

PML PAGE MODIFICATION LOGGING

APERCU

Notre objectif est de pouvoir se servir des logs enregistrés par le mécanisme du PML pour estimer le working set des VM.

Pour le faire nous utilisons XEN, qui dans son code prend en compte le mécanisme lorsqu'il est supporté par le processeur. Seulement XEN n'implémente le PML que dans le cadre de la migration, donc pour pouvoir utiliser cette fonctionnalité, il faut d'abord activer le mode dirty log pour la VM dont on souhaite calculer le working set.

Premièrement nous avons donc implémenté des commandes xl qui permettent d'activer et de désactiver le mode log-dirty pour une VM.

ALGORITHME DE CALCUL

De façon globale les grandes étapes de l'algorithme sont :

- Activer le mode log-dirty : xl enable-log-dirty
- Collecter les logs pendant une période T. Pour définir T il faudra tenir compte de la taille du log, et du temps nécessaire pour remplir une page de log et effectuer les traitements, mais aussi des capacités de la machine. En effet :
 - o La taille d'une entrée de la page de log étant de 4KB, et le pml buffer enregistrant 512 entrées avant un VMExit, pour une bitmap de 512Mo par exemple nous avons déjà $(512 \times 1024 \times 1024) \times 512 \times 4 \times 1024$ octets de logs enregistrés
 - o Etant donné qu'un VMExit se produit chaque fois que la page de log est pleine, il faudrait éviter un temps de collecte trop grand ou de grandes quantités pour réduire le nombre de VMExit.
 - o Les capacités de la machine quant à elles concernent les temps d'accès mémoire, le temps de parcours des registres, la vitesse de calculs du processeur, etc.
- Désactiver le mode log-dirty : xl disable-log-dirty
- Traiter les logs : ici il faudra étudier le contenu du log pour déterminer quelles sont les pages qui constituent le working set de la VM. On pourrait partir sur le principe des pages les plus utilisées, et utiliser des algorithmes de machine learning pour l'apprentissage dans le cas des fluctuations ou des pics non uniformes d'utilisation.

COLLECTE DES LOGS

Pour la collecte des logs il faudra :

- Éliminer les pages superflues i.e. enlever des logs les entrées de la table des pages. Pour cela on pourra utiliser le bit reserve pour l'EPT et le mettre à 1 lorsque l'adresse enregistrée est en entrée de la table
- Compter le nombre de fois qu'une adresse est enregistrée : étant donné que le pml n'effectue le log que lorsque le bit dirty passe de 0 à 1, il faudra le remettre à 0 après un log pour pouvoir savoir au prochain log si la page a été réutilisée ou non. Ainsi on pourra compter le nombre de fois que chaque page est enregistrée et fixer un seuil minimal m d'utilisation pour qu'une page puisse être considérée dans le working set (ou alors se dire toute page utilisée fait automatiquement parti du working set : $m=1$)

- Utiliser 2 buffers pour le log : en effet lorsque le pmlog est plein (512 entrées), il se produit un VMExit. Pour pallier cette situation on pourrait utiliser 2 buffers de sorte que lorsque le 1^{er} est plein, on bascule sur le 2^{ème} qui continue d'enregistrer les logs pendant qu'on vide le 1^{er} (on copie ses logs dans la bitmap), ainsi au lieu d'effectuer un VMExit, on pourrait générer une exception qui sera capturer et gérée par l'hyperviseur.

EVALUATION

Une fois la solution mise sur pied, il faudra effectuer des tests et la comparer aux autres techniques de calcul de working set existant (Geiger, VMWare, etc.).

ARCHITECTURE

