TECHNIQUE D'ESTIMATION DU WORKING SET BASEE SUR LE PAGE MODIFICATION LOGGING (PML)

Stella BITCHEBE, Alain Tchana {Institut de Recherche en Informatique de Toulouse}
Laurent REVEILLERE {Laboratoire Bordelais de Recherche en Informatique}

Compas'2018

PML

Le PML est une amélioration matérielle dans le cadre de la **virtualisation**. Introduit par Intel, il étend les capacités des gestionnaires de machines virtuelles (telles que XEN) qui utilisent le mécanisme de table de pages étendue (**EPT**), en leur donnant la possibilité de traquer les pages mémoires (guest-physical pages) modifiées par les machines virtuelles (VM) lors de leur exécution [1].
Le schéma ci-contre résume le fonctionnement du PML. Lorsqu'une VM (domU) modifie une page en mémoire (bit dirty de la page passe de 0 à 1) :

- Le processeur (CPU) enregistre l'adresse de cette page (**guest-physical address : gpa**) dans un buffer appelé PMLog
- Lorsque le PMLog est plein (512 adresses enregistrées) le CPU génère une interruption (imputée à la VM) gérée par l'hyperviseur sous forme d'un VMExit : pendant ce vmexit, le PMLog est vidé et les pages concernées sont effectivement marquées **dirty** en mémoire centrale.

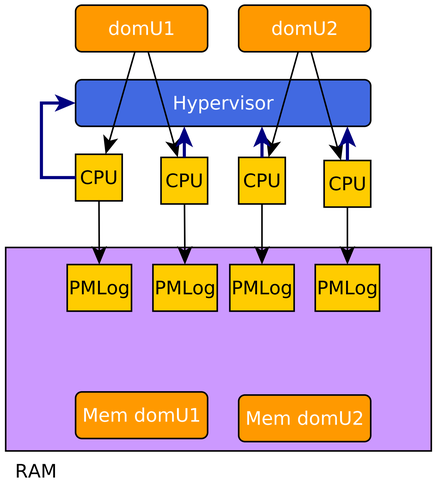

Figure 1: Fonctionnement global du PML

Problematique

Les utilisateurs ont tendance à surdimensionner les ressources de leurs applications ce qui a pour conséquences :

- **Un gaspillage de la mémoire**
- **Un faible taux de consolidation**
- **Une proportionnalité énergétique faible dans les datacenters**

La **mémoire** étant la **ressource critique**, il serait aventageux d'allouer à une VM la quantité exacte de mémoire dont elle a besoin à un instant donné. Il faudrait donc être capable d'anticiper sa demande (aussi bien à la hausse qu'à la baisse), i.e. de connaître à tout moment la taille de son **working set** (mémoire minimale dont la VM a besoin pour s'exécuter sans défaut de page).

Motivations

Techniques existantes basées sur des solutions logicielles :

- **Intrusives**
- **Surchargent la VM**
- **Surchargent l'hyperviseur**

Technique basée sur le PML :

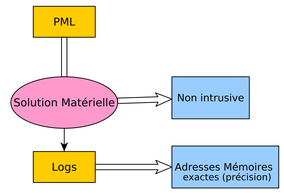

Figure 5: Avantages du PML

Contributions

- Mécanisme (hypercalls) d'activation et de désactivation du PML pour une VM : **xl enable-log-dirty * xl disable-log-dirty**
- Mécanisme de copie des logs consolidés de l'hyperviseur vers le dom0 (où sont effectués les calculs d'estimation)
- Améliorations matérielles résumées ci-contre :
  - Structure de données pour consolider les logs (bitmap) : stock de toutes les adresses enregistrées par le PMLog
  - Redirection de l'interruption lorsque le PMLog est plein : elle sera dorénavant gérée par le dom0, son coût ne sera plus imputé à la VM
  - Introduction d'un 2^{eme} buffer (PMLog)
- Simulateur pour tester ces modifications matérielles

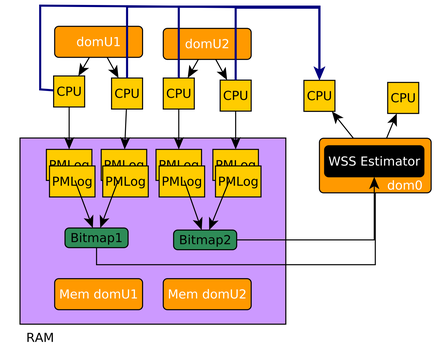

Figure 2: Fonctionnement du PML après amélioration

Resultats d'experience

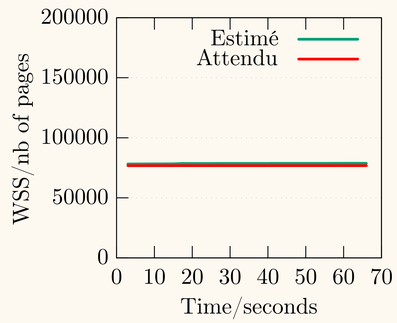

Figure 3: Expérience 1 - working set constant

- Hyperviseur : XEN
- VM
  - OS : Ubuntu server 16.04
  - RAM : 1Go
  - Disque : 50Go
- Expérience 1 : charge synthétique manipulant 300Mo (76800 pages mémoire) de mémoire.

L'algorithme d'estimation constiste à copier périodiquement les logs générés par le PML depuis l'hyperviseur vers le dom0 en comptant le nombre d'adresses distinctes dans ceux-ci.

- Expérience 2 : charge synthétique manipulant 300Mo de mémoire en 2 phases. Dans la 1^{re} phase, elle n'en manipule que la moitié soit 38400 pages mémoire, et dans la 2^{nde} phase la totalité.

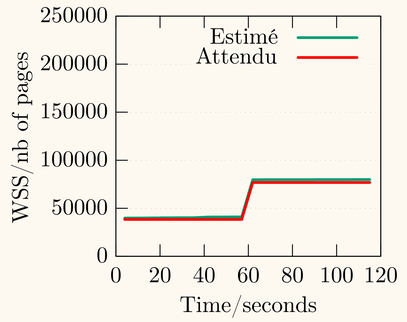

Figure 4: Expérience 2 - working set variable

References

[1] *Page Modification Logging for Virtual Machine Monitor White Paper*. 2015.
[2] Vlad Nitu et al. Working set size estimation techniques in virtualized environments: One size does not fit all. *Sigmetrics*, 2018.

Contact

Web www.enseeiht.fr
Email alain.tchana@enseeiht.fr