

- Accueil
- A propos
- Nuage de Tags
- Contribuer
- Who's who

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source

30 jan 2009

Problématique de consolidation et atteinte des objectifs de niveau de service (SLO) avec Xen

Catégorie : <u>Administration système</u> Tags : <u>GLMF</u>



Retrouvez cet article dans: Linux Magazine 92

Ou comment gérer les priorités applicatives entre plusieurs environnements partagés au sein d'un même serveur.

1. Défis de la [para]virtualisation dans des contextes de consolidation

De nombreuses entreprises s'interrogent sur l'optimisation de leur infrastructure et de leurs investissements informatiques. En effet, les restrictions budgétaires imposent une limitation du nombre de serveurs, ainsi qu'une remise en cause de la logique dite « des silos » où un serveur héberge une application. Historiquement, les serveurs étaient configurés en fonction

des pics de charges applicatives. Ce modèle architectural entraîne généralement une sous-utilisation des capacités de traitement, mais aussi parfois une saturation de celles-ci. Dans la figure 1, nous proposons un exemple de taux d'utilisation CPU moyen par serveur (extrait d'un cas réel). On constate qu'une grande majorité de ces machines dispose de capacités de traitement dont l'entreprise en question ne profite pas. Une étude récente a d'ailleurs montré que le taux moyen d'utilisation d'un serveur « Unix » était de 30% alors que pour « Windows » ce taux tombait à 15%.

Utilisation moyenne des serveurs

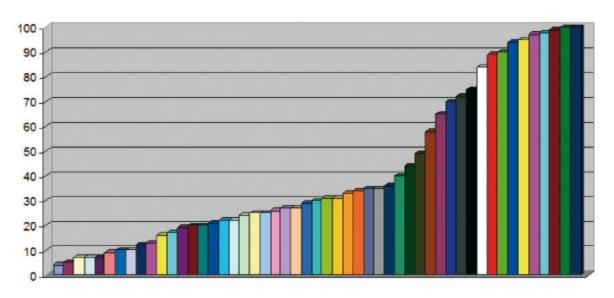


Figure 1 : Utilisation moyenne des CPU dans une entreprise pilote

Alors, que faire de ces capacités de traitement inutilisées? La réponse à cette question se trouve en partie dans les projets de consolidation qui, pour nombre d'entre eux, reposent sur des technologies de virtualisation.

Ces technologies peuvent être des suites complètes de type adaptive infrastructure basées sur des composants matériels et logiciels comme HP Virtual Server Environment ou IBM Virtualization Engine.

Elles peuvent aussi être complètement virtualisées au niveau logiciel en offrant, dans ce cas, des fonctionnalités moindres. La référence dans ce domaine étant ESX de VMWare. Mais, il existe d'autres solutions comme l'Integrity Virtual Machines de HP ou encore Xen qui servira de support à cet

article.

Très concrètement, ces solutions de virtualisation ou de paravirtualisation permettent de faire fonctionner plusieurs serveurs logiques sur une même machine physique (host). Les deux principaux défis de telles solutions sont, d'une part, l'isolation et la sécurité des machines virtuelles les unes par rapport aux autres. L'octroi de ressources et la gestion des priorités entre les différentes machines virtuelles (VM), d'autre part. Ce point est d'autant plus critique qu'il conditionne l'atteinte des objectifs des niveaux de service (SLO pour Services Level Objectives). C'est ce deuxième point que nous allons détailler dans cet article au travers d'un exemple de gestion de charge et d'allocation de ressources. Mais commençons par un rappel de ce qu'est Xen et de ses principes de fonctionnement.

2. Xen: quelques rappels

Xen [1][2] est un Moniteur de Machines Virtuelles (VMM) dont certains aspects ont été présentés dans les Gnu/Linux magazine 85, 87 et 89. Xen est une technologie dite «de paravirtualisation» dans le sens ou une machine virtuelle de type Xen nécessite une modification de son noyau. On utilise parfois dans ce cas le néologisme «Xenification» pour parler d'un OS modifié de la sorte.

Une machine virtuelle Xen est appelée «Domaine». La première d'entre elles est le Domain-0 en référence au numéro d'identification affecté à chaque domaine actif. Ce Domain-0 est celui qui est créé lors du boot du serveur. Aux yeux d'un utilisateur, ce domaine initial est similaire à une machine standard. Il est cependant différent dans la mesure où il a la charge de la gestion et de la communication avec les autres domaines instanciés sur le serveur. Les domaines utilisateurs, généralement appelés «domaineU», se caractérisent principalement par un fichier de configuration fournissant de nombreuses informations comme le nom, la mémoire allouée, le nombre de CPU virtuelles (appelées par la suite «VCPU» dans cet article), un identificateur universel unique (UUID), le pont réseau, l'amorce de démarrage, ainsi que de nombreuses autres informations optionnelles. Une machine virtuelle Xen se compose également d'un espace de stockage (qui peut être un fichier, un disque, une partition physique ou encore un volume logique).

3. Problématique de la gestion de la charge et des SLO

Lorsque plusieurs applications cohabitent sur un même système, se pose la question de l'allocation des ressources, du partage des capacités de traitement

ainsi que de la gestion des priorités. Dans un environnement multidomaine Xen, l'accès aux capacités de traitement repose sur un ordonnanceur (scheduler) qui séquence et équilibre la charge générée par les VCPU des systèmes invités entre les CPU disponibles du système hébergeur multiprocesseur. Distinguons ici les ordonnanceurs de type préemptif qui assignent un client à une CPU en fonction de différents types d'événements (une VCPU est bloquée, cède sa place, a consommé son temps ou est réveillée) et les non préemptifs qui n'exécuteront une décision d'ordonnancement que lors de la libération volontaire d'une CPU par un client actif. Le mode préemptif est particulièrement adapté dans des contextes de fortes charges (e/s et CPU) ainsi que dans des environnements multiprocesseurs. Xen est singulier dans le monde de la virtualisation dans le sens où il permet de choisir entre 3 ordonnanceurs différents. Charge à l'administrateur système d'opter pour le mieux adapté à ses besoins. Le choix devra se faire entre :

- Borrowed Virtual Time (BVT). Basé sur le concept de temps virtuel, ce scheduler classe les VM en fonction de leur temps d'exécution supposé. Il exécute celle ayant le temps le plus court en premier. Il était utilisé avec Xen version 2.
- Simple Earliest Deadline First (SEDF). Le SEDF utilise un algorithme temps réel pour assurer du temps garanti. Il dédie une portion de temps global définissable à un domaine et permet de dire si oui ou non le domaine peut excéder le temps CPU qui lui a été dédié initialement (en cas de non saturation du système).
- Credit Scheduler. Il est le plus récent des schedulers. Il permet de gérer automatiquement l'équilibrage de charge sans intervention de l'administrateur. L'étude réalisée ici se base sur ce modèle. Nous allons donc expliquer en détail son fonctionnement.

Pour connaître le type de scheduler actif sur un système, l'option dmesg (comme diagnostic message, une commande OS qui imprime les messages du buffer de noyau) de la commande xm-devra être utilisée:

```
XEN# xm dmesg | grep -i sched
(Xen) Using scheduler: SMP Credit Scheduler (credit)
```

La sélection d'un scheduler se fera en modifiant le fichier grub.conf-et en ajoutant le paramètre sched=XXX. XXX-correspondant au mode d'ordonnancement retenu. Toutefois, même s'ils restent disponibles pour le moment, il est prévu de désactiver voire de supprimer BVT et SEDF du noyau Xen. Il est donc fortement conseillé d'utiliser le Credit Scheduler.

Du point de vue algorithmique, chaque CPU physique gère sa propre file d'attente de VCPU. Cette file d'attente est triée selon le niveau de priorité de chacune des VCPU. La priorité d'une VCPU peut être de deux types, over ou under, c'est-à-dire au-dessus ou au-dessous. Ce statut caractérise les VCPU qui ont ou n'ont pas dépassé leur temps d'accès (en anglais, on parle de fair share) aux ressources CPU sur la période courante. Très logiquement, lorsqu'une VCPU est ajoutée à une file d'attente, elle est placée après les autres VCPU de priorités similaires.

Quand une VCPU est active, elle consomme des crédits. La valeur atomique est appelée le tick, elle est de 10 ms. À intervalles réguliers – toutes les 30 ms – un thread système recalcule les crédits qu'une VM active a gagnés et les alloue. Une valeur négative indique une priorité de type over. Tant qu'une VCPU n'a pas consommé ses crédits, sa priorité est de type under.

Chaque CPU, à chaque décision d'ordonnancement, activera la VCPU ayant un statut under et se trouvant au sommet de la file d'attente. La décision de scheduling fait partie intégrante de l'activité de l'ordonnanceur. Il est par conséquent prévu pour être léger et efficace. Il est à noter que le Credit Scheduler a été écrit pour assurer une meilleure efficacité dans les gros systèmes symétriques multiprocesseurs (SMP) que BVT et SEDF qui ont de réels problèmes dans ce genre d'environnements.

Si une CPU ne trouve pas de VCPU ayant une priorité de type under dans sa file d'attente locale, elle ira chercher dans les autres files. Une CPU ne passera en statut inactif (idle), que si aucune autre VCPU n'a de tâche à exécuter (aucune VCPU en attente, quel que soit son statut). Ce système d'équilibrage de charge (load balancing) garantit que chaque VM disposera d'un temps d'accès aux ressources CPU équitable.

Il est cependant possible de désactiver cet équilibrage de charge en affectant une VCPU particulière à une CPU physique et ainsi restreindre l'usage de celle-ci en utilisant la fonction générique vcpu-pin. Notons que l'affectation d'une VCPU à une CPU n'est pas exclusive. C'est-à-dire que la CPU pourra traiter des tâches soumises par des VCPU d'autres domaines. Pratiquement, le temps total de traitement va être réparti entre les différentes VM. Dans l'exemple ci-dessous, nous affectons la VCPU 0 du domaine swingbenchym-à la CPU 3 de la machine hôte. Avant cela, la commande xm-list fournit la liste des domaines actifs sur le serveur. Ceci permet d'identifier les noms et identifiants des domaines. Ces informations sont nécessaires à l'utilisation de la fonction vcpu-pin.

Usage: xm vcpu-pin <Domain> <VCPU> <CPU> XEN# xm list ID Mem(MiB) VCPUS State Time(s) Name Domain-0 0 1817 1 r---- 24974.0 joevm1 9 1024 2 r---- 71304.3 joevm2 7 1024 4 -b---- 11381.7 512 1 -b---- 33227.4 swingbenchvm 8

XEN# xm vcpu-pin swingbenchvm 0 3

Les outils xentop (mode texte) ou virt-manager [3] (mode graphique) délivrent une image instantanée de l'activité globale du système hôte.

XEN# virt-manager

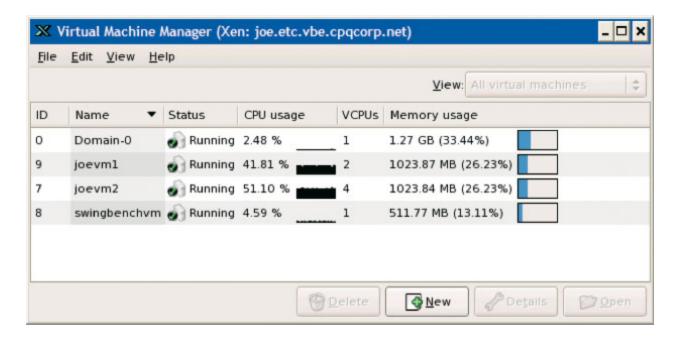


Figure 2 : Répartition du temps CPU entre les différentes VM – consommation instantanée plus historique

La gestion des charges et donc des priorités applicatives nécessaires à l'atteinte des SLO s'appuiera sur l'algorithme du Credit Scheduler. De plus, un administrateur pourra, manuellement, pondérer une VM. Il pourra aussi limiter le temps qu'une CPU consacre à un domaine (on parle alors de caping). Pour cela, la commande xm sched-credit, dont nous donnerons plusieurs exemples par la suite, sera utilisée.

Présentons maintenant ces fonctions de caping (ou activation de seuil supérieur) et de pondération.

4. Le caping de l'activité CPU

D'une manière générale, un ordonnanceur peut fonctionner selon les deux modes: work-conserving et non work-conserving (NWC). Dans le premier mode, une CPU est considérée comme libre si aucune tâche associée à une VCPU, quel que soit le domaine d'origine, n'est en attente sur le système

hébergeur.

A contrario, en mode «NWC», le temps CPU dédié à un domaine ne pourra excéder la part initialement allouée. On parlera alors de seuil supérieur. Le caping est donc l'option permettant de limiter le montant maximum de temps CPU qu'un système invité va être capable de consommer, même si le système hôte dispose de cycles CPU non utilisés. L'activation du caping permet ainsi de passer du mode work-conserving (mode par défaut de Xen avec le Credit Scheduler) au mode NWC.

Ce seuil supérieur est exprimé en pourcentage d'utilisation d'une CPU physique. 100 est donc 1 CPU physique, 50 est la moitié d'une, 400 est l'équivalent de 4 CPU, etc. La valeur par défaut, 0, signifie qu'aucun seuil n'a été défini pour une VM.

Dans l'exemple ci-dessous, le caping d'une VM disposant de 4 VCPU a été activé. Au temps 0, la VM consomme l'ensemble des ressources disponibles puisqu'il n'y a ni compétition, ni limite de définie. Dans le temps 1, nous limitons à 50% la consommation CPU maximum autorisée pour le domaine joevm2. Le seuil supérieur étant exprimé en pourcentage global, il aura une valeur de 200 (50%x4). On constate alors qu'approximativement 50% des ressources CPU sont inutilisées ou réservées aux autres domaines.

#	Domain id	Action	Tpm result
1	2, 4, 6	-c=0 -w=256 for all; startup swingbenh; hammer down	5400
2	2, 4, 6	Startup hammerora	4600
3	6 (swb db)	-c=50	1400
4	6 (swb db)	-c=25	550
5	6 (swb db)	-c=Ø	4200
6	2 (hammer)	-w=16384	220
7	2 (hammer)	-w=2048	1100
8	6 (swb db)	-w=2048	4700
9	2 (hammer)	-w=256	5300
10	6 (swb db)	-w=256	4400

Enfin, précisons que quel que soit le nombre de VCPU, à caping identique, deux VM consommeront très sensiblement la même charge CPU. Par exemple, une VM disposant de 4 VCPU limitées à 25% chacune (soit un total de 100) disposera des mêmes ressources que 2 VCPU limitées à 50% l'unité. Nos tests ont confirmé cette évidence.

5. Pondération des VM

La pondération se distingue de la logique de seuil dans la mesure où elle ne s'attache pas à limiter la consommation CPU d'une VM, mais plutôt à valoriser l'importance d'un environnement par rapport à un autre. De plus, la pondération n'est pas une option, puisque tout domaine Xen a un poids dès sa création.

Ce poids s'exprime au travers d'un entier qui aura une valeur comprise entre 0 et 65535. Plus le poids est élevé, plus le temps CPU consacré au domaine sera important. Un domaine avec un poids de 512 aura deux fois plus de temps CPU qu'un domaine avec un poids de 256. A condition bien entendu qu'il y ait de la contention sur le système. Il est conseillé de pondérer une VM avec une valeur multiple de 256 qui constitue la valeur par défaut. À chaque recalcul de crédit par le scheduler – toutes les 30 ms – ce poids sera pris en compte et aura pour effet de ralentir ou d'accélérer le passage d'une VCPU du statut under à celui de over et donc à privilégier une VM plutôt qu'une autre.

La ligne de commande suivante permet de modifier le poids d'un domaine :

XEN# xm sched-credit —d mon-domaine —w 512

-d, pour domain, est le paramètre pour l'identifiant du domaine et -w, pour weight, celui du poids affecté.

Maintenant que ces concepts sont définis, il est temps de les évaluer. C'est ce que nous nous proposons d'étudier dans la deuxième partie de cet article.

6. Création d'un environnement de test

Nous l'avons dit précédemment, la gestion de la charge nécessitera un arbitrage uniquement en cas de contention sur le serveur physique. Si l'usage des CPU n'est pas saturé, il n'y a nul besoin d'intervenir, puisque chaque domaine dispose des ressources dont il a besoin.

Dans ce contexte de consolidation avec Xen ou tout autre outil de virtualisation, il sera également judicieux de faire cohabiter une application critique soumise à SLO avec une ou plusieurs autres applications moins sensibles. Ceci permettra de définir aisément les priorités.

C'est dans cet esprit que nous avons créé un environnement de test capable de générer suffisamment de charge CPU pour démontrer les fonctionnalités de caping et de pondération. Cet environnement, se voulant le plus réaliste possible, comprend une application de type transactionnel et une autre de type décisionnel.

La figure 3 décrit les machines virtuelles créées pour l'occasion. En plus du domain-0, deux applications réparties entre trois domaines se partagent les ressources du host. D'un coté, se trouve hammerora [4], application de type décisionnelle consommatrice intensive de CPU et d'accès disque. De l'autre, l'environnement OLTP nommé swingbench [5]. Dans le détail, hammerora-est complètement intégré au sein du même domaine (joevm2-dans la figure 3) alors que swingbench-est réparti en deux domaines qui, très prosaïquement, représentent un serveur d'application et un serveur de base de données (joevm1). La communication se faisant par le pont réseau interne. Dans les deux cas, nous avons retenu Oracle 10g comme support de base de données. Il est à noter qu'Oracle supporte l'utilisation de sa base de données avec Xen. Cependant, si les conseils du support Oracle ne permettent pas de résoudre un problème identifié, celui-ci devrait être transféré au support Xen. S'il s'avère que le problème est bien lié à la base de données, il sera alors demandé au client de reproduire celui-ci dans un environnement natif (non paravirtualisé).

Du point de vue matériel, un serveur équipé de 4 CPU de type Intel Xeon avec 4GB de ram héberge ces domaines.

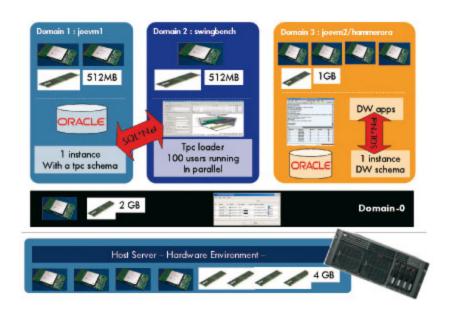


Figure 3 : Description de l'environnement de test

Avant de passer au contenu et aux résultats des tests, présentons succinctement les générateurs de charge swingbench-et hammerora.

Swingbench-est un générateur de charge de type freeware, c'est-à-dire sans licence ni coût de support. Il présente l'avantage de fonctionner avec des bases mono-instances et en grappes (connu sous le nom de RAC). Son interface graphique permet de visuellement mesurer la performance de l'environnement testé.

Le benchmark (banc de test) fourni se base sur le schéma oe qui est livré en standard comme modèle avec les bases de données. Il peut être utilisé sans interruption jusqu'à ce que l'espace disque dédié au schéma soit complètement saturé. Swingbench introduit une forte contention sur un petit nombre de tables.

L'activité générée se compose d'un mix de 5 types de transactions associant des ordres de lecture (select), de mise à jour (update) et d'ajout (insert) de données. Ceci garantissant un équilibre entre les ordres de lecture et d'écriture.

Le framework complet a été développé en Java et peut, par conséquent, être utilisé sur une grande variété de plateformes.

Hammerora, quant à lui, est un outil du monde libre pour les bases Oracle 8i, 9i et 10g écrit en TCL/TK. Il nécessite l'installation du client Oracle pour fonctionner.

Hammerora-transforme des fichiers traces (de requêtes SQL, PL/SQL ou encore procédures stockées) en programme tcl.

Le tcl présente l'avantage d'offrir des interactions avec la base de données sans l'inconvénient d'avoir à recompiler les programmes générés. Il s'appuie pour cela sur les interfaces d'appels Oracle (OCI)[6]. Chaque programme utilisé ici est compatible tcl/oratcl et peut, par conséquent, être exécuté indépendamment d'Hammerora-en ligne de commande TCL shell.

L'outil permet donc de simuler de multiples utilisateurs accédant à une base pour y exercer une action. Le nombre de connexions virtuelles et le nombre d'itérations sont paramétrables.

En standard, Hammerora-fournit un schéma et des requêtes tpc-c (mode transactionnel donc). Nous avons, dans le cadre de ce test, préféré développer un environnement spécifique composé d'un schéma et d'une requête de type datawarehouse qui se trouve ci-dessous.

```
select l.region,sum(si.quantity) qty_sold
from locations l,warehouses w, inventories i,products p,sale_items si
where l.location_id=w.location_id
and w.warehouse_id = i.warehouse_id
and i.product_id = p.product_id
and p.product_id = si.product_id
group by l.region
```

Environ deux secondes sont nécessaires pour exécuter cette requête. Les tables les plus importantes sont locations et sale items avec respectivement

1946 et 534624 enregistrements.

La métrique retenue pour évaluer l'impact du caping et de la pondération sur les performances est le nombre de transactions par minute générées par swingbench.

La figure 4 représente l'interface des tests. Des outils comme GlancePlus [7] ou Xosview [8], permettent de visualiser graphiquement l'activité système de chacune des VM.

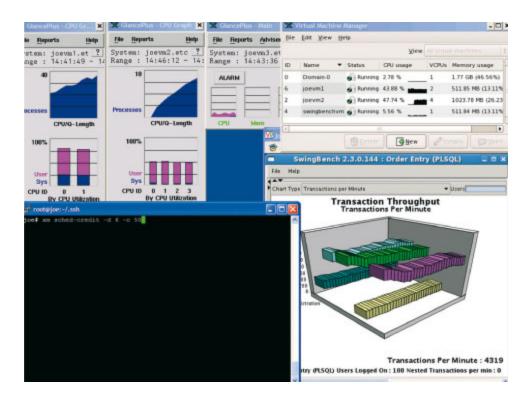


Figure 4 : Copie de l'écran de test

7. Résultats

Les problématiques d'atteinte de SLO dans les environnements consolidés et les outils de virtualisation ont servi de fil rouge à cet article. Comme nous l'avons vu, Xen fournit des environnements applicatifs isolés et sécurisés. Il fournit aussi des options de gestion de charge détaillées précédemment. Grâce à l'environnement de test, nous avons maintenant la possibilité de mesurer les écarts de performances et ainsi prouver le bon fonctionnement de ce concept (tableau suivant et figure 5).

Les étapes 1 et 2 montrent l'activité transactionnelle sans caping et à pondération identique. À l'étape 1, seul l'environnement OLTP – swingbench—est actif. À l'étape 2, hammerora-est démarré et entre en compétition avec

swingbench. Puis les VCPU de l'environnement de base de données OLTP sont bridées (étapes 3 et 4) avant que l'option de caping ne soit désactivée (étape 5). Dans un deuxième temps, le poids de la VM hammerora-est considérablement augmenté (64 fois supérieur (étape 6) puis 8 fois supérieur (étape 7) à la valeur par défaut). A l'étape 8, l'équilibre entre les pondérations est rétabli grâce à l'augmentation du poids de la VM swingbench. Pour finir, les deux VM sont ramenées à leurs paramètres initiaux (étape 9 et 10).

#	Domain id	Action	Tpm result
1	2, 4, 6	-c=0 -w=256 for all; startup swingbenh; hammer down	5400
2	2, 4, 6	Startup hammerora	4600
3	6 (swb db)	-c=50	1400
4	6 (swb db)	-c=25	550
5	6 (swb db)	-c=Ø	4200
6	2 (hammer)	-w=16384	220
7	2 (hammer)	-w=2048	1100
8	6 (swb db)	-w=2048	4700
9	2 (hammer)	-w=256	5300
10	6 (swb db)	-w=256	4400

Tableau de correspondance entre les actions de scheduling et l'impact sur les performances

La courbe ci-dessous (figure 5) reprend l'évolution des transactions au cours des différentes étapes du test.

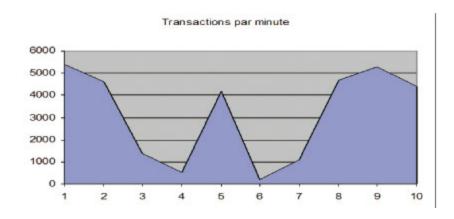


Figure 5 : Évolution du nombre de transactions

On constate, grâce au tableau précédent et à la figure 5, que l'impact des fonctions de seuil supérieur et de pondération sur les performances est réel. Nous avons volontairement fortement modifié l'allocation des ressources sur le système afin de «forcer le trait» et d'avoir des résultats probants. Une gestion affinée de ces paramètres serait plus réaliste dans une logique d'exploitation d'environnements de production.

Conclusion

Force est de reconnaître que les outils d'ordonnancement disponibles avec Xen ont un réel impact sur la capacité à allouer les ressources CPU aux domaines prioritaires. Bien entendu, cette fonctionnalité n'a d'intérêt que sur des serveurs souffrant de contention.

Notons aussi que, dans une problématique d'équilibrage de charge et d'atteinte de SLO, Xen offre la possibilité de migrer une VM d'un serveur physique vers un autre serveur physique sans interruption de service (ou presque). Cette caractéristique, appelée «Vmotion » chez VMware, est une alternative, à l'équilibrage de charge au sein d'un même serveur hôte. Il est aussi indispensable de rappeler que l'augmentation ou la diminution des priorités n'a pas un effet linéaire sur la performance transactionnelle d'une application. C'est le principe du goulet d'étranglement (bootleneck), bien connu des testeurs de performance. Ceci pour souligner que les résultats affichés dans cet article ne peuvent en aucun cas servir de comparateur de performance avec d'autres serveurs, OS ou solutions de virtualisation. Par ailleurs, nous n'avons pas eu la possibilité de mesurer la surcharge (overhead) générée par les processus Xen. Celle-ci est annoncée comme faible. Enfin, dans un souci de professionnalisation, il serait souhaitable d'associer les options de pondération de l'ordonnanceur à un outil de monitoring des performances afin de lever une alerte en cas de non-atteinte d'un SLO puis. dans un deuxième temps, de modifier le poids d'une VM afin de lui octroyer plus ou moins de temps CPU. Ceci offrirait l'avantage d'adapter automatiquement les ressources aux contraintes business sans intervention humaine.

Rérérences

- [1] http://www.cl.cam.ac.uk/Research/SRG/netos/xen/
- [2] http://www.xensource.com/
- [3] http://virt-manager.et.redhat.com/download.html
- [4] http://hammerora.sourceforge.net/
- [5] http://www.dominicgiles.com/swingbench.html
- [6] http://www.oracle.com/technology/tech/oci/index.html
- [7] http://h20229.www2.hp.com/products/gplus/index.html

• [8] http://xosview.sourceforge.net/

Glossaire

ms: milliseconde.

SLO: Service Level Objective ou Objectif de niveau de service. **Host**: Serveur physique exécutant Xen et hébergeant les VM.

VM: Machine virtuelle, invitée, domaine. VCPU: CPU virtuelle (une ou plus par VM).

CPU: CPU physique disponible sur le système hébergeur.

Tick: Unité de click d'horloge (10 ms).

Time-slice : Temps octroyé à une VCPU avant d'être remplacée par une autre

 $(30 \, \text{ms}).$

Poids: Partage proportionnel de ressources CPU par les VM.

Cap: Seuil supérieur optionnel du temps CPU consommable par une VM.

Retrouvez cet article dans : Linux Magazine 92

<u>Posté par</u> (<u>La rédaction</u>) | Signature : Yann Allandit | Article paru dans



Laissez une réponse

Vous devez avoir ouvert une session pour écrire un commentaire.

« <u>Précédent Aller au contenu</u> » <u>Identifiez-vous</u> <u>Inscription</u> <u>S'abonner à UNIX Garden</u>

Articles de 1ère page

- Git, les mains dans le cambouis
- PostgreSQL 8.3 : quoi de neuf?
- Introduction à Ruby on Rails
- <u>Linux Pratique HS N°17 Mars/Avril 2009 Chez votre marchand de journaux</u>
- 88 miles à l'heure!

- Développement et mise en place d'un démon Unix
- <u>Calculer ses rendus Blender en cluster ou comment faire sa propre</u> «render farm» avec DrQueue
- <u>Technologie rootkit sous Linux/Unix</u>
- CMake: la relève dans la construction de projets
- Des petits sondages pour améliorer nos magazines





Actuellement en kiosque :

Catégories

- • Administration réseau
 - o Administration système
 - o Agenda-Interview

- Audio-vidéo
- Bureautique
- Comprendre
- <u>Distribution</u>
- Embarqué
- Environnement de bureau
- Graphisme
- o Ieux
- o Matériel
- o News
- Programmation
- Réfléchir
- Sécurité
- Utilitaires
- o Web

Articles secondaires

• 30/10/2008

<u>Google Gears : les services de Google offline</u>

Lancé à l'occasion du Google Developer Day 2007 (le 31 mai dernier), Google Gears est une extension open source pour Firefox et Internet Explorer permettant de continuer à accéder à des services et applications Google, même si l'on est déconnecté....

Voir l'article...

7/8/2008

Trois questions à...

Alexis Nikichine, développeur chez IDM, la société qui a conçu l'interface et le moteur de recherche de l'EHM....

Voir l'article...

11/7/2008

Protéger une page avec un mot de passe

En général, le problème n'est pas de protéger une page, mais de protéger le répertoire qui la contient. Avec Apache, vous pouvez mettre un fichier ...htaccess dans le répertoire à protéger....

Voir l'article...

6/7/2008

hypermail: Conversion mbox vers HTML

Comment conserver tous vos échanges de mails, ou du moins, tous vos mails reçus depuis des années ? mbox, maildir, texte... les formats ne manquent pas. ...

Voir l'article...

6/7/2008

<u>iozone3</u>: Benchmark de disque

En fonction de l'utilisation de votre système, et dans bien des cas, les performances des disques et des systèmes de fichiers sont très importantes....

Voir l'article...

1/7/2008

<u>Augmentez le trafic sur votre blog!</u>

Google Blog Search (http://blogsearch.google.fr/) est un moteur de recherche consacré aux blogs, l'un des nombreux services proposés par la célèbre firme californienne....

Voir l'article...

■GNU/Linux Magazine

- <u>GNU/Linux Magazine N°113 Février 2009 Chez votre marchand de journaux</u>
 - Édito : GNU/Linux Magazine 113
 - Un petit sondage pour améliorer nos magazines
 - GNU/Linux Magazine HS N°40 Janvier/Février 2009 Chez votre marchand de journaux
 - o Edito: GNU/Linux Magazine HS 40

• <u>■GNU/Linux Pratique</u>

- <u>Linux Pratique HS N°17 Mars/Avril 2009 Chez votre marchand de journaux</u>
 - Édito : Linux Pratique HS N°17
 - o <u>Linux Pratique HS 17 Communiqué de presse</u>
 - <u>Linux Pratique Essentiel N°6 Février/Mars 2009 Chez votre</u> marchand de journaux
 - Édito : Linux Pratique Essentiel N°6

• **MISC Magazine**

- <u>Un petit sondage pour améliorer nos magazines</u>
 - MISC N°41: La cybercriminalité ...ou quand le net se met au crime organisé Janvier/Février 2009 Chez votre marchand de journaux
 - o Édito: Misc 41
 - o MISC 41 Communiqué de presse
 - Les Éditions Diamond adhèrent à l'APRIL!

© 2007 - 2009 <u>UNIX Garden</u>. Tous droits réservés .