souvent de manière séquentielle pour des opérations d'export de base…

De même, il est important de savoir si les disques sont locaux, distants (par exemple montages NFS[54]) ou bien situés sur une baie de stockage. Certains sont accédés par la carte réseau, la plupart par des cartes contrôleur dédiées (HBA [55]– fibre/SCSI)

✓ Utilisation

Un disque très occupé sera souvent l'objet d'une contention ; il faut aussi être vigilant à ne pas extrapoler une suractivité temporaire à une contention réelle du disque. Par exemple, en ce qui concerne les disques de Baie, bien qu'apparaissant à 100 % utilisés *via* ***iostat*** (%b ou %tm_act) ne témoignent toujours d'une saturation, car il s'agit un accès logique vu depuis le système.

Cette métrique est pertinente surtout pour les disques internes. Un ratio d'utilisation moyen d'un disque de plus de 40% est un signe avant-coureur de saturation.

✓ La saturation

[54] NFS : *Network File System*, protocole d'échange de données entre deux serveurs distants, inventé par SUN Microsystem.
[55] Host Bus Adapter

Celle-ci est synonyme d'une forte utilisation du périphérique dans laquelle des transactions supplémentaires ne peuvent être acceptées et viennent donc engorger les files d'attentes ; c'est un corollaire à une utilisation intensive.

✓ Les débits

Les débits sont de très bons indicateurs des performances des disques, encore faut-il que le mode d'accès soit connu; en général les accès séquentiels sont une dizaine de fois supérieur à un accès aléatoire.

Pour avoir une idée du débit de fonctionnement, nous pouvons utiliser la commande ***dd*** [56] pour copier des blocs séquentiels.

Les débits sont bien sûr fonction des caractéristiques mêmes des disques et des moyens d'accès, et des différents paramétrages des couches. Cela peut être utile, lors des audits, de tester cette capacité de transfert.

On considère que le débit de sortie d'un adaptateur SCSI ne doit pas être au-delà de 70% de sa capacité maximale.

✓ Temps de service

Le temps de service pour une requête est la donnée à prendre en compte (*service_time* ou *asvc_t*). Ils ne doivent pas excéder plus 20-25 ms pour des disques physiques ; sur les baies EDF, ils doivent être inférieurs à 12 ms.

Ainsi les paramètres concourants à une baisse de performance des disques peuvent être dus :

- aux recopies (miroirs de disques, mécanisme de redondance matérielle)
- à la fragmentation du disque surtout pour la lecture séquentielle
- à la mécanique des disques ;
- à la saturation des adaptateurs et des bus ;
- à une mauvaise localisation des données sur les plateaux de disques ;
- à l'épuisement de la mémoire tampon.

2.5.3 Sous-système de disques

Il est important de connaître la configuration des disques en termes de redondance et recouvrabilité : on parle de configuration RAID. Ces mécanismes sont impactant sur les performances des disques.

[56] Commande de recopie bit à bit sous Unix par exemple la commande dd if=/dev/zero of=/monrep/monfichier bs=1024k count=2048

Au sein d'EDF, en ce qui concerne les disques de baie, nous avons du RAID 5 (7+1 : c'est-à-dire une répartition sur 7 disques avec 1 disque de parité.) Les disques de baie sont de capacité de 29 Go à 160 Go pour les plus récents. Chez EDF, un disque de baie pour le système est en fait une enveloppe et un accès logique à plusieurs disques physiques. Ces grappes de disques seront accédées par le biais d'adaptateurs performants.

Ces chemins d'accès aux disques se feront par le biais de cartes et de commutateurs par les SAN. Ces accès sont généralement redondés avec des débits de l'ordre du Gigaoctet. Les disques internes sont en miroir[57], ce qui nécessite de prendre en compte les synchronisations prises charge automatiquement par le gestionnaire de volumes. Pour que le système puisse gérer au mieux l'interface avec ses disques, la machine doit être également à jour au niveau des microcodes ou version de drivers ; par ailleurs, il existe des pré-requis essentiels au niveau des mises à jour du système d'exploitation afin de pouvoir être conforme au matériel de la baie de disques (NB : préconisations constructeurs).

Les modèles de matériels pour se connecter aux disques de baie et la baie de disques sont définis par une équipe de spécialiste au sein d'EDF, l'équipe OGD[58].

2.5.4 Gestionnaire de volume et file system

Le rôle du gestionnaire de volume est de pouvoir gérer des volumes logiques sur un ou plusieurs disques physiques ou logiques d'un point de vue système ; il offre un certain nombre d'option de gestion améliorant la fiabilité et la performance des disques. Un volume manager permet de définir des groupes de disques [59]et des volumes logiques, ces volumes logiques permettent de s'affranchir des limitations de volume de données liées aux disques physiques.

Par ce biais, il est possible de ventiler un système de fichier sur plusieurs disques physiques. Les disques mis à disposition par le stockage de données ont généralement une capacité logique de quelques dizaines de giga-octets à plus d'une centaine de Go, cette volumétrie est réellement repartie sur plusieurs disques physiques (même pour des volumes logique de quelques Go). Il est possible de voir un certain nombre de paramétrage pour les gestionnaires de volume *via* les commandes d'administration.

La gestion de volume laisse davantage à l'administrateur système le loisir de pouvoir ajouter des optimisations logicielles. En fonction du gestionnaire de volume, il est même possible de définir la distribution des données qui y seront inscrites et la répartition de charge également aux niveaux des disques logiques.

2.5.5 Optimisation du système de fichier

Lorsque les systèmes de fichiers (appelés dans le jargon : FS comme File System) sont créés ceux-ci le sont en général au sein d'un volume logique maintenu par le gestionnaire de volume. A la création des FS, il est important de définir les paramètres appropriés à leur usage futur, notamment la taille de block [60] ou le nombre d'inodes que pourra contenir un système de fichier. Après sa création, il est possible d'apporter un certain nombre d'optimisations. On

[57] Miroirs à deux membres.
[58] Opérateur à la Gestion de Données. Il s'agit d'une cellule d'expert gérant le stockage de données.
[59] Il s'agit d'un agrégat de disques physiques dédiés.
[60] Unité élémentaire de stockage de données sur disque.

dénombre une multitude de type de FS sous les différents systèmes ; voici ceux utilisés chez EDF :

Systèmes d'exploitation	Type de système de fichiers (pour EDF)
Solaris	UFS en natif, VxFS
AIX	JFS/JFS2
Linux	Ext2 / Ext3

Les optimisations des filesystems *via* le volume manager doivent se baser sur une approche pas-à-pas avec des scenarii différents (on peut par ailleurs trouver en annexe un exemple de scénario de test effectué dans ce cadre par l'équipe OGD).

Les FS comportent également un certain nombre d'optimisations, celles-ci sont en relation avec des paramètres mémoires et du paramétrage du gestionnaire de volume.

Des optimisations peuvent être tirées par les options de montage (*via* la commande ***mount***). Ces options de montage peuvent désigner un mode d'accès. Voici ces modes :

- Le Direct I/O (DIO) permet de ne pas cacher la donnée dans le cache I/O de la mémoire centrale.
- Le Concurrent I/O (CIO) permet à de multiples processus d'écrire et de lire de manière concurrente un fichier en évitant le verrou sur l'inode et bypass le cache I/O.
- Les Asynchrones I/O évitent les attentes d'I/O sur la complétude d'une demande I/O ; un certain nombre d'application peuvent bénéficier de ce mécanisme comme Oracle.

Les accès en raw devices ("accès direct brut" avec les disques) ne sont pas mis en œuvre chez EDF. Les trois autres le sont et font d'ailleurs l'objet de préconisations de notre part.

✓ Cas de Solaris

Chez EDF deux types de système de fichiers sont utilisés : UFS et VxFS. Nous allons décrire les principaux mécanismes pour l'optimisation.

UFS [61] est le système de fichier par défaut de SUN, il va en général de pair avec le gestionnaire de volume natif SDS[62] pour les versions 8 de Solaris ou SVM[63] pour la version 10. Ainsi il est possible d'anticiper une lecture séquentielle dans ce système de fichier afin de réduire la latence induite par les accès disques[64], en supplément l'accès en DIO peut être opéré.

[61] Pour Unix File System.
[62] Pour Solaris DiskSuite.
[63] Solaris Volume Manager qui est la version améliorée de SDS implémentée a Solaris 10.
[64] Ce paramètre sous UFS est renseigné par le *maxcontig* qui se paramètre *via* la commande *tunefs*.

VxFS est un type de file system qui a été développé par Veritas (SYMANTEC), il est particulièrement performant pour des bases de données et de forts volumes de stockage/transfert ; il est aussi beaucoup plus abouti qu'UFS pour l'accès séquentiel, pour être exploité il se doit d'être supporté par le volume manager de Veritas (VxVM) ; VxVM peut également gérer l'UFS. Il a aussi l'avantage d'être aussi supporté sur AIX et LINUX, sa licence reste onéreuse.

ZFS [65]n'était pas implémentée à EDF en 2011, il offre cependant une grande flexibilité et robustesse et des performances accrues vis-à-vis d'UFS.

Les valeurs optimisables sous VxFS sont nombreuses et la plupart sont définies par les préconisations que le constructeur des baies fournit à OGD.

Quelques-unes [66]me semblent les plus significatives en terme de performances et préconisées soit par notre cellule ou la cellule du stockage de données (cf. annexe pour exemple)

Pour voir les paramétrages d'un FS de type VxFS, il suffit d'exécuter en tant que super utilisateur (appelé compte root) : ***vxtunefs –p*** *</point de montage>*.

Le mécanisme de Direct I/O est implicite sur VxFS.

Si des valeurs venaient à être reconsidérer sous VxFS, il est impératif d'avoir le consentement de l'équipe stockage, tout simplement parce que les baies sont mutualisées entre plusieurs serveurs et maintenues par cette équipe - et aussi parce que l'impact peut être important sur les performances de la baie, donc avoir des externalités négatives significatives sur les autres serveurs connectés à cette même baie.

En pratique chez EDF, les optimisations I/O du côté du logiciel portent sur les options de montage et sur la couche VxFS, elles sont monnaie courante et se font de manière concomitante à l'installation des progiciels et/ou au raccordement des disques à une baie. Les améliorations peuvent être très importantes, j'ai pu observer des gains de 400 % sur des temps d'export de bases de données liées à la modification d'option de montage et de paramètres du gestionnaire de volume.

✓ <u>Cas d'AIX</u>

JFS [67]et JFS2 sont les deux types de FS exploités chez EDF pour AIX ; JFS est en voie de disparition. JFS 2 présente un certain nombre d'amélioration par rapport JFS. Il faut être vigilant, à la création de FS, des valeurs par défaut appliquées aux FS : il est possible de

[65] ZFS comme Zeta FS est le nouveau type de système de fichier implémenté depuis Solaris 10, il comble les défauts principaux de SVM et est d'une gestion plus souple et s'avère performant.

[66] elles se dénomment *Discovered_direct_iosz* : taille à partir de laquelle une lecture est considérée comme séquentielle et qui permet de contourner le cache mémoire ;*Read_pref_io*, *read_unit_io / write_pref_io* et *write_unit_io* : tailles I/O préferées;*Qio_cache_enable* : mécanisme d'écritures multiples (intéressant pour Oracle 32 Bits) ;*Max_direct_iosz* : taille maximale d'émission de l'I/O depuis le FS ;

[67] JFS pour *Journalized FileSystems*

modifier ces valeurs par la commande d'administration ***ioo***[68]. JFS étant de moins en moins rencontré chez EDF et en extinction je ne le commenterai pas.

Sous JFS 2 on peut noter principalement certains mécanismes :

- l'I/O pacing ("fluidification" des demandes I/O) .
- La lecture anticipée[69];
- Les Concurrents I/O peuvent être activés sur certain FS notamment avec Oracle (particulièrement sur les FS contenant les tables) ;
- Le Direct I/O est possible mais le CIO supplante les fonctionnalités offertes par celui-ci et offre les mêmes avantages étendus.

En pratique, nos efforts d'optimisations portent sur le montage des FS. Pour ma part je n'ai jamais eu à modifier les paramètres du gestionnaire de volume *via* la commande ***ioo*** sous AIX (les paramétrages par défaut hérités des recommandations du constructeur sont adéquats) et sont très rarement modifiés. OGD suit les préconisations IBM sur les valeurs modifiables par ***ioo***. Sur les audits effectués sur AIX je n'ai jamais eu à les modifier.

✓ Linux

Sous Linux, 3 types de FS coexistent par ordre de diffusion : ext3, ext2 et JFS.

Ext 3 est une amélioration de ext2, il est possible de convertir assez facilement du ext2 vers de l'ext3. JFS est un transfuge de la technologie IBM. Ext3 apporte des améliorations en termes de capacité et de vérification/réparation de filesystems dues à l'implémentation directe de la journalisation sous ext3.

De la même manière il est important, à la création du FS, d'apposer les bons paramétrages. Il est possible de paramétrer les FS par la commande ***tune2fs*** - notamment le *blocksize*. Expliqué dans la partie mémoire, le tuning du démon *bdflush* et *pdflush* peuvent jouer un rôle important dans les performances I/O, par la mise à jour des disques par rapport aux pages modifiées en mémoire centrale qui implémentent aussi un cache I/O en mémoire.

Des mécanismes de lecture anticipée sont aussi possibles[70] et d'autres paramètres mémoire sont efficaces pour la gestion de caches FS.

Mon expérience personnelle sur le tuning I/O pour Linux sur des cas pratiques s'est essentiellement orientée sur la partie implémentation: type de FS, montage, mirroring, multipathing. En effet avant d'émettre des recommandations mettant en œuvre des mécanismes assez complexes, il est préférable d'inspecter la conformité de la machine sur sa configuration de disque et pile I/O avant de chercher à utiliser ***tune2fs*** pour améliorer les performances.

Il faut toujours s'épargner la modification de paramètres mal compris ou mal appréhendés.

---

[68] **io*o* –*L*** pour lister les paramètres

[69] Par les variables *j2_xxxPageReadAhead* xxx qui correspondent soit à minimum soit à maximum

[70] *Via* les paramètres */proc/sys/vm/min-readehead* et *max-readhead*

2.5.6 Outils pour suivre les performances I/O disques et interprétation

✓ La commande iostat

Sur tous les systèmes UNIX, ***iostat*** est la commande la plus triviale de celles utilisée pour avoir une idée des activités disques ; elle n'est cependant pas la plus aboutie pour avoir une interprétation correcte de l'activité disque. Il y a quelques légères différences entre les versions d'***iostat*** sous les différents Unix.

✓ Sous Solaris

Exemple de capture d'écran de la commande :

```
# iostat -cnx 5

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    0.0   0   0 c0t0d0
0.3    1.2    1.2    2.4  0.0  0.0    0.2    8.7   0   1 c1t0d0
0.2    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t1d0
0.2    0.9    0.5    1.9  0.0  0.0    0.0    6.8   0   1 c1t2d0
7.7    7.2  246.3  175.6  0.0  0.4    0.0   26.6   0   6 c1t3d0
```

Les 4 premières métriques sont liées au taux de transfert (lecture/écriture) sur les transactions (nombre d'opérations / volumétrie en kilos)

wait = nombre moyen de transactions en attente ;

actv = nombre moyen de transactions actuellement en cours de service ;

wsvc_t et *ascvt* = temps d'attente et de service moyen en ms pour une transaction ;

%w/%b = ratio d'attente/d'utilisation du périphérique.

L'ensemble de ces métriques est surtout pertinent pour les disques internes.

✓ Cas de la commande vmstat

Une métrique était très utilisée pour apprécier les attentes liées aux disques : le *wait* i/o[71]. Cette métrique indique le pourcentage de temps où le processeur est en état libre et pour lequel une i/o est en cours, donc en attente de terminaison. Il y a une certaine forme de confusion à interpréter cette métrique, puisque le processeur peut effectivement traiter en simultané des threads éligibles surtout sous les systèmes multiprocesseurs parallélisés (SMP) : cette pertinence est dorénavant presque désuète, elle n'est plus considérée sous Solaris et est à évaluer avec prudence sous AIX et Linux² .

Pour avoir une vision plus précise des attentes i/o sur un processus, il est possible d'utiliser des commandes plus prolixe comme ***truss***, ***strace***, ou même ***dtrace*** [72].

[71] Il s'agit de la colonne *iowait* (AIX) %wio (Solaris – Linux).
[72] *Dtrace* est uniquement utilisable pour Solaris 10.

✓ La commande Sar

Cette commande omniprésente sous Unix/Linux n'est pas pour autant installée ou activée par défaut sur toutes les serveurs, effectivement le nombre de métriques remontées est assez conséquent. J'utilise fréquemment cette commande pour Linux qui est très synthétique pour une première approche.

✓ La commande topas

Sous AIX, cette commande est assez utilisée pour également recouper les activités disques avec les activités de mémoire, CPU, et informations sommaires sur le système.

2.5.7 Autres commandes moins usitées sous AIX

Lvmstat : utilisée pour la supervision au niveau Logical Volume ;

Filemon : très peu utilisée chez EDF ; permet de tracer les activités I/O d'un périphérique logique ;

Fileplace : permet de voir la contiguïté d'un fichier.

2.5.8 Cas particulier des disques logés en baie de stockage

En effet les disques ou LUN vus depuis le système *via* la commande ***iostat*** sont en fait plus un chemin vers un disque logique de baie; ce disque logique est accédé potentiellement par deux ou quatre cartes fibres (en accès round robin load-balancing / failover : accès tout à tour, équilibrage de charge et tolérance de panne) et correspond à une multitude de disques réellement physiques. Les commandes systèmes que nous disposons pour évaluer la saturation des disques en SAN sont donc relativement inappropriés.

De plus les mécanismes de cache de la baie rendent l'interprétation des métriques sur les temps de services peu représentatifs de la réalité, notamment lors des écritures par des acquittements en avance de phase ; en effet la « baie » peut recopier en différé la donnée depuis le cache de la baie vers les disques physiques. Ainsi les temps de services seront aperçus très bas notamment pour la partie écriture #4ms car le cache de la baie effectue un acquittement "instantané" ; quant aux lectures, la donnée doit être présente dans le cache de la baie pour être acquittée ; en conséquence, les temps de service les plus importants concernent les lectures aléatoires #8ms. Dans ce contexte la commande ***iostat*** mesurera davantage la latence entre la sortie des cartes fibres du serveur et le cache de la baie de stockage.

Dans le cas d'EDF nous disposons de l'interface intranet de supervision d'OGD pour estimer des performances de la baie de disque. Ces disques peuvent aussi concerner les disques systèmes, technologie du Boot on SAN.

Cette interface permet de faire des analyses de performance liminaires :

- Sur les temps de services en lectures / écritures (de 2 à 10 ms) ;
- Occupation des liens SAN (taux de transfert et équilibrage de charge).
- Conformité de la configuration (fonctions de test implémentées sur l'IHM).
- Volumétrie transitée.
- Efficacité du cache de la baie (cf. Hit ratio).
- Information sur la baie, numéros de ports et configuration basique.

Les métriques mises à notre disposition sont très généralistes ; l'équipe de stockage, en fonction des observations retournées par ASC, mènera des investigations supplémentaires si une anomalie venait à être décelée par notre cellule.

2.5.9 Optimisation de la couche NFS

NFS est un protocole RPC à la base développé par SUN mais largement répandu, notamment sur les serveurs Jumpstart et NIM[73] pour un partage de ressource disque ; aussi il est utilisé dans le cas d'un montage sur un serveur NAS. Les transferts I/O transitent sur le réseau TCP/IP, donc dépendant de la fiabilité du réseau et des machines client et surtout serveur. Un certain nombre d'optimisations peuvent être menées, dont les tailles de fenêtres de réception et d'émission de la couche TCP sur le cache NFS. Nous ne mettons pas œuvre pour le moment d'optimisation sur NFS et est utilisé à la marge.

2.5.10 Préconisations générales sur les disques et couche I/O chez EDF

- Bien concevoir l'intégration des différentes couches I/O, notamment en termes de performance en fonction des applications ;
- Disposer d'un matériel récent, mettre à jour les microcodes (Firmware) ;
- Tuner la mémoire avant de tuner les couches I/O. (cf synoptique)
- Veiller à utiliser et à paramétrer les bons type de systèmes de fichiers notamment en termes de performance pour des lectures séquentielles.
- Informer / travailler avec les équipes de stockage.
- Surveiller la saturation de l'espace des disques plus propices à la fragmentation.

[73] Jumpstart est un serveur SUN dont la tâche est de pouvoir déployer et installer des souches ou objets, il permet également l'amorçage de machines sans OS, NIM (*Network Installation Management)* est l'équivalent pour IBM/AIX.

- Ne pas hésiter à utiliser des commandes en temps réel ; certaines saturations I/O se présentent de manière fugitive mais sont pénalisantes en fonctionnement interactif ;
- Etre attentif aux temps de services rendus par les disques (interne, et aussi sur baie et la saturation potentielle des cartes fibre notamment).
- Corréler le temps de service des disques avec les volumes transférés.
- Passage en revue de la configuration des disques et de la conformité.
- Utiliser les disques de baie au lieu de disques locaux si possible.
- Migrer vers un système de fichiers plus performant.
- Repartir la charge sur plusieurs disques, notamment les objets des bases de données.
- Revoir les appels d'I/O pour les applications , notamment des bases des données par exemple : optimisation de l'application , indexation des tables, buffer applicatifs , optimisation des requêtes …
- …

2.6Analyse des performances réseaux

2.6.1 Généralités

Dernier élément étudié dans l'étude de la performance des serveurs informatiques : le réseau. Le réseau étant défini par l'ensemble des moyens d'interconnections (routeurs, équipement fédérateur, commutateurs) et des cartes de transmission des serveurs.

Il faut en premier lieu définir pour un administrateur système ce que cela sous-entend : il est évident que du point de vue du réseau, nous n'avons pas véritablement une visibilité sur tous les éléments physiques du réseau ; le serveur n'est en général qu'une "terminaison" du réseau. A ce titre il a en général une ou plusieurs cartes connectées à un ou des "réseaux". Il est possible d'exécuter quelques commandes qui nous permettrons d'observer les débits d'entrée et de sortie à partir des cartes et les temps de réponse à une requête.

Ce qui se passe en terme de débit et de fiabilité sur un commutateur extérieur au serveur est donc parfaitement inconnu et n'est pas véritablement de notre ressort; enfin le commutateur est un des éléments des plus propice à des externalités négatives sur les autres équipements et machines desservies. Le diagnostic d'une contre-performance du réseau est donc assez délicat puisqu'il suppose de travailler avec des équipes réseaux et d'être très prudent en terme de diagnostic - puisqu'il y a une multitude d'éléments pouvant concourir à une dégradation des performances réseau.

✓ Symptômes des problèmes de performance réseaux

Parmi les symptômes les plus courants :

- Une application qui répond lentement à une requête (par exemple une identification sur une IHM web, ou bien un résultat de requête SQL …) ;
- Un serveur de temps à autre non disponible par la télésurveillance.
- Des temps de sauvegardes trop longs.
- Des réplications de base et des transferts de fichier trop longs.
- Déconnexions inopinées.

Les raisons génériques

Elles peuvent être très diverses.

- Problème d'interfaces réseau : une vitesse insuffisante et des taux d'erreur élevés dus à un matériel défectueux ou mal configuré. Les modes de négociations des périphériques matériels y compris les incompatibilités concernant les modes uni/bi

directionnels [74] peuvent provoquer des erreurs réseaux et de collisions élevées et des performances lamentables , les tables de routages du serveur erronées.

- Les adaptateurs réseaux, les concentrateurs, les commutateurs et les périphériques réseau en général tombent rarement en panne, mais produisent des taux d'erreurs de plus en plus élevés et/ou performances qui se dégradent au fil du temps. Une dégradation peut également être due à des câbles trop anciens.

- Les serveurs surchargés peuvent aussi avoir des temps de réponse médiocres. Un trafic trop important pour le serveur, une saturation des tampons mémoire réseaux, une bande passante des IO disques insuffisante.

- La bande passante peut aussi être insuffisante, les "Timeout [75]" et les latences peuvent être grandes, ceci peut être lié aux commutateurs ou à une mauvaise segmentation du réseau / sous-réseau ;

- La machine distante dans le cas d'une communication bilatérale peut elle-même être en insuffisance de ressources.

Comme on le remarque les raisons sont à la fois endogènes (configurations de la carte ou saturations des autres éléments) et exogènes au serveur (machine distante saturée, commutateurs et bande passante du réseau insuffisante en termes de performance…).

2.6.2 Moyens d'observations du trafic réseau

Dans cette partie, ne seront évoqués que les outils propres aux souches EDF, ou facilement implémentables à la souche ; il s'agit encore, pour notre part, de commandes.

netstat est la commande la plus triviale sur les différents Unix ; un certain nombre d'options et d'arguments en fonction des OS peuvent donner des sorties d'écrans différentes. La commande ***sar*** peut être aussi utilisée, de manière plus marginale.

✓ Autres commandes de supervision des données réseau sous Solaris

Nous avons la possibilité de voir un certain nombre de statistiques sur les couches TCP/IP *via* la commande ***netstat –s –P*** *<protocole>,* qui est utile afin de voir la congestion, les retransmissions (NB :elles doivent être inférieure à 15 %) et les paquets susceptibles d'être

[74] Nous appelons aussi ces modes HALF DUPLEX pour Unidirectionnel et FULL DUPLEX flux bi directionnel.
[75] Timeout : échéance révolue.

jetés ou dupliqués. De même l'état général des connections sockets afin de mieux optimiser l'automate. Cette commande est très prolixe.

Attention, certains compteurs sont réinitialisés indépendamment des autres (cf. tour de compteur à l'instar des véhicules auto) : les mesures doivent être faites par intermittence, notamment en corrélation avec les périodes de charges réseaux, périodes les plus susceptibles de mettre en exergue les problèmes de configuration et de performance.

De même ***kstat –m tcp***, ou des scripts à base de ***Dtrace***[76] peuvent être utilisés pour donner les métriques sur les transferts *via* les interfaces réseau ou les processus les plus impliqués dans les transferts réseau encore une fois.

Afin de mesurer les collisions réseaux, la commande ***netstat –i*** est très usitée, elle permet en général de détecter les problèmes liés au type de négociation des commutateurs (Half/full Duplex) : il faut que les équipements soient compatibles aux types de négociation des serveurs et à leur vitesse.

✓ Pour AIX

Sous AIX nous disposons de quelques outils supplémentaires pour le diagnostic, bien sur ***netstat*** peut être déclinée avec différents arguments. Les options –m et –v sont les plus intéressantes (la première se réfère aux tampons et la deuxième affiche des informations dans un mode verbeux.

Aussi ***perfmon***, ***netpmon*** ou ***netperf*** peuvent être utilisés afin de tracer l'activité réseau. Ces outils très verbeux n'ont jamais été utilisés dans le cadre d'audits sur les machines EDF pour les performances réseaux. ***topas*** fournit aussi des statiques sommaires sur le trafic réseau de manière instantanée. Surtout ***entstat*** peut fournir de nombreuses statistiques intéressantes. Ci-dessous le type de statistiques pouvant être remontées, sortie tronquée de la commande :

```
$ entstat -d ent0[77]
-------------------------------------------------------------
ETHERNET STATISTICS (ent0) :
Device Type: Host Ethernet Adapter (l-hea)
Hardware Address: 00:21:5e:51:53:a4
Elapsed Time: 9 days 13 hours 7 minutes 8 seconds

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 181980790                            Packets: 85276498
Bytes: 204676886304                           Bytes: 9129186323
Interrupts: 0                                 Interrupts: 76272131
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 4
                                              Bad Packets: 4
Max Packets on S/W Transmit Queue: 65
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 56

Broadcast Packets: 669                        Broadcast Packets: 8969411
Multicast Packets: 13650                      Multicast Packets: 339661
```

[76] Par exemple ceux développés par Brendan Gregg, ingenieur sénior illustre Solaris.
[77] En gras, les métriques les plus pertinentes.

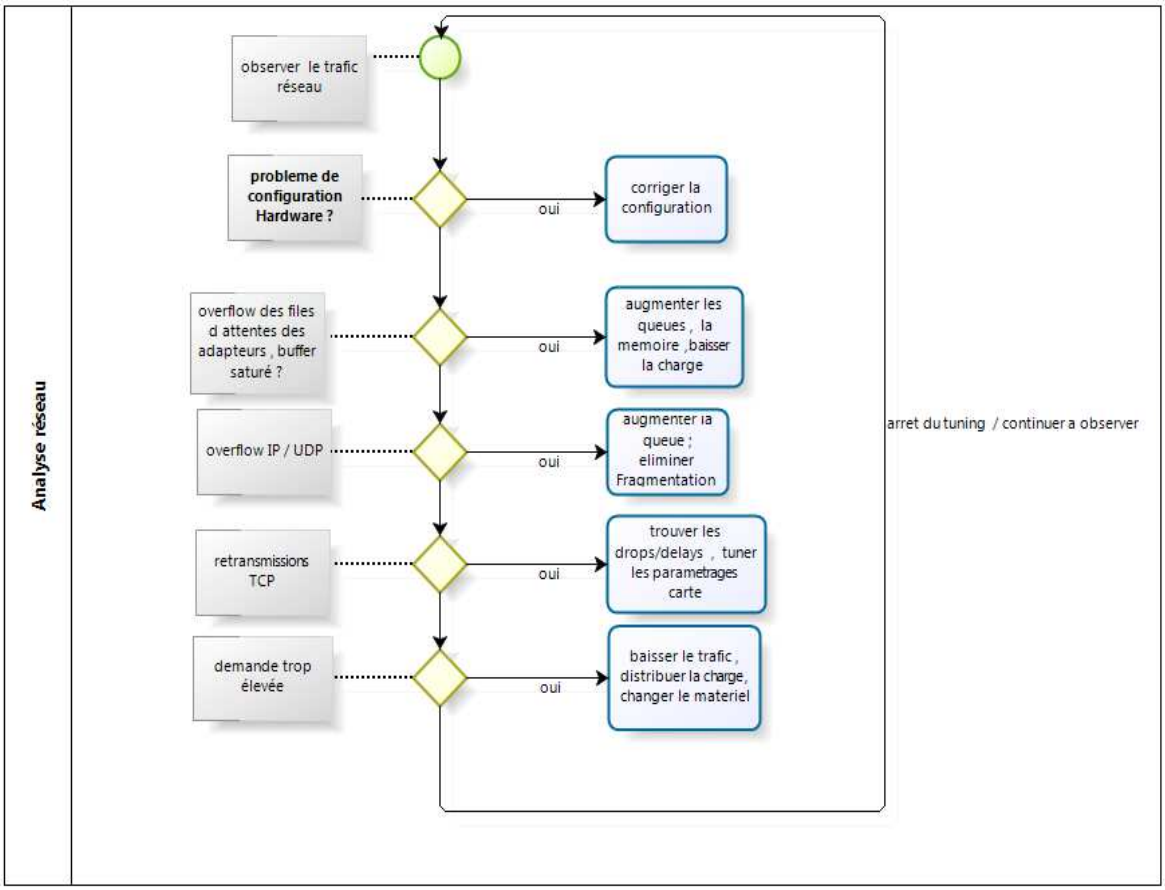
```
 No Carrier Sense: 0                          CRC Errors: 0
 DMA Underrun: 0                              DMA Overrun: 0
 Lost CTS Errors: 0                           Alignment Errors: 0
 Max Collision Errors: 0                      No Resource Errors: 0
 Late Collision Errors: 0                     Receive Collision Errors: 0
 Deferred: 0                                  Packet Too Short Errors: 0
 SQE Test: 0                                  Packet Too Long Errors: 0
 Timeout Errors: 0                            Packets Discarded by
Adapter: 0
 Single Collision Count: 0                    Receiver Start Count: 0
 Multiple Collision Count: 0
 Current HW Transmit Queue Length: 56

 General Statistics:
 -------------------
 No mbuf Errors: 0
 …/...
```

✓ Supervision et Métrologie sous Linux:

Linux, tout comme AIX et Solaris, possède les commandes ***netstat*** et ***sar***. ***Netstat*** est très régulièrement utilisé par une mise en forme de graphique au travers d'une macro Excel développée chez EDF; ***Nmon*** peut être lui aussi utilisé et l'est d'ailleurs spécifiquement en environnement virtualisé, c'est pour ma part mon outil de prédilection. Pour des informations plus complètes sur les cartes les commandes ***mii-tool*** et ***ethtool*** sont utilisées.

2.6.3 Synoptique simple du traitement de la performance réseau

On peut résumer les observations et les actions associées dans le synoptique suivant :



Les points les plus importants

Les points les plus regardés sont les paramétrages de la couche TCP du pilote de la carte réseau (la plupart des protocoles implémentent TCP), les collisions (inférieures à 5 % du nombre total de paquets émis), le nombre de paquets entrants et sortants si possible le trafic en volume d'octet et les retransmissions réseaux ; on se proposera de passer en revue cet ensemble d'éléments nuisibles à la performance dans la partie suivante.

D'autres commandes qui ne relèvent pas de la performance peuvent indiquer un problème au niveau de la topologie réseau, par exemple la commande ***ping***[78].

[78] Commande ICMP mesurant le temps d'aller-retour entre un hote distant et notre interface réseau.

Cette commande, qui peut paraître triviale, est surtout utile dans trois cas :

- Mesurer le temps de parcours afin de déceler si la requête ICMP n'est pas devenue trop longue ; tous les points de la chaîne sont ainsi testés ;
- Tester si le partenaire distant est en mesure d'être connecté au réseau (test de connectivité) ;
- Estimer le temps de parcours[79] aide à paramétrer les fenêtres d'émission et de réception des sockets TCP/UDP.

On peut ajouter à cette commande, dans un cas de support technique, l'examen d'une communication entre deux machines : ***tcpdump***, ***snoop*** (Solaris), ***iptrace*** (AIX), se placent comme outils analytiques protocolaires. Le détail de ces commandes ne fait pas l'objet de notre thème.

Les performances réseaux étant très souvent attachés à des services de la machine, on peut aussi naturellement inspecter si les services sont (ou pas) bien démarrés et bien configurés.

2.6.4 Mise en œuvre du paramétrage

✓ Configuration générale

Le bon paramétrage des cartes réseaux est indispensable : il consiste à connaitre

- Au mode de négociation (Half / Full Duplex)
- A la vitesse de transfert (10 Mb / 100 Mb / 1 Gb)
- Au modèle des cartes réseaux (carte en multiports, mise à jour des drivers)
- Le mode des cartes en redondance-agrégation (Loadbalancing[80], Failover, Bonding/Etherchannel) ;
- Nombre d'adresses virtuelles sur 1 seule carte ;
- Carte physique/virtualisée ou partagée par un serveur de ressources.

Pour ce faire, les commandes ci-dessous peuvent nous aider à établir un aperçu de la configuration de la carte :

Sous Solaris : ***ifconfig*** + fichiers de configurations sous */etc ;*

Sous AIX : ***ifconfig , lsdev , lscfg , lsattr , entstat*** ou via la base ODM[81];

[79] Exprimée en anglais par le RTT (*Round Trip Time*)

[80] "Loabalancing" désigne l'équilibrage de charge, le fail over, les mécanisme pour la résistance à la panne

[81] ODM est la base objet des périphériques et de la configuration de la machine ; une grande partie de l'administration sous AIX repose sur l'utilisation de cette base.

Sous Linux : ***ifconfig, dmidecode, mii-tool, ethtool*** + outils propres aux pilotes et informations sous `/etc` et `/pros/sys.`

✓ Paramétrage des cartes réseaux pour les couches TCP/UDP - IP

La configuration des cartes une fois connue, il est nécessaire de connaître les paramétrages mémoires liées aux drivers des cartes réseaux et les paramétrages liés aux sockets des couches protocolaire TCP/IP et UDP/IP. Ce paramétrage est en fonction du type d'activité de la machine : la configuration d'une machine serveur http avec un gestionnaire de base de données sera très différente. (NB à EDF, un script au démarrage positionne certains paramètres de la couche IP et TCP/UDP en fonction du profil de la machine.)

Pour ce faire nous avons à notre disposition les commandes suivantes :

✓ Sous Solaris

Consultation :#***ndd –get /dev/tcp*** <*nom_interface*> <*paramètre*> ou </?> par défaut

Modification:#***ndd –set /dev/tcp*** <*nom_interface*> <*paramètre=valeur*>

✓ Sous AIX

Consultation :#***no –a*** , et la base ODM voire ***entstat***

Modification :#***no –o*** <*paramètre*> ou #***chdev –l*** <*interface*> ***-a*** <*paramètre=valeur*>

✓ Sous Linux

Le contenu des fichiers `/proc/sys/net/ipv4`, `/proc/sys/net/core`, `/etc/sysctl.conf`, la commande ***systcl –w*** ou bien ***echo*** <*valeur*> > <*paramètre*>

Il est assez fréquent de tuner ces paramètres, tout du moins de les dimensionner, *via* ces commandes au démarrage de la machine par des fichiers. Le paramétrage réseau de la couche TCP peut être classé en plusieurs familles : je ne mentionnerai que les paramètres les plus fréquemment modifiés :

- Dimensionnement des fenêtres d'émission et de réception des sockets réseau et des caches : il faut pour cela avoir une idée de la latence du réseau, de la vitesse du réseau et du type de serveur.

Operating SYSTEME	paramètres
Solaris	tcp_max_buf, tcp_cwnd_max, tcp_xmit_hiwat, tcp_xmit_lowat, tcp_recv_hiwat
AIX	tcp_recvspace, tcp_sendspace, nbc_max_cache, nbc_min_cache sb_max

	rfc=1323 thewall (non modifiable)
Linux	Net.core.rmem_max , Net.core.wmem_default, Net.core.wmem_max , Net.core.wmem_default Net.ipv4.tcp_rmem , net.ipv4.tcp_wmem Tcp_windows_scaling net.core.optmem_max

➢ optimisation de TCP (libération des sockets, attente ...)

OS	Paramètres
Solaris	tcp_time_wait_interval, tcp_conn_req_max_q, tcp_conn_req_max_q0, tcp_keepalive_interval, tcp_naglim_def, tcp_ip_abort_interval, tcp_deferred_ack_interval tcp_rexmit_interval_max, tcp_rexmit_interval_min
AIX	Tcp_nodelay,tcp_nagle_limit,tcp_nodelayack Tcp_init_window Tcp_finwait2,tcp_keepidle, tcp_timewait
Linux	Tcp_fack,tcp_fin_timeout,tcp_keepalive_intvl, tcp_keepalive_probes,tcp_keepalive_time,tcp_retries1

Au-delà de ces paramètres, il est aussi possible de mettre en œuvre des algorithmes prenant en compte, par exemple, la congestion[82]. D'autres options sont possibles pour la partie UDP ; de même, un contrôle peut être fait du coté IP (notamment, contrôle du routage).

De même, à des fins de performance, changer la MTU [83] ou laisser la machine découvrir la MTU par un algorithme de découverte[84] peut apporter un gain substantiel de performance - notamment sous AIX si on utilise le Jumbo Frame MTU à 9000 octets.

Tous ces paramètres font l'objet de recommandations en fonction des applications abritées par les serveurs, ils peuvent être revus dans le cadre d'audit de performance.

2.6.5 Remarques importantes supplémentaires

✓ Un bénéfice difficilement quantifiable

Le paramétrage de la partie réseau est sans doute l'une des taches dont les effets peuvent être les moins facilement mesurables, d'une part parce que les paramétrages sont

[82] Contrôlé par *Tcp_en* sous Linux, *tcp_ecn* sous AIX.
[83] Sous Solaris Directive */dev/ip ip_path_mtu_discovery*.
[84] Ceci n'est à faire que si les équipements intermédiaires peuvent le supporter.

généralement changés en bloc alors qu'il faudrait les modifier de manière unitaire consécutivement dans le cas de tuning au cas précis.

✓ Problème de la documentation

Le paramétrage de la couche réseau n'est pas particulièrement bien documenté du point de vue des Benchmarks, et les effets sont parfois mal mesurés ou appréhendés.

✓ Forte externalité

Il ne faut pas oublier que plusieurs machines partagent les mêmes équipements réseaux, *in fine* ce qu'une machine peut gagner peut se faire au détriment d'une autre. C'est notamment le cas dans un environnement virtualisé où les médias de transmissions sont partagés.

✓ Tendance à l'auto-tuning intelligent

De même la tendance est à l'auto-paramétrage (auto-tuning) intelligent de la machine, notamment sous Linux où les fenêtres de congestion / émission et de réception s'adaptent en fonction des capacités de l'hôte distant et du réseau.

✓ Problème de la taille des paquets et du nombre de paquets comme métrique[85].

Lorsqu'on mesure l'activité réseau, il faut bien distinguer le volume en termes de paquets et en termes d'octets. Un paquet est en effet de taille variable jusque 64 Ko. A cause de la MTU[86] cette taille est limitée à 1500 octets à la sortie de l'élément réseau.

Ainsi, on a essentiellement accès (*via* les commandes usuelles de types ***netstat***) au nombre de paquets reçus ou envoyés. Le terme paquet est utilisé dans les documents de manière abusive, effectivement il est normalement le terme de l'unité de transmission au niveau de la couche 3 ou 4 du système OSI (TCP/UDP ou IP).

Or, dans la métrologie de ***netstat*** il correspond à une trame (terme de la couche de niveau 2 OSI). Un paquet TCP peut avoir une taille maximale de 64 Ko. La taille de donnée transmise à la couche liaison peut donc varier de quelques octets à 1500 octets. Il faut donc être très vigilant sur cette notion de paquets transmis. Par une règle de trois, nous pouvons déduire que le nombre maximal de paquets pour une interface à 100 Mb s'élève à environ 8000 paquets si ceux-ci sont à la taille maximale de la MTU (1500 o).

Raisonner en termes de paquets peut donc être dangereux si on ne connaît pas cette taille. La volumétrie[87] en octet est donc plus appropriée pour estimer si la carte est à sa saturation physique. Le nombre de paquets transmis est par ailleurs la seule métrique disponible sous Solaris. Pour Linux et AIX les volumétries en termes d'octet sont facilement disponibles. Ce qui nous permet, lors d'audits, d'avoir une meilleure approche.

Typiquement, une interface saturée s'observera par un plateau sur sa courbe de transmission et également par plus de retransmissions (voir étude de cas en annexe)

[85] En particulier pour Solaris.
[86] MTU (Maximum Transmited Unit) : taille maximale transmise à la couche physique de la carte.
[87] Le volume transmis est le nombre moyen de paquets transmis par la taille moyenne à un instant donné.

Les retransmissions sont aussi symptomatiques d'autres problèmes:

- Latence du réseau ;
- Saturation des tampons réseaux du partenaire ;
- Congestion réseau ;
- Equipement réseau défectueux.

2.6.6 Préconisations générales en terme de tuning réseau et mise en place chez EDF

- Vérifier la configuration réseau de la machine
- Surveiller les collisions qui mettent en exergue un problème de négociation ;
- Surveiller la saturation de la machine (paquets et volume en Kilo octets) ;
- Désactiver les services réseau inutiles ;
- Paramétrer la couche TCP/UDP et les fenêtres d'émission et de réception au besoin, et l'automate TCP/UDP si des métriques indiquent une anomalie ;
- Appliquer si nécessaire les recommandations des éditeurs de logiciels ;
- Tuner la carte réseau en fonction du profil applicatif de la machine.

Nous proposons des paramétrages réseaux de la couche TCP/IP en adéquation avec les activités de la machine, de même des scripts au démarrage de la machine permettent en fonction du profil de la machine d'ajuster des paramètres plus justes ; ces profils s'appliquent à des machines avec pour profil : standard (par défaut) ; serveur web ; serveur de base de données.

3 Considérations techniques supplémentaires et Outils.

Le début de ce troisième chapitre se fera dans le prolongement du deuxième afin de compléter certains points essentiels, ce qui concerne un certain nombre d'implémentations :

- Le micro-partionnement CPU ;
- La virtualisation mémoire,
- Virtualisation des carte fibres et réseau.

En effet ces implémentations viennent invalider ou changer un certain nombre de remarques indispensables dans le cadre du tuning.

3.1 Techniques de partionnement et micro-partionnement des CPU et virtualisation des processeurs sous AIX

3.1.1 Généralités sur la virtualisation

L'une des dernières évolutions en termes de technologie visant à optimiser l'utilisation des ressources informatiques est la virtualisation de serveur, c'est-à-dire sur une même machine physique pouvoir définir plusieurs partitions chacune comportant un système d'exploitation. Ces partitions sont logiquement étanches unitairement, et partagent un certain nombre de ressources physiques.

Chez EDF, un grand nombre de machines sous AIX comme sous Linux mettent en œuvre des mécanismes de virtualisation de ressources physiques.

Cette technique, à l'instar de la mémoire virtuelle, laisse penser que la partition dispose de la ressource physique uniquement pour elle ; ces ressources sont la mémoire, la CPU, les cartes Ethernet et la carte fibre optique pour les disques de baie. Les mécanismes de virtualisation lui sont masqués, et seul l'hyperviseur (gestionnaire de ressources) a connaissance du partage ou de la virtualisation de la ressource. Coté performance, pour toutes choses égales par ailleurs une machine purement physique aura des performances souvent supérieures par rapport à une machine virtuelle.

3.1.2 Le micropartionnement

Nous le savons, un des éléments les plus onéreux du système est le processeur ; ceux-ci sont de plus en plus complexes et puissants, et les derniers modèles permettent justement de tirer parti de ces mécanismes de virtualisation. Il est donc normal de vouloir optimiser leur

utilisation[88], ce qui pourrait se résumer par :"N'utiliser que la puissance CPU nécessaire à la demande pour l'exécution des tâches."

Vu précédemment un système parfaitement optimal devrait être un système pour lequel il n'y a pas de gaspillage de ressources.

Au sein d'EDF, nous disposons donc des dernières techniques de virtualisation en ce qui concerne les processeurs POWERPC avec les technologies PowerVM et aussi le logiciel VmWare pour la partie processeur INTEL.

Dans le partitionnement logique de machine[89] pour AIX (LPAR) il est possible de partager des ressources tout comme de les dédier à des machines virtuelles. Aussi est-il donc possible d'attribuer à minima 10 % du temps d'exécution d'un processeur physique au fonctionnement de la partition. Ce quantum de temps CPU physique est donc vu comme une CPU virtuelle (VCPU). Ce découpage d'une CPU physique entre partitions logiques est, chez IBM, appelé micro-partionnement.

Si les fonctionnalités de multithreading sont activées, le système pourra également avoir à sa disposition plusieurs files d'exécution simultanées dénommées CPU logique. Bien entendu tout ce mécanisme de partage de ressources est pris en charge par un hyperviseur comme Power Hypervisor pour IBM mais il en existe bien d'autres : Xen, HyperV(MS), KVM, Vsphere (VmWare) pour les plus connus. Un hyperviseur s'assure de l'allocation du contrôle, du partage, de la protection des ressources entre les partitions.

3.1.3 Partage de CPU dans l'univers AIX : Mode de fonctionnement

Connaître le mode de fonctionnement de la partition est un pré requis nécessaire lorsqu'on étudie la saturation CPU en environnement virtuel.

En effet il existe deux modes de fonctionnement d'une partition : une partition partagée pour laquelle les ressources sont attribuées dynamiquement à partir d'un pool de processeurs partagé (mode Shared) par les différentes partitions, et les partitions dites dédiées qui peuvent ne pas partager (mode Dedicated) certaines ressources, ce dernier mode rejoint celui dans un système physique et dispose de meilleure performance (moins de latence et meilleure affinité).

Pour les partitions le mode shared est le plus utilisé sur les LPAR EDF, la CPU peut être en mode capped ou uncapped (bridé ou non bridé).

Dans le premier cas, ce montant d'unités de calcul ou Unit Processing n'est jamais dépassé. C'est une ressource CPU dite garantie ; cela constitue donc l'allocation maximale en CPU [90]que la partition peut se voir attribuer, pour le mode non bridé, il s'agit d'une limite qui peut être dépassée si nécessaire. Cette CPU dite garantie est de facto la réservation qui sera faite pour la partition. Le total des CPU garanties de toutes les partitions ne peut dépasser le nombre de CPU mises à disposition pour le partage, évitant ainsi la sur-allocation.

---

[88] Une CPU est en moyenne utilisée à 12 % sur des machines physiques.

[89] On utilisera parfois son acronyme LPAR (Logical PARtition) pour des facilités d'expression.

[90] Cette limite peut être obtenue en ayant connaissance via les commandes usuelles *vmstat* ou *lpstart* et prend comme dénominatif E.C ou Entitled Capacity.

Ainsi si la partition sollicite davantage de temps puissance de calcul, celle-ci peut effectivement dépassée cette limite garantie, le maximal d'allocation CPU sera atteint par le nombre maximal de CPU en ligne visible par la Partition si et seulement si l'Hyperviseur peut attribuer ces temps processeurs autrement dit si les autres partitions ayant accès à ce pool ne sont ni fortement consommatrices ni plus prioritaires. Cette notion de priorité est définie par une notion de poids .Cette notion de poids peut être mis en avant si on veut favoriser des LPAR critiques ou les I/O serveur (VIOS[91]).

Il est donc très important de connaître ces modes de fonctionnement ce qui est visualisable par la commande ***lpartstat –i***

Le dimensionnement pour les besoins de calcul doit se faire sur un nombre d'unité de processeur raisonnable. Une attribution d'un nombre CPU trop élevée peut augmenter les commutations de contexte entre les CPU si elle est trop basse de ne pouvoir bénéficier des cycles CPU non utilisées par les autres partitions.

Afin d'éviter les migrations de contexte trop importante entre les CPU, les CPU ont un mode de fonctionnement favorisant l'affinité ainsi les processus tendent à s'exécuter sur les mêmes processeurs afin de maximiser l'utilisation du cache processeur. (En annexe un exemple démontrant l'affinité de CPU).

In Fine le système de virtualisation des processeurs permet de mieux optimiser l'occupation des processeurs donc d'être en mesure de moins gaspiller les ressources processeurs. Ainsi il permet à des partitions avec une charge en générale faible d'absorber des pics de charge sur plusieurs processeurs, ils sont donc préconisés dans le cas d'applications sur des systèmes transactionnels (OLTP), serveur WEB, SMTP ou serveurs de fichiers et de pouvoir redistribuer leur puissance à d'autres partitions.

A contrario le micro-partionnement est préjudiciable dans le cas de système Décisionnel ou de calcul haute-performance.

3.1.4 Incidences sur les audits de performance :

Jusqu'à présent lors des audits aucune partition en mode dédié donatrice n'a été vu.

Ce qui est utilisé chez EDF sont les partitions en mode shared uncapped (cf partagé et illimité).

Ne sera disserté que les partitions dans ce mode, à la vue de ce qui a été expliqué ci-dessus il est aisé d'imaginer les conséquences en terme de supervision de la performance.

N'utiliser que ce que la partition peut avoir besoin, remet en cause l'interprétation des métriques CPU. En effet l'occupation optimale du temps processeur aura pour effet de réduire le temps libre ou non utilisé pour les taches (appelé *idle time*) donc d'avoir des occupations faciales pour la partie utilisateur et système très élevées.

Voici un exemple de ce que l'on peut obtenir en termes de supervision par ligne de commande :

[91] VIOS VIRTUAL I/O Server, il s'agit de partition spéciale mutualisant et virtualisant l'utilisation des disques, cartes Ethernet et carte fibre.

```
System configuration: lcpu=8 ent=2.00 mode=Uncapped
08:30:00    %usr    %sys    %wio   %idle   physc   %entc
08:30:10      90      10       0       0    2.58   129.1
08:30:20      90       9       0       0    2.85   142.6
08:30:30      91       9       0       0    2.61   130.6
08:30:40      91       9       0       0    2.38   119.2
08:30:50      90      10       0       0    2.77   138.3
08:31:00      90       9       0       0    2.84   142.0
08:31:10      90       9       0       0    2.76   137.8
08:31:20      91       9       0       0    2.78   138.9
08:31:30      91       9       0       0    2.79   139.6
08:31:40      90       9       0       0    2.71   135.7

Average       90       9       0       0    2.71   135.4
```

Explication des données visualisées:

Ci-dessus, la CPU est occupée à 90 % par des applications et de 9 % par le système, la CPU garantie dépassée de 35 % donc 2.7 CPU consommées (*physc*) mais 4 CPU physiques sont disponibles, 1 CPU étant égale à 2 CPU logiques (le mode SMT2[92] étant activé).

Nous pourrions donc en effet au premier abord affirmé que les ressources CPU sont saturées mais il n'en n'est rien au final. La véritable limite maximale se situe à 200 (%*entc*)

Ce qui est donc maintenant important est de voir si les limites allouées par la configuration au niveau de l'hyperviseur sont dépassées ou pas, donc si la puissance CPU maximale allouable devient insuffisante ou pas.

Une saturation CPU s'observe par un graphique dont la courbe plafonne, cette puissance CPU est limitée par la valeur : *VCPU Online* (visible via la commande ***lpartstat –i)***.

Exemple de sortie de commande pour une machine limitée à 4 VCPU maximale saturée: observez la colonne *r, pc* = puissance consommée) et *ec* (puissance attitrée).

```
$vmstat 5

kthr    memory              page              faults              cpu
----- ----------- ------------------------ ------------ -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec
25  0 956477 496217   0   0   0  25  160   0 1130 58738 176525 61 39  0  0  4.00 400.0
25  0 956473 496229   0   0   0  25  196   0 501 61539 174070 62 38  0  0  4.00 400.0
26  0 956467 496243   0   0   0  25  153   0 616 57107 173753 61 39  0  0  4.00 400.0
25  0 956486 496276   0   0   0  51  306   0 560 55951 170885 60 40  0  0  3.99 399.1
25  0 956472 496195   0   0   0   0    0   0 550 56074 176324 60 40  0  0  3.99 399.4
24  1 956476 496231   0   0   0  51  213   0 733 57764 175186 61 39  0  0  4.00 400.0
25  1 956474 496202   0   0   0  25  101   0 514 58774 177235 61 39  0  0  4.00 400.0
26  0 956473 496185   0   0   0  25   88   0 657 56890 175736 60 40  0  0  4.00 400.1
27  0 958241 494620   0   0   0  77  565   0 592 56836 176094 60 40  0  0  4.00 399.8
25  0 956399 496364   0   0   0   0    0   0 1229 80905 171675 60 40  0  0  4.00 400.1
24  1 956413 496226   0   0   0   0    0   0 439 56560 176016 60 40  0  0  4.00 400.0
25  1 956656 496075   0   0   0  51  862   0 610 57185 175805 60 40  0  0  4.00 399.8
23  0 956411 496226   0   0   0   0    0   0 560 56109 176774 60 40  0  0  4.00 400.0
```

[92] SMT = Simultaneous Multi Threads , mode de fonctionnement permettant l'exécution en parallèle de plusieurs fils d'exécution sur le même cœur processeur. Ici SMT2 permet 2 fils d'exécution par cœur. Le SMT peut permettre un gain de 40% des performances pour une application fortement multithreadée.

3.2Autres systèmes de Virtualisation utilisé pour la CPU

3.2.1 Allocations des CPU via ESX – VSPHERE pour les partitions Linux

L'hyperviseur ESXi ou Vsphere (nouvelle dénomination donnée par VmWare) est aussi utilisé pour virtualiser ces partitions. Sous Linux, l'allocation des CPU obéit à la loi du tout ou rien, soit il alloue le total de la puissance CPU demandée durant un temps imparti, soit il alloue juste le minimal requis pour son traitement des routines systèmes de base. Pour supposer d'une contention au niveau de la ressources CPU, il faut dans ce cas mesurer le temps d'attente pour obtenir cette ressource complètement.

On le remarque aisément, en absence de pondération de priorité, l'hyperviseur a donc plus de chance de servir les partitions dont les besoins sont les plus faibles. A EDF à ma connaissance cette pondération n'est pas mise en place sur les serveurs d'application.

Cette politique désavantagera donc les partitions dont les besoins sont les plus importants, ce qui signifie qu'en général les partitions les plus actives en terme de consommation ou bien terme d'importance fonctionnelle sont pénalisées, ce qui va à l'encontre de l'effet désiré.

De plus en plus de partitions Linux hébergent des applications critiques notamment par le fait qu'au sein d'EDF ce qui était hébergé sur des machines Sun le devient sur des solutions Linux avec processeurs INTEL.

Il est à noter cependant qu'une VCPU correspond à un thread CPU pour les versions 4.1 VSPHERE (en phase finale de déploiement) et un Cœur CPU pour ESX 3.5 (solution historique en voie de disparition à EDF) du fait de l'activation de l'Hyperthreading. Le facteur démultiplicateur de performance est en fait de 1,2 pour un cœur de calcul.

L'Hyperthreading ne donne pas de gain de performance dans des applications dites « CPU Intensive ». Le VMkernel essaie par ailleurs de jouer l'affinité de CPU afin d'éviter des migrations de contexte.

L'Hyperthreading s'active au niveau du BIOS[93] et découle par ailleurs de la reconnaissance matérielle de VSPHERE.

Le dimensionnent des VCPU dépend de la charge applicative et doit donc s'ajuster au plus juste afin de ne pas tomber dans une trappe à attente. A EDF le surbooking CPU est de 2. C'est à dire que l'Hyperviseur offre deux fois plus de CPU qu'il n'en dispose.

L'attente de CPU peut être donc le fait de :

- Du nombre de VM sur le châssis (plus le nombre est grand plus il faut partager)
- Trop forte réservation CPU (loi du tout ou rien).
- Surbooking trop élevé
- L'utilisation globale de la CPU est trop élevée[94].

[93] Microcode pour la prise en charge matériel des architecture Intel.
[94] Ce qui augmente le « ready time » , temps de mise à disposition de la ressource CPU

- Les charges des autres VM sont spatio-temporellement trop corrélées.

3.2.2 Cas des plateformes SUN.

Il ne sera point nécessaire de trop développer sur le sujet car EDF a opté pour l'abandon des solutions proposées par Sun-Oracle en Novembre 2010 pour la dépendance trop forte entre la solution logicielle Oracle et la plate-forme matérielle; aussi les technologies de virtualisation proposés par les *Logical DOMain* (LDOM) ou les *Containers/Zone* dans le cadre de la consolidation n'ont jamais convaincu EDF.

D'une part les LDOM ne peuvent être implémentés que sur une certaine catégorie de machines [95] et d'autre part les processeurs des stations SUN étaient en retard en termes de fonctionnalités [96] et de puissance CPU. La machine la plus puissante de la série T, la seule a implémentée les LDOM, est le T5440 qui possède 2 processeurs physiques cadencé à 1.6 GHz. (Ndt : les derniers PowerPC sont de l'ordre de 3-4 GHz).

Ce type de matériel et de technologie restent très approprié à des serveurs Web.

Aussi un autre élément a prédisposé à son refus d'implémentation au sein d'EDF: des problèmes de compatibilité avec les baies de stockage.

✓ Pour la solution des Zones (solution de consolidation):

Dans le cas des containers ou plus communément appelé zones, il est possible via une partition maîtresse jouant le rôle d'Hyperviseur et de système hôte de créer des partitions filles de différents types (Solaris 10, 8 ou Redhat). Le système maître sous traitera les demandes faites par les partitions sous-jacentes, il mettra donc à disposition ses ressources logicielles et matérielles. En terme de CPU, le micropartionnement se fait par nombre de threads/core, le cloisonnement des partitions étant moins drastique, on aura donc la possibilité de n'utiliser que la puissance CPU nécessaire (le nombre de threads nécessaire par exemple) le reste de la puissance restant disponible pour les autres partitions, encore une fois le monitoring du système maître rendra véritablement compte des contentions liées à la gestion des ressources.

Cette solution a cependant des avantages en terme de maintenance pour des systèmes d'exploitation anciens tel que Solaris 8, peu ou plus maintenus sur du matériel plus récent. Ainsi certaines applications permettent de finir leur fin de vie car elles ne sont pas développés ou optimisées sur les plate-formes matérielles plus récentes.

3.3 Virtualisation des cartes Fibre pour le SAN et Ethernet

[95] il s'agit des machines T Series avec la technologie Coolthread

[96] le micropartionnement existe mais se décline à la file d'exécution possible dans un cœur, c'est a dire 32 files pour un cœur . Ainsi nous pouvons donc attribuer 1/32 de CPU a une partition logique.

3.3.1 Virtualisation des cartes fibres

Sous les systèmes Unix notamment pour AIX avec la technologie NPIV, on a la possibilité de partager la même carte fibre gérée par le VIOS[97] qui ne joue le rôle que de pont. Il va de soi qu'un accès logiciel doit être nécessaire pour la partition pour cette ressource. D'un point de vue performance bien qu'étant mis en place dernièrement à EDF, je ne détaillerai pas ce point sachant que nous entrons dans un partage de bande passante et qu'il est aisé de comprendre le type de problématique pouvant découdre de cette technique. Il a été testé par les architectes que la carte fibre peut être partagée par 8 LPAR de manière relativement optimale. Les cartes mettant en jeu ce mécanisme ont des capacités de 8 Go/s. Pour ma part je n'ai pas eu de problématique de performance liée à NPIV à la rédaction de ce document, je n'entrerai donc pas plus dans le détail.

3.3.2 Virtualisation des cartes Ethernet

A l'instar de ce qui est fait pour les cartes fibres, il est possible aussi de faire le partage d'une même carte Ethernet pour plusieurs partitions ou zones. Chaque partition a donc accès à la capacité de transmission de la carte qui souvent de l'ordre du Gigaoctet. Afin d'allouer suffisamment de bande passante à une partition des mécanismes de QoS[98] peuvent être mis en œuvre afin de garantir un minima de transmission à chaque partition.

Depuis la partition nous avons donc exactement les mêmes limitations en termes de monitoring sur le trafic de la carte, il faut aussi donc se positionner au niveau de la partition partageant ces ressources physiques: le VIOS. Ainsi une mauvaise configuration au niveau de l'IO serveur peut porter préjudice à l'ensemble des partitions filles, c'est un cas que j'ai par ailleurs abordé au cours de mon travail à EDF, ce qui m'a permis de détecter cette anomalie était le nombre d'erreurs de la carte virtuelle et la saturation relative pour des niveaux de transfert trop bas. Il convient donc d'être particulièrement vigilant sur la configuration de la partition faisant office de IO Server et des performances collectées sur cet élément , d'autre part cette partition « intermédiaire » doit toujours disposer de ressources physiques nécessaire à son bon fonctionnement, par exemple une carence mémoire sur le VIOS engendre la perte de visibilité des disques aux partitions.

3.4 Virtualisation de la Mémoire

Cette partie ne traite pas de la mémoire virtuelle vis à vis de la mémoire physique, notion déjà vue précédemment dans la deuxième partie, la mémoire virtuelle est un concept déjà utilisé depuis des décennies.

Ce paragraphe vise à présenter une implémentation déjà en place sur Linux via l'utilisation de VmWare via son hyperviseur ESX qui consiste à partager d'un pool physique de mémoire entre plusieurs partitions de manière totalement dynamique. Un certain nombre de machine Linux à EDF fonctionnent sous VmWare, les problèmes liés à la mémoire ne sont pas inconnus à EDF.

---

[97] Cette technique de virtualisation de la carte fibre sous AIX prend le nom de NPIV

[98] mécanisme de qualité de services donc de garantie de bande passante.

L'AMS, l'implémentation IBM du partage mémoire inter partitions, ne sera point détaillée car encore non exploité pour le moment à EDF. Aussi les principes de base de l'AMS sont assez voisins de la virtualisation mémoire sous VmWare.

3.4.1 Techniques sous VmWare

C'est à l'hyperviseur de VmWare ESX (ou Vsphere) d'attribuer et gérer la mémoire qu'il va fournir aux Machines Virtuelles, appelées Virtual Machine ou VM.

Afin de compléter le mécanisme de partage mémoire et pour permettre éventuellement la sur-allocation de mémoire, l'hyperviseur ESX-VSPHERE utilise 3 techniques de manière concurrente et complémentaire en fonction du "stress" mémoire. En effet l'hyperviseur dispose à priori moins de mémoire physique que la somme attribuée aux VM filles. C'est le principe de la sur-allocation.

Pour ce faire il utilise de manière complémentaire et éventuellement nécessaire.

- Le transparent page sharing (TPS) :

Certaines pages identiques ne seront dupliquées entre les OS, il faut noter à cet effet qu'il y a une forte probabilité que les OS virtuels aient le même système d'exploitation ce qui facilite ce type de technologique.

Le principe consiste à scruter la mémoire, de ne garder une copie et de faire pointer les VM vers cette page et de libérer les doublons.

Cette technique est transparente pour les VM. C'est en quelque sorte le principe des bibliothèques partagées dans un OS ramené à l'échelle de la RAM et de la page entre plusieurs VM. Il est possible de suivre cette implémentation par la métrique Shared Common sous le Vcenter (les pages étant par défaut de 4 Ko).

Ce mécanisme prend son efficience lorsque le fonctionnement de la VM est déjà effectif depuis un certain temps.

- le Ballooning :

C'est une technique de prêt de mémoire inter-partition qui nécessite l'installation d'un VMWareTool *vmmemctl*. Une partie de la mémoire est moins sollicitée en moyenne dans chaque VM, pour permettre de gonfler le ballon, le driver *vmmemctl* va donc identifier les pages éligibles au prêt et donc contraindre l'OS à libérer de la mémoire grâce ses propres algorithmes de gestion de la mémoire si nécessaire. On va effectivement fictivement faire pression sur la mémoire de la VM afin d'obtenir cette mémoire pour la fournir via l'hyperviseur à une autre VM créant ainsi une pagination de sortie, la déflation du ballon crée le mouvement antagoniste (du page-in)

La machine virtuelle (VM) n'a pas connaissance du but du *vmmemctl* (driver de balloning) qui est de fournir des pages et au Vmkernel.

Attention cette technique est valable dans le cas où la mémoire est sur-allouée par l'ensemble des VM sur la capacité physique de prêt de l'hyperviseur.

- le Hard Swapping

Les besoins mémoires sont gérés au niveau de l'hyperviseur qui dispose d'un paging space distinct pour chaque VM afin d'y loger éventuellement et si besoin entièrement les pages d'une VM; ainsi les pages copiées sur cet espace peuvent être libérées et donc attribuées à une autre VM. Cette éviction ne fait pas de distinction entre les différents types de pages déplacées vers le disque de pagination. Ce mécanisme est le plus radical et le plus pénalisant en termes de performance.

De ce fait déceler du *hard swapping* au niveau de l'hyperviseur signifie un gros problème de mémoire.

De nombreuses fois, l'équipe ASC a été sollicité pour diagnostiquer des problèmes mémoires sur les VM ce qui ne peut se faire avec une vue sur les ressources et la métrologie de l'hyperviseur de la machine physique via l'interface du Vcenter.

En effet cette analyse doit se faire sur la VM mais également sur l'hyperviseur.

A EDF , toute contention sur une VM doit être signalée aux équipes en charge de l'infrastructure VmWare c'est une autre prestation d'ingénieurs au sein de EDF. Les phénomènes de contention mémoire se traduisent sur la VM par des temps de réponse plus faible, une dégradation de performance et une utilisation du disque de swap de la VM même. En effet la VM n'a pas connaissance de la nature de la mémoire qui lui est prêtée par l'hyperviseur, cela est soit de la RAM physique complétée par une part plus ou moins importante d'une partie disque (donc avec un accès beaucoup plus lent).

3.5Outils d'aide aux audits de performance.

Jusqu'à une date récente depuis fin 2010, tous les rapports d'audits étaient fournis grâce à l'analyse faite de la collecte de sondes "maison" en partie , de NMON et aussi grâce d'autres logiciels déjà en partie évoqués en première partie plus marginalement.

Cette partie traitera donc de la présentation de ces outils de manière plus formelle, de leurs forces et de leurs insuffisances et enfin d'Omnivision Investigation, outil choisi par EDF et récemment mis à disposition aux ingénieurs systèmes de la cellule afin de répondre pour le suivi de la performance des systèmes.

Dans le cadre de notre activité nous avons donc besoin de collecter des données dans le cadre d'audit de performance de manière ponctuelle pour y exercer une action corrective mais aussi de produire des indicateurs de performances de l'infrastructure pour une action préventive.

Dans le cadre d'une réflexion sur l'optimisation des processus de collection et d'archivage de donnée, il semblait important que cet outil remplisse ces critères

- Simplicité de mise en place sur les serveurs
- Uniformité des procédés entre les plate-formes

- Centralisation relative de l'archivage et d'accès à l'outil d'analyse.
- Grand nombre de données consultables

3.5.1 Etude des Outils existants au sein de la cellule.

✓ Utilisation des scripts de collecte et mise en forme via Excel.

En ce qui concerne la partie UNIX, nous utilisons des scripts de collecte de statistiques de données système qui peuvent être archivées sur la machine elle-même avec une durée de rétention à définir et de 15 jours par défaut, ces scripts sont au nombre de deux : ***perfstat.sh*** (script « maison » tout UNIX en annexe) et ***Nmon*** (script maintenu par IBM disponible pour AIX et Linux).

A la marge il est possible d'écrire en Shell des scripts qui peuvent collectés des informations non couvertes par ces scripts.

Ces scripts constituent principalement la source de nos données d'analyse sur la problématique de performances.

Ces scripts doivent donc déposés sur les machines et activés par les outils d'ordonnancement système propre à chaque compte tel que ***crontab***.

La sortie de ces scripts donne lieu à des données de type ASCII qui pourront être converties en .csv format utilisé par le logiciel Excel.

Chaque script subit à une adaptation prenant en compte le type d'OS, ce qui est résolu par des tests et des sélections de cas interne au script [99], cependant toutes les sorties des commandes de statistiques ne sont pas toutes uniformes au sein d'un même système d'exploitation, ce qui est particulièrement le cas pour les distributions Redhat Linux.

De ce fait la macro Excel doit aussi être adaptée pour prendre en compte cela.

Ces scripts ont l'avantage par ailleurs d'être ouverts et intelligibles pour des modifications pour ajouter au besoin des données de collecte supplémentaires ce qui arrive parfois dans le cadre des outils menés dans la cellule.

Cependant ces scripts sont d'une part pas systématiquement déployés et très sensible à la maintenance des infogérants, les scripts en ***crontab*** peuvent désactivés de manière arbitraire, de même la présence de ces scripts n'est pas surveillée par des outils tierces. Ainsi une machine dont on a la charge de faire l'audit a une chance de ne pas disposer au moment de la demande d'audit des sondes adéquates afin de faire le diagnostic dans le passé du problème de performance supposé c'est le cas du script ***perfstat.sh***

✓ La sonde « maison » : *perfstat.sh*

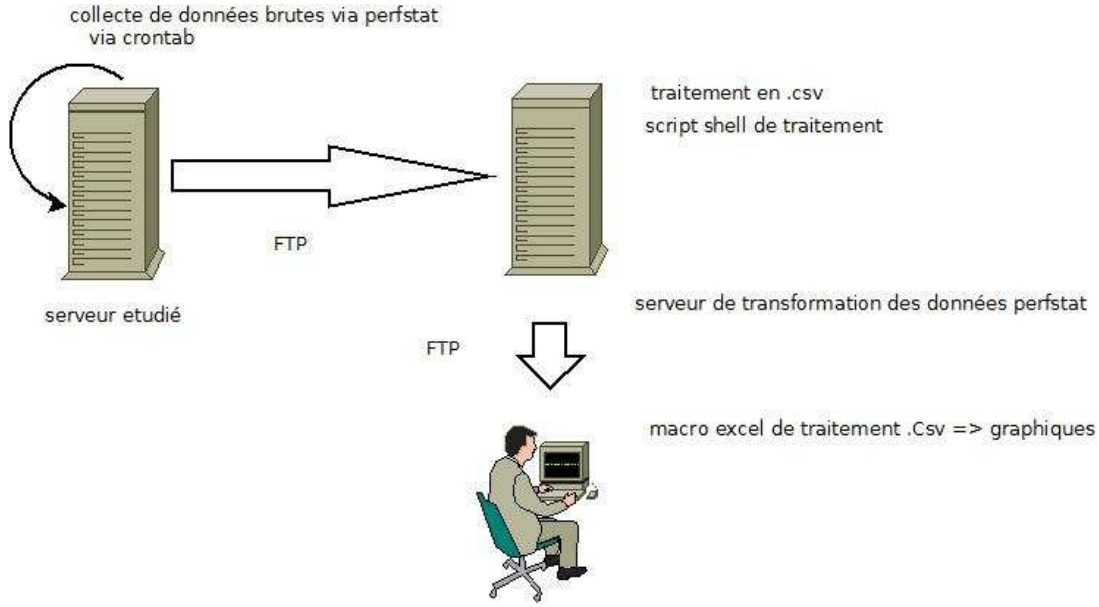
Il s'agit d'un script Shell collectant les données à la journée, les données traitées en .csv puis intégrées sous Excel. La macro Excel génère un fichier graphique journalier par composant,

[99] Le script réalisant le relevé de statistiques en mis pour information en Annexe , il s'agit d'un script Shell Unix

les entrées doivent parfaitement cohérentes et intègres. Les sondes ne sont activées qu'à minuit, ce qui suppose que les redémarrages de serveurs arrêtent pour le restant de la journée les sondes et tronquent ainsi les données et leur traçage. (cf. annexe pour exemple).Une prise de mesure est faite par minutes ainsi on totalise 1440 mesures par jour pour chaque commande lancée.

Dans ce dispositif archaïque, il n'y aucun serveur qui centralise le rapatriement des données en automatique prises à partir des serveurs, les transferts s'opèrent en FTP vers une machine dédiée au traitement des données et des scripts convertissent d'un format ASCII vers un le format CSV de Excel.

Schéma fonctionnel technique :



Les données collectées sont archivées quotidiennement. La supervision en temps réel n'est possible que par une ou plusieurs sessions d'exécutions de commandes.

✓ Outil NMON

Nous avons également déjà évoqué NMON, il a été développé par un ingénieur d'IBM - Nigel Griffith, il subit à peu près les mêmes contraintes que ***perfstat*** en terme de collecte/centralisation et de dépose aux différences suivantes :

- Les scripts de collecte et se déclinent en fonction des versions logicielles, il est donc pas nécessaire de systématiquement redéveloppé la sonde et la macro Excel.
- Les métriques les plus importantes sont mises en valeur et présentent les faits de manière intelligibles pour les ingénieurs systèmes et lecteur des audits.

- Pas de problème au niveau de l'horodatage même dans le cas d'un fichier tronqué sur la durée d'exécution de la sonde, les données sont horodatées par le binaire
- Les processus les plus consommateurs sont aussi reportés succinctement.
- Les configurations systèmes sont reportées quotidiennement.
- Nécessité d'être super-utilisateur (root) pour la mise en place de la sonde.
- La sortie de données plus volumineuse, il y a nécessité de purger les données trop anciennes.

Tableau synthétiques des avantages et inconvénients des sondes Perfstat et NMON

AVANTAGES	INCONVENIENTS
- Script modifiable (Perfstat) - Données collectées non dépendante d'un SPOF[100] de l'infrastructure - Format des graphiques - S'appuie sur les commandes systèmes les plus connues et usitées. - Rétention des données réglable - Format CSV facilement utilisable	- Maintenance de la sonde pour son obsolescence - Pas de centralisation des données - Déploiement de la sonde - Fonctionnement conditionné aux aléas d'administration de la machine - Pas de suivi temps réel (perfstat) - Maintenance de la macro Excel ardue - transfert manuel des données de collecte - Besoin d'être root pour déposer/récupérer - Pas de suivi consommations des processus le plus gourmands (Perfstat)

3.5.2 Outils existants d'aide à la performance issus d'éditeurs de logiciel.

✓ PATROL

[100] Single Point Of Failures , point de faiblesse , il s'agit de nœud de l'infrastructure pouvant présenter mettre en péril le fonctionnement

Logiciel édité par BMC, celui-ci permet de remonter des alertes sur chaque serveur comportant un agent Patrol déployé. Ces alertes sont activées par des seuils de déclenchements préalablement établis à l'aide du DEX [101]sur une console (serveur) dédiée à cet effet, chaque infogérant possède une console attitrée. Non seulement Patrol permet la supervision d'une machine mais également permet de suivre la consommation CPU, Mémoire, occupation des disques, trafic réseau, et consommation de certains processus sur une durée passée de 15 jours ou en temps réel sous réserve que les modules soient activés sur l'hôte.

L'ergonomie et la facilité d'utilisation du produit sur la partie suivi de performance peuvent être d'une grande efficacité lorsqu'il est demandé à notre équipe d'expertise de suivre les performances de la machine en temps réel. Patrol permet notamment lorsque l'outil est installé au préalable de la sonde ; la possibilité d'avoir un historique des consommations CPU/Mémoire. L 'infrastructure Patrol est gérée par une équipe dédiée, cependant les accès aux consoles est limitée et authentifiée, chaque infogérant gère sa propre liste d'auditeurs.

Patrol est un outil pouvant être installé sur les machines Windows et Unix.

AVANTAGES	INCONVENIENTS
- Interface Web de Consultation ergonomique - Déploiement généralisé sur la plus grande partie du Parc - Echantillonnage à la Minute - Modules optimisables en fonction des serveurs - Centralisation des Informations - Suivi des processus les plus consommateurs définis - Supporte l'hétérogénéité du Parc	- Mode de stockage propriétaire, impossibilité d'avoir accès aux Données - Accès restreint a une personne par profil - Autant de profil/console que d'infogérants - Pas de suivi de performance des disques - Graphiques non exportable - « Développement » des modules a la charge de l'équipe Patrol - Client léger faisant preuve de lenteur - Données remontées parfois en contradiction avec les sondes de statistique de la machine (pas toujours fiable).

In fine Patrol est un outil d'appoint pour le travail sur les audits de performances et se révèle parfois précieux lorsqu'on ne dispose pas des données collectées par les sondes perfstat ou Nmon. Cependant la finalité du produit n'est pas l'étude de la performance d'où la certaine pauvreté du produit sur cette thématique. D'autre part la fiabilité de ce produit sur le suivi de la consommation mémoire est parfois sujet à caution.

[101] DEX : Dossier d'EXploitation

✓ OMNIVISION

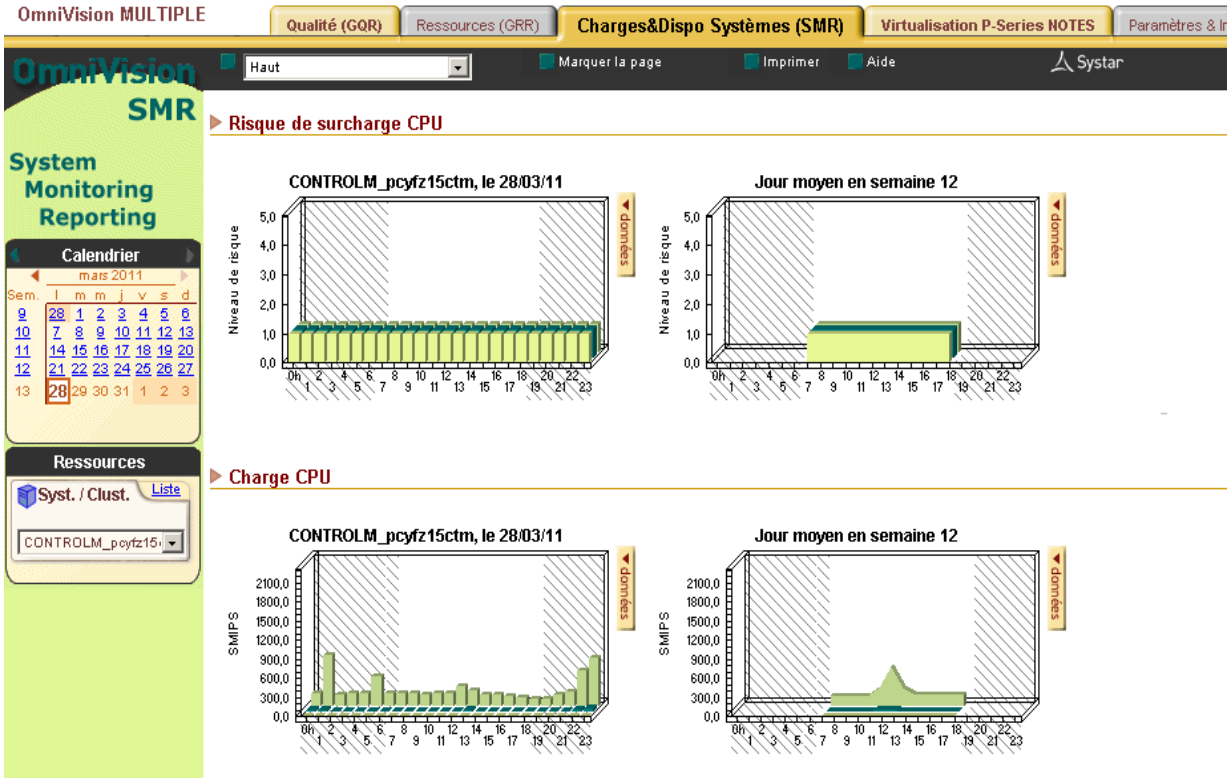
Outil de monitoring et de Capacity Planning développé par SYSTAR, il est surtout populaire et utilisé pour son approche synthétique au près des Charges d'affaires.

C'est une IHM accessible par une URL intranet, les machines les plus critiques du parc disposent d'un agent Omnivision chargé de la collecte des données.

Il comporte un résumé de la configuration du serveur (ceux-ci sont regroupés par application) et des graphes en forme de bâtonnets vu d'une manière horaire, les statistiques sont collectées à la seconde mais agrégées pour 20 secondes.

A l'heure on a donc accès : au minima, maxima et à la moyenne d'une métrique de la machine.

Ci-dessous une capture d'écran de l'IHM



C'est un outil qui permet notamment de donner une échelle de risques de 1 à 5 sur le comportement de la machine, et permet notamment de donner une tendance de fond sur la consommation de la machine.

Cet outil est très rarement utilisé par les équipes d'expert système car l'échantillonnage et la pertinence des métriques ne sont pas suffisantes.

Tableau des avantages / inconvénients de OMNIVISION.

AVANTAGES	INCONVENIENTS
- Accessibilité à tous (pas d'authentification) - Très intuitif - Déployé sur les machines les plus critiques - Centralisation des données - Classification des serveurs/ application - Informations succinctes sur la configuration - Indicateur de tendance - Forte rétention des données	- Données propriétaires - Export de données impossible - Faible précision, pas de précision inadéquat pour de la performance - Agrégation des données trop importante - certaines métriques de risques sont absconses dans le mode de calcul notamment les échelles de risque.

✓ SYSLOAD

Cet outil de monitoring fut dans certains cas d'audits assez utilisé; il peut répondre à des sollicitations sur des audits aussi bien sur Unix que sur les environnements Microsoft Windows, l'échantillonnage est toutes les 15 secondes.

Cet outil est soumis à licence donc d'un nombre limité jetons à attribuer sur les machines concernées. Il faut pour obtenir les données que l'infogérant installe/désinstalle régulièrement les agents

AVANTAGES	INCONVENIENTS
- Déjà qualifié - Support éditeur (version mise à jour) - Logs circulaires - Portables sur de nombreux systèmes	- licence en nombre limité car payant. - Jeton qui implique une dés/installation - Logs non centralisées

Conclusion

Ces trois progiciels sont surtout orientés pour deux types de populations proche de la production et l'exploitation des machines, et leur finalité n'est pas de fournir du monitoring/collecte pour de la performance, d'où leurs lacunes intrinsèques sur cette thématique pour les deux premiers.

Les sondes de performance utilisées (certaines développées en internes) ne peuvent répondre à tous les besoins, notamment en termes de monitoring en temps réel, d'automatisation pour obtenir facilement les données, de la pertinence des données dans certaines infrastructures.

3.5.3 Propositions d'améliorations pour la collecte des données de performance dans le cadre d'une MCOI 102

Des solutions de développement en interne pour le redéveloppement de divers modules dans le traitement des audits de performance avaient été évoquées afin d'améliorer notre outil d'analyse et concernaient trois points principaux :

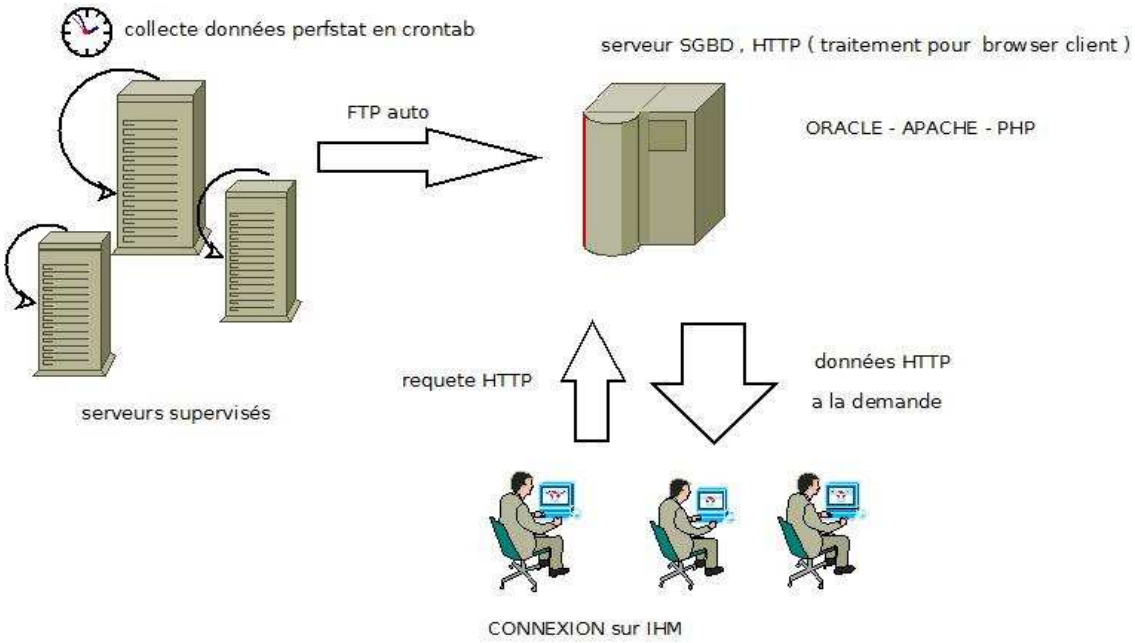
- Pour la collecte
- Pour le système d'affichage
- Pour l'archivage et la centralisation des données

La solution qui avait été étudié reposait sur le schéma suivant: le module de collecte et d'archivage sur les machines restaient sous la forme d'un script Shell générant un fichier par jour, le transfert s'opérant entre les machines clients et une machine de base de données part FTP avec un script injectant en base les données collectés à partir des archives.

Cette base interrogeable par requête SQL pouvait mettre en forme dans une IHM développé en PHP des graphiques de performance.

[102] Maintenance en conditions opérationelles des infrastructures

On aurait pu aboutir à ce type d'architecture.



Un certain nombre de considérations étaient à prendre en compte dans ce type d'architecture notamment en terme de goulot d'étranglement sur les transferts de fichiers, la volumétrie à définir et les métriques à sélectionner, d'évolution de manière modulaire l'IHM afin de prendre en compte les nouvelles souches et les nouveautés liés aux dernières implémentations.

In fine EDF s'est plutôt tourné vers une solution clé en main avec une architecture de fonctionnement propre au progiciel et les mises à jour opérées par un éditeur. Cette solution est mise en place depuis mi-2010 sans pour autant ne rencontrer quotidiennement des problèmes à sa mise en place et son déploiement progressif sur les serveurs à auditer. Ce progiciel est en fait une amélioration d'un progiciel déjà acheté et utilisé par EDF : OMNIVISION Investigation.

3.5.4 OMNIVISION INVESTIGATION :

✓ Généralités

A l'instar d'OMNIVISION, il est développé par SYSTAR société française spécialisée dans la fourniture de solution de gestion de performance. EDF a été choisi comme partenaire privilégié pour mettre en place ce nouveau progiciel qui dans quelques mois devrait être l'outil de prédilection pour des audits de performance et des suivis de performance en temps réel. Ce module Investigation d'Omnivision est davantage dédié aux ingénieurs systèmes, personnes en charge des audits, et service de support qu'à des Capacity manager, architectes, ou application Manager.

Le module Investigation est intégré à la technologie Omnivision et permet de faire un suivi en temps réel et sur une période écoulée ; il est par ailleurs non intrusif ; ainsi on peut :

- identifier les causes et conséquences d'un comportement suspect
- identifier les points de contention
- réaliser les rapports d'audits
- faciliter la collaboration et l'échange d'information.

L'architecture OMVI acronyme que nous utiliserons pour designer OMNIVISION Investigation permet la collecte de données de performances sur des systèmes hétérogènes (Windows et Unix/Linux) et en environnement virtualisé : sur 500 métriques, la granularité est de 20 secondes (présent et passé). Le grain peut être plus large lorsque la vue désirée est plus ancienne, ce qui permet un gain de place en termes de stockage lors qu'il semble moins probable d'intervenir ou d'expertiser des périodes éloignées.

Pour EDF le grain retenu est de 20 secondes pour une période moins de 5 Jours, de 1 min entre 5 jours et 15 jours et de 24 min au-delà des 15 jours.

Les données Omvi restent collectées localement sur les serveurs observés et sont rapatriées sur le serveur principal OMVI pour être visualisées à la demande. Aucune opération de transfert de fichier est nécessaire manuellement. On est donc un cas client-serveur avec un stockage repartie de la donnée sur les différentes machines supervisées.

OMVI permet de créer des modèles de tableaux de bord (cf. Dashboard) en fonction des profils accédant à l'IHM. Il est donc vital de sélectionner les Dashboards les plus intelligibles en fonction du public visé.

Ces Dashboards constituent donc l'élément principal de présentation de la performance de la machine, à partir de 500 métriques disponibles. Ils doivent être donc bien définis.

Certaines de ces métriques sont dites natives donc obtenues directement depuis le système d'exploitation, d'autres sont dites intelligentes, c'est à dire fabriquées par OMVI à partir d'algorithmes comme les échelles de risques. Il est par ailleurs assez délicat de connaître le calcul de certaines de ces métriques artificielles. Il est à noter que certaines fonctionnalités intéressantes telles que la corrélation inter-métriques permet également de faire apparaître plus facilement des liens de causalités ou les corrélations entre les éléments. Attention pour autant de ne pas confondre causalité et corrélation, établir des liens de fausses causalités est très dangereux.

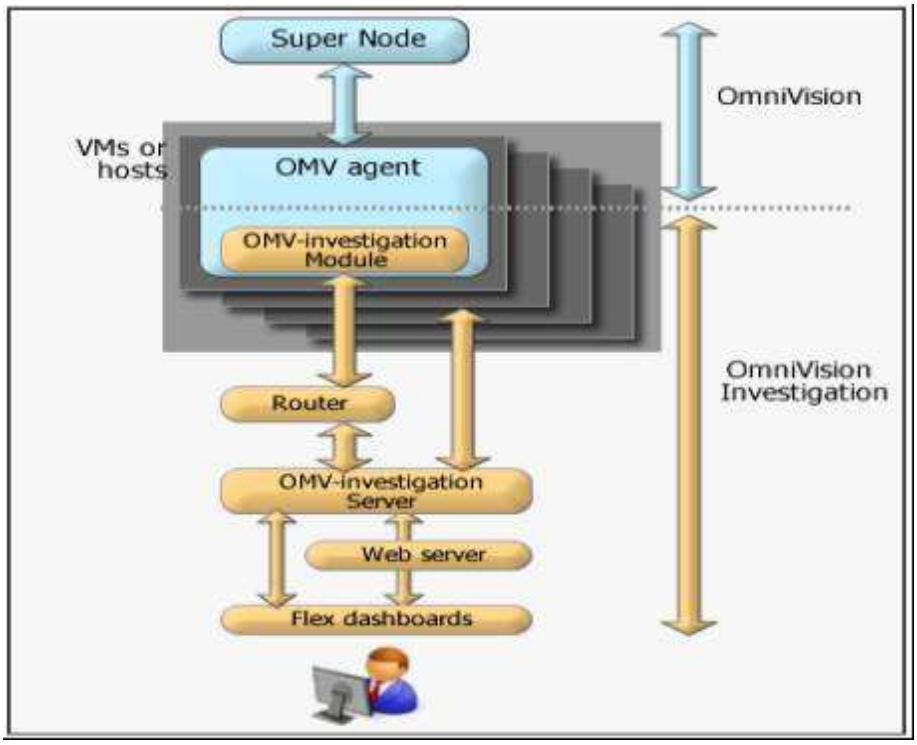
Etant donné que l'IHM est accessible à tous via une URL intranet (sous réserve de l'existence d'un compte nominatif propre à la personne), il est plus aisé de faire connaître les métriques à ces personnes et donc de s'abstenir d'envoyer à chacun des graphiques issus de macros Excel.

En cas d'expertise faite sur une situation de "crise", il est assez aisé de commenter les graphiques aux différents interlocuteurs présents.

✓ <u>ASPECT TECHNIQUES de la solution OMVI</u>

ARCHITECTURE GLOBALE DE OMVI et OMV

On remarque l'aspect Modulaire de OMVI (qui est une surcouche dédiée de l'architecture OMV)



Architecture hiérarchiques entre les nœuds.

Les collecteurs sous UNIX ont la forme d'un script SHELL « OMVKIT »

Un certain nombre de démons avec des fonctionnalités identifiés se chargent des tâches distinctes dans le process OMVI. Les communications se font par des sockets ouvertes en TCP.

Les atouts principaux :

- Analyse en temps réel et à posteriori
- Tableaux de bords paramétrables en fonction des profils utilisateurs
- Tableaux de bords existants pouvant être personnalisés
- Possibilité d'exporter les données (fichier .csv) ou des graphiques en .gif
- Analyse de bout en bout des infrastructures complexes virtualisées.
- Niveau de consolidation des données entre 20 secondes et 24 minutes.

- Echelle de risques possible, utile lorsque le public est néophyte à la problématique de performance.
- Métriques se déclinant sur chaque type d'OS et classées par type de manière intuitive.
- Rétention des données paramétrables, les données sont réécrites.
- Possibilité d'avoir les processus les plus consommateurs sur une période déterminée.
- Grande richesse au niveau des métriques.

Cf. : Voir en Annexe pour des captures d'écran OMVI

✓ Cependant un outil toujours en phase de maturation …

A noter que l'outil est en cours d'homologation chez EDF, c'est à dire que des points d'améliorations sont demandés à l'éditeur afin de rendre l'outil plus adapté aux besoins d'EDF ou faire part d'oubli sur l'outil.

A l'heure actuelle on peut reprocher à l'outil (fin 2011):

- De ne pas rendre possible la sélection spécifique d'une période donnée c'est à dire avec une date de début et de fin exacte.
- Que l'échantillonnage agrégé à 20 secondes des données soit parfois encore trop important, des micros événements ont une durée de vie de quelques secondes parfois comme un lien fibre défectueux rendant un disque indisponible quelques secondes. Et donc in fine suivre dans une période avec un grain plus fin les activités de la machine n'est pas un luxe.
- L'oubli de certaines métriques telles que l'activité par processeurs, rendant compte des problèmes de l'équilibrage de charge.

3.6 Améliorations fonctionnelles et organisationnelles des processus

3.6.1 Des points améliorants l'efficacité de ASC

✓ SIPI services

Un certain nombre d'améliorations ont été effectuées assez récemment afin de rendre la fonction ASC plus performante.

La première concerne la mise en place de SIPI services, c'est un premier niveau de prise en compte des demandes de support et d'audit. Cette cellule composée de 3 personnes a pour mission d'aiguiller le demandeur vers le bon interlocuteur et d'apporter des réponses aux questions les plus simples, et de constituer une interface entre les demandeurs et l'équipe ASC

dans le cas de demandes de support système. Ainsi cette prise en charge permet de mieux qualifier la nature et le niveau de l'incident et d'éviter de traiter des demandes ou le niveau de technicité est faible.

✓ Les audits payants

Jusqu'à une période très récente les audits étaient fournis gratuitement aux commanditaires dorénavant ceux-ci sont payants.

Non seulement ce flux financier fait apparaître des inputs/outputs, afin de mesurer la « rentabilité » d'une activité en interne et le coût/gain, mais c'est aussi un moyen de se conformer aux pratiques ITIL: faire apparaître les flux financiers.

Cela a deux effets sur la qualité du travail de l'équipe :

- Grâce à une meilleure formalisation des demandes, celles-ci sont plus précises et forcent les commanditaires à être plus descriptifs et précis.
- Les demandes d'audits sont mieux justifiées, les demandes à mauvais escient sont moindres, le phénomène de la gratuité poussait certains commanditaires à exagérer leurs besoins. L'équipe ASC est donc sollicitée sur la partie audits de manière plus pertinente. L'effet pernicieux est que le coût des audits constituent une barrière d'entrée trop importante pour certains projets puisque que le prix d'un audit est imputé aux ressources financières des projets et donc ostracise les projets les moins « riches ». Depuis cette mise en place, nous observons une diminution nette des demandes superflues. J'ai pour ma part chiffré à une baisse de 50 % des demandes, auparavant nous avions tout de même 50-60% de demandes d'audits injustifiées.

✓ OMVI

L'apport d'OMVI est aussi intéressant. D'une part il permet pour l'étude de performances sous Solaris et Windows l'obtention de graphiques et métriques de manière plus aisée. On comble véritablement une des lacunes de ces systèmes en termes de métrologie et de mise en forme de ces métriques même si certaines sont sujettes à caution. Mais cela nous permet également d'être plus réactif lors de conférence de crise et de pouvoir plus facilement commenter ce qui se passe sur la machine en temps réel. Parallèlement elle permet à certains intervenants d'avoir la possibilité de visualiser facilement les performances de la machine par simple connexion au serveur OMVI.

✓ Reporting du stockage en baie

On peut noter également que l'IHM pour la performance générique des Baies de stockage et des formations données par les équipes a eu pour incidence d'améliorer la qualité et la rapidité (NB : Lorsque le reporting du stockage n'était pas en place par l'IHM nous pouvions attendre 2-3 semaines avant d'avoir des analyses de la part des ingénieurs stockage) de nos analyses et d'être moins dans une position expectative et attentiste lors de supputations de dégradation de performance sur les disques de baie. Cette interface est déjà une bonne réponse à nos besoins en tant que auditeur système même si cette interface est encore perfectible dans le détail des éléments et dans les effets de corrélation entre les machines partageant la même baie.

Si un certain nombre de points ont été bénéfiques dans le travail de l'auditeur système, il reste néanmoins des points à encore améliorer.

3.6.2 Quelques points à améliorer

En tant qu'auditeur largement impliqué dans les problématiques de support et de résolutions d'incident, il y aurait un certain nombre de points à améliorer à des fins d'efficience.

Dans un environnement où les machines sont maintenant très fortement virtualisées, connectées à des baie de stockage de disques, partageant également des périphériques réseaux.

✓ Le Manque d'informations et d'accès aux infrastrcutures

L'équipe ASC n'a pas des moyens suffisants mis à sa disposition ; sans passer par les autres équipes pour diagnostiquer ou avoir des informations rapidement, sur les Virtual I/O server par exemple. Si les informations sont maintenant plus facilement accessibles de la part du stockage, nous n'avons pas à ce jour véritablement de mesure de performance du réseau.

Aussi il faudrait donc étendre le périmètre de l'équipe ASC aux performances des « châssis » (vue verticale) , des VIOS ou ESX/VSPHERE qui permettent de faire la virtualisation afin de répondre plus efficacement aux demandes d'audits de performance. Nous sommes encore trop dépendants d'équipes externes, qui ne sont pas forcément expertes dans les problématiques de performance. Plus que jamais dans ce type d'environnement la performance se diagnostique à plusieurs niveaux. Cela nécessite donc des accès et privilèges étendus, une meilleure formation et une plus grande communication entre les deux équipes.

✓ Communication à améliorer avec l'ingénierie système

Dans la même optique il semble judicieux d'être davantage intégré dans les équipes d'ingénierie serveurs, ce sont en effet les équipes qui produisent les socles et les solutions techniques et sont directement en relation avec les constructeurs. Le constructeur peut faire plus facilement part de problème de performances sur certaines implémentations. Sur la partie résolution d'incidents c'est d'ailleurs encore plus le cas.

D'autre part les équipes d'ingénierie serveur possèdent plus d'experts pointus et dédiés à un domaine particulier. Les ingénieurs d'ASC ont une compétence système plus généraliste sur les systèmes Unix. Pouvoir mieux intégrer les équipes en terme aussi de localisation géographique permettrait des créer des externalités positives pour les deux entités et augmenter l'émulation intellectuelle entre les différents experts. Le système actuel ne maximise pas les compétences de chacun et ne les confine que sur un périmètre restreint.

L'équipe d'ingénierie a notamment plus de facilité à tester les solutions sur du matériel spécifique, nous pourrions donc bénéficier de cet apport directement.

✓ Communication à améliorer avec les architectes

Pareillement il semble avantageux de pouvoir bénéficier plus fréquemment des formations et des informations émanant des architectes, ce qui n'est pas assez le cas. Néanmoins l'équipe ASC se doit de connaître les nouvelles implémentations et les gains susceptibles d'être obtenus avec les nouvelles architectures des dernières machines.

✓ Retour d'expèrience des commanditaires à mettre en place et suivi

Lors de la mise en place des préconisations fournies à l'issue des audits, il se trouve que bon nombre de préconisations ne sont pas appliquées rapidement ou soit ne le sont pas du tout pour des raisons parfois inexpliquées, certaines peuvent toutes fois être dues à ce que :

- Les modifications ne peuvent être appliquées que lors de redémarrage des machines ou arrêt des applications ainsi l'infogérant daigne à faire les modifications rapidement.
- Certaines modifications sont lourdes à mettre en place notamment en termes de revue de la configuration.
- Les commanditaires cherchent non pas de savoir comment corriger ou améliorer la performance, mais si véritablement la machine est plus peu ou pas performante par un constat de l'auditeur

Le fait que l'auditeur système ne sache pas si les préconisations ont été portées ou quand elles auront/eues lieues ne permettent pas à ce dernier de véritablement juger des effets. Certains commanditaires font parfois un retour, notamment cela est gratifiant si les résultats ont été à la hauteur des espérances ou si les effets ont été peu probants de corriger notre jugement.

Ce manque de retour d'informations ne permet donc pas au processus de la performance d'être réentrant et alimenter de nouveau la base de connaissance de l'auditeur ou de la cellule. De plus cela créé parfois un sentiment de travail fait « inutilement ».

Un meilleur retour des préconisations formulées en terme de résultats serait aussi envisageable par le biais d'un formulaire une forme de retour sur résultat, afin d'enrichir la base de connaissance et permettrait de savoir où on en est terme de suivi.

4 Bilan et Conclusion

4.1Bilan de la mission

Au cours de ces cinq années effectuées à EDF, j'ai pu apporter de nombreux éléments et être contributeur sur de nombreux sujets sur mon activité "Audit de Performance", on peut les résumer par :

- un apport d'un document inédit au sein de EDF (une version complémentaire plus étendue et technique a été rendue aux personnes d'EDF); le thème de la performance couvrant trois types de système d'exploitation n'est pas simple à synthétiser sur un mémoire d'ingénieur. Mais il apparait surtout essentiel pour aborder cette problématique de bien comprendre les bases de fonctionnement des éléments visés par l'optimisation système.

- Des notes complémentaires techniques sur la performance : en parallèle de ce document, j'ai rédigé de nombreuses notes visant à expliquer certaines parties optimisables telle que le gestionnaire de volume VxVM, Solstice Disque Manager ou même encore le tuning de la couche TCP/UDP pour Solaris, Linux etc .... Ces notes ont été mis à la connaissance des autres auditeurs. Ces notes de plusieurs de pages m'ont par ailleurs aidé à rédiger ce mémoire.

- Aide à l'élaboration des Dashboards pour Omnivision Investigation (sélection des métriques les plus pertinentes avec le responsable du projet)

- Participation à la base de connaissance (Référentiel de Développement et d'Exploitation) d'EDF, notamment pour la partie tuning.

- Démocratisation de l'utilisation de NMON pour le Linux Redhat Entreprise qui n'était pas utilisé pour l'élaboration des audits de performance il y a 5 ans. Il se révèle plus complet que l'outil qui était jusqu'à présent utilisé.

- Etude et Réalisation de plus de 150 audits tout Unix confondus dont 30% ont pu résoudre un véritable problème de performance. (50 % des audits étant des audits demandés sans problème de performance clairement identifiés)

- Participation et implication à des campagnes de test majeures, en support des équipes projets des applications.

En plus de ces apports ; pour l'entreprise :

-Un gain économique.

En complément pour l'entreprise ces audits de performance ont pu faire économiser à EDF parfois l'achat supplémentaire de Mémoire ou "de CPU". Concernant les parties disques et réseau, l'apport économique des audits de performance est difficilement quantifiable car l'apport d'un disque dans un système résoud plus généralement un problème de volumétrie. Le tuning sur la baie de stockage n'est pas du ressort de la cellule ASC. Les cartes réseaux sont quant à elles assez rarement impliquées dans les problèmes de performances, les problèmes réseaux sont généralement d'ordre capacitif pour la partie serveur.

- Amélioration du fonctionnement des applications, des équipes et utilisateurs plus satisfaits (ce qui est infine un des buts de l'étude de la performance)

- Un accompagnement permanent et une meilleure sensibilisation des équipes projets à la problématique de la performance.

Je dois ajouter mon implication sur les problématiques de support aux administrateurs système sur des problèmes de fonctionnement des systèmes d'exploitation, ou lors de la mise en place de logiciels ... Cette activité étant qualifié support de niveau 2-3. Cette dernière représentait 50% de mon temps d'activité à EDF.

Plus personnellement cette spécialisation dans la performance des serveurs a été une découverte pour moi, puisque lorsque je suis arrivé sur cette mission EDF, je possédais uniquement de vagues notions de la performance sur les machines Unix, ces notions étaient héritées de mon activité d'administrateur système mais n'étaient comprises et abordées que de manière superficielle. Grâce à cette expérience j'ai pu approfondir mon expertise système de manière générale que je peux mettre dorénavant à profit.

Il faut aussi bien comprendre que la problématique performance dans les cursus de formation IBM, SUN ou autre constructeur est une activité d'un ingénieur systèmes à part entière et abordée lorsque l'administration d'un système d'exploitation est déjà bien assimilée.

4.2 Conclusion

L'optimisation des serveurs est une tâche pour l'administrateur et ingénieur système bien particulière dans le cadre de son activité quotidienne, dans certaines entreprises surtout dans les plus importantes, il est souvent à la charge d'une équipe dédiée, aussi les impacts sont souvent importants.

Le tuning nécessite une veille technologique permanente, la documentation à ce sujet n'est pas toujours facile à trouver notamment pour les systèmes libres par ailleurs très versatiles au fil des versions. Ce travail est aussi de plus en plus le fait de plusieurs équipes inter agissantes sur le système, les équipes gérant les baies de stockage ou même le réseau pour ne citer qu'elles ont aussi leurs implications.

Il est fort à parier que ce mémoire est d'ailleurs en partie d'un point de vue technique déjà obsolète du fait qu'il fait référence pour une partie à des systèmes d'exploitation plus supportés par les éditeurs et tant les évolutions technologiques sont nombreuses et viennent complexifiées le travail de l'ingénieur. Le but in fine de nombreuses de ces améliorations est de rendre la machine de plus en plus indépendante de certaines contraintes matérielles et "intelligente" si on peut le dire.

Il est d'ailleurs à noter que le nombre de serveurs logiques ou machines virtuelles au sein d'un système d'information tend souvent à croitre alors que le nombre de personnes opérationnelles en charge de ces serveurs diminuent, il est donc vital que ces serveurs s'affranchissent de plus en plus de l'intervention humaine au quotidien et puissent grâce notamment à la virtualisation et de ses sous-jacents rendre ces problèmes plus transparents pour leur exploitation.

Ces implémentations nouvelles nécessitent plus que jamais aussi de bonnes connaissances techniques du personnel ayant à faire aux problèmes de performances, une bonne connaissance également des autres briques du système d'information comme le système de stockage et le réseau afin de comprendre le discours de chacun et des impacts potentiels.

De nombreux outils dédiés à la performance permettent de faciliter le travail de l'ingénieur système mais il est à sa charge de savoir bien interpréter ce qu'il observe ou collecte.

Même si la montée en puissance de la virtualisation au sens large et l'émergence du cloud computing tentent d'éloigner ou de gommer les limites et les problématiques inhérentes des serveurs physiques, celles liées à la recherche de la performance restent très actuelles et toujours de plus en plus complexes à traiter et s'avère de plus en plus impactant pour les acteurs industriels ayant optés vers ces technologies notamment si certains d'entre eux hébergent l'activité de plusieurs clients sur les mêmes plateformes matérielles.

REFERENCES :

Livres Tout UNIX

Aeleen Frisch - Essential System Administration, 3eme Edition (1077 p)- Ed OReilly

AIX-IBM:

LIVRES IBM/AIX:

Kumiko Hayashi - Kangkook Ji :AIX 5L Practical Performance Tools and Tuning Guide (744 p) (2005) - IBM Ed

Bruno Digiovanni - Oliver Stadler - Federico Vagnini :POWERVM Virtualization Active Memory Sharing (redp4470) (92 p) – IBM Ed

TECHNOTES :

Vijay Adik : AIX Performance Configuration & Tuning for Oracle – ATS Oracle Solution Team (56 p) (2009)

David Hepkin : Overview of AIX page replacement (15 p) – IBM (2008)

Sujatha Kashyap - Bret Olszewski - Richard Hendrickson :Improving Database Performance With AIX Concurrent I/O – (13 p) (2003) –IBM

Susan Schreitmueller - Turning The Knobs: Practical AIX Tuning 770134 - Susan Schreitmueller (52 p) (2005)

URL :

http://www.ibm.com/developerworks/views/AIX/libraryview.jsp?search_by=Optimizing+AIX+5L+performance

http://www.ibm.com/developerworks/AIX/library/au-vmm/index.html

http://docs.oracle.com/html/A97297_01/appa_AIX.htm

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.AIX.prftungd/doc/prftungd/AIX_mem_aff_support.htm

http://kalwin.fr/Unix/add_hoc/techdocs/91401108611004.html

SOLARIS:

LIVRES :

Solaris™ System Performance Management SA 400 - Student Guide – (602 p) SUN

Richard McDougall - Solaris 10 and OpenSolaris Kernel Architecture – SUN (2006)

Adrian Cockroft - Sun Performance and Tuning – SUN (1998)

Richard Mc Dougall and Jim Mauro - Solaris Internals -SUN (2006)

TECHNICAL NOTES:

Mohan Bhyravabhotla - Bob Rader - Vern Wagman : VERITAS Quick I/O - ORACLE (40 p) (1998)

Bob Larson - Performance Oriented System Administration (Blue Print) (22p) (2002)

Tales from the Trenches: The Case of the RAM Starved Cluster (Blue Print) Richard Elling (10 p) (2000)

Solaris Tunable Parameters Reference Manual – SUN (200 p) (2006)

John Brady - A Strategy for Managing Performance - (Blue Print) (13 p) (2002)

Deepak Kakadia - Understanding Tuning TCP - (Blue Print) (30 p) (2004)

Richard McDougall - Understanding Memory Allocation and File System Caching in OpenSolaris

URL:

http://docs.oracle.com/cd/A95434_01/a86676/concpts.htm

http://www.princeton.edu/~Unix/Solaris/troubleshoot/

http://www.dba-oracle.com/t_tuning_cpu_usage_vmstat.htm

http://docs.oracle.com/cd/B19306_01/server.102/b14211/ch23_os.htm

LINUX

LIVRES :

Sandra K.Johnson - Performance Tuning for Linux Servers (2009)

Eduardo Ciliendo - Takechika Kunimasa Linux Performance and Tuning Guidelines - IBM REDBOOK

Technical Notes :

Oskar Andreasson - Ipsysctl tutorial 1.0.4

Rik van Riel - Page replacement in Linux 2.4 memory management

Abhishek Nayani Mel Gorman & Rodrigo S. de Castro - Memory Management in Linux

Darren Hoch - Linux Performance Monitoring - StrongMail Systems, Inc

Performance Best Practices for VMware 4.0 - VMWARE

Larry Woodman et John Shakshober - Red Hat Enterprise Linux Performance and Tuning

Tuning and Optimizing RHEL for Oracle 9i and 10g Databases - REDHAT EDITION

Nick Carr - Linux Kernel 2.6 Features in Red Hat Enterprise Linux – REDHAT EDITION

Mladen Gogala - Tuning Linux VM on Kernel 2.6

URL:

http://www.kernel.org/doc/Documentation/sysctl/vm.txt

http://datatag.web.cern.ch/datatag/howto/tcp.html

http://osr507doc.sco.com/en/PERFORM/CONTENTS.html

http://www.performancewiki.com/Linux-tuning.html

http://www.speedguide.net/articles/Linux-tweaking-121

http://www.westnet.com/~gsmith/content/Linux-pdflush.htm

http://www.puschitz.com/TuningLinuxForOracle.shtml#32BitArchitecture

Résumé :

Pour une entreprise de taille mondiale, le traitement des problématiques de performance de ses serveurs est important, en effet celui-ci lui permet non seulement de fournir un service qui doit répondre aux attentes des utilisateurs finaux ou clients, d'éviter l'indisponibilité de ses applications et serveurs et de disposer au mieux des ressources matérielles. Elle met en général en place une équipe dédiée, chaque spécialiste a à sa charge un périmètre de machine relatives à ses compétences. Ce document fait particulièrement référence aux optimisations possibles et mises en œuvre sur les systèmes d'exploitation Solaris, AIX et Redhat Enterprise.

Ces optimisations de serveurs portent sur quatre types de composants abordés consécutivement à savoir sur les processeurs, la mémoire, les disques et le réseau correspondant au cheminement méthodologique employé pour les audits de performance. Ces éléments doivent être donc supervisés par le biais de métriques et de commandes, ces données résultantes sont étudiées, analysées ce qui donne lieu à des préconisations.

Cependant dans un système ou un certain nombre de ressources sont virtualisées et/ou partagées entre plusieurs machines, un certain nombre de point sont à reconsidérer et viennent modifier l'approche habituellement admise dans l'optimisation des serveurs. Pour aider le spécialiste un certain nombre d'outils sont mis en place, ils y sont ici rapidement évaluer.

Mots clés : Unix, Linux, Optimisation, performance, AIX , Solaris ,Redhat

Abstract

For a global company, the performance problematic is important, thus it allows a service according to end users or customers' expectations, avoids unavaibility of her services and servers and finally optimizes the use of resources. In general this company has a core-team dedicated to this task. Each specialist is in charge of an array servers regarding to his skills. This document refers specially to possible tunables parameters and these who take place in operating system like Solaris, AIX and Redhat Enterprise.

These optimizations are related with four different type of components reviewed sequentially: processor, memory, disk system and network according to the methodology path used in the performance audit. These components have to be monitored through specific metrics and commands, the output data must be analyzed and studied so that we can propose hints and tweaks.

Nevertheless for a machine; some resources are virtualized or shared through different servers, some statements have to be reconsidered and thus modify our usual admitted approach in tuning. In order to help the specialist, some tools are used, here they are briefly evaluated also.

Keywords: Unix , Linux , Tuning , Tweaking , performance , AIX , Solaris , Redhat