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

Mémoire de fin d'études

Soutenu et présenté par :

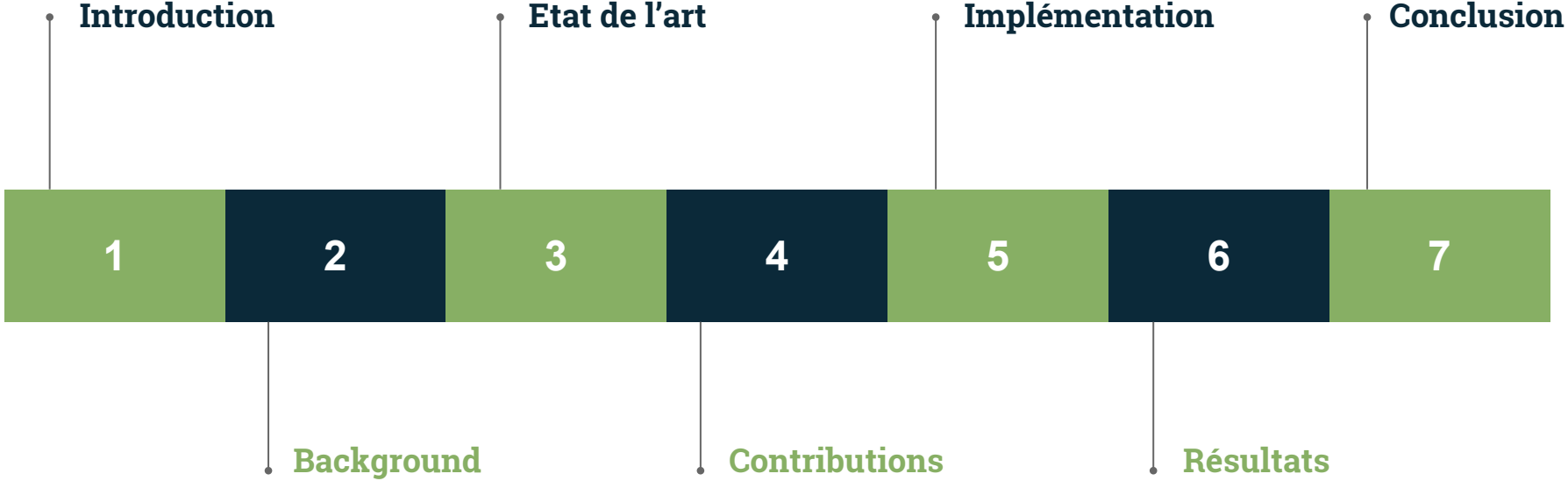
Célestine Stella N'DONGA BITCHEBE

En vu de l'obtention du :

Diplôme d'Ingénieur en Conception, option Génie Informatique

14 Septembre 2018

Grandes étapes



1

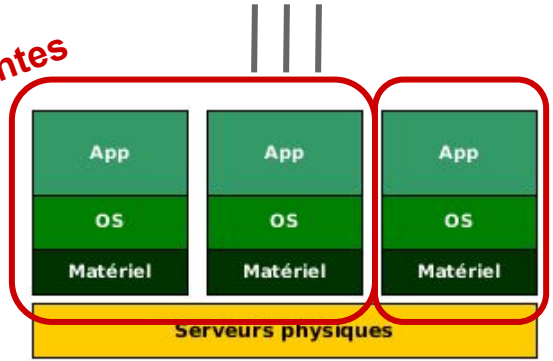
Introduction

- Situer le contexte
- Exposer la problématique
- Expliquer les motivations
- Présenter les objectifs

Contexte

Datacenter

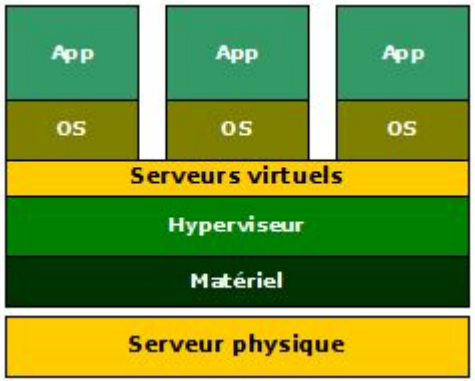
éteintes



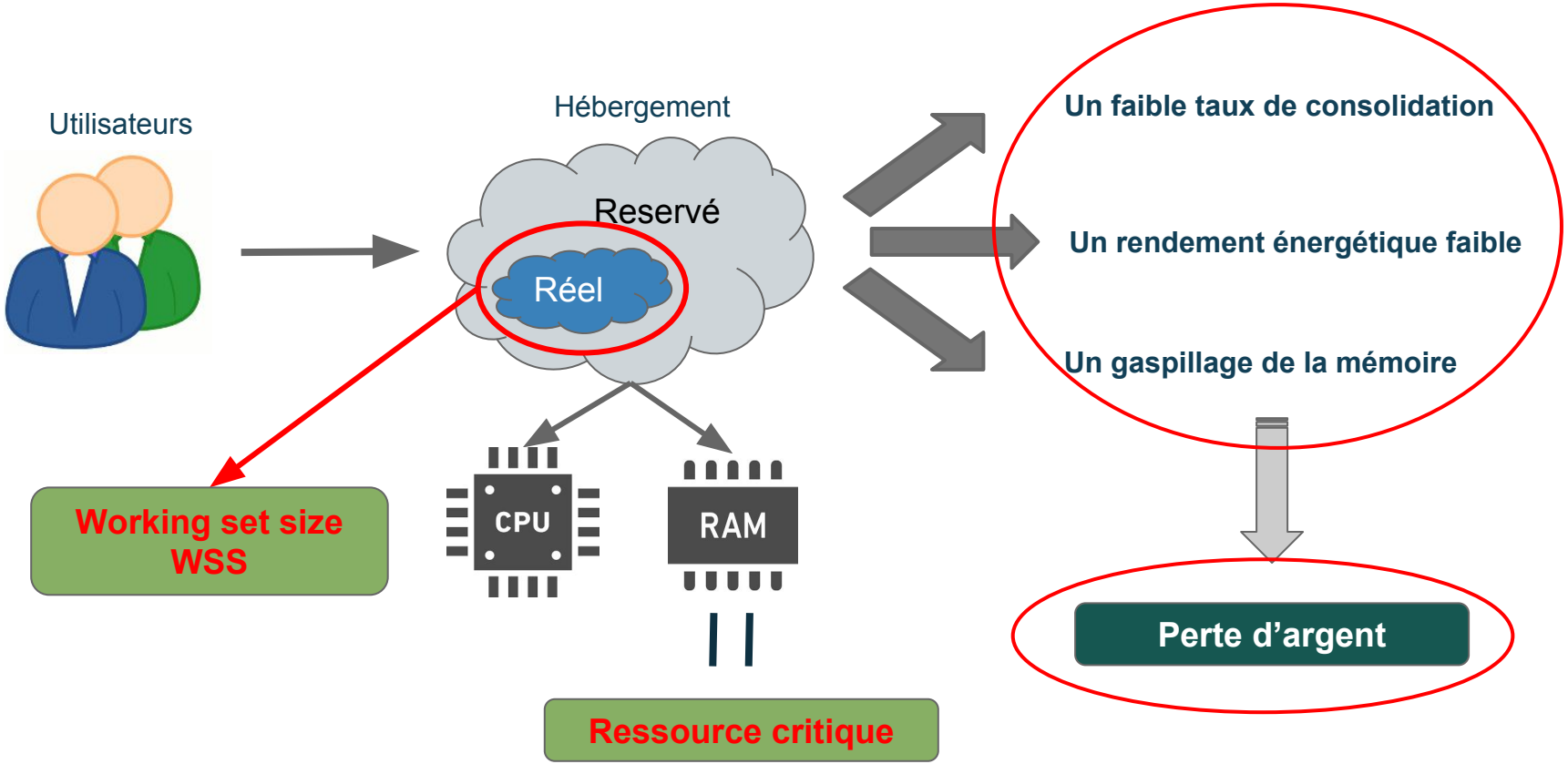
Virtualisation

50 - 70% des dépenses

Réduction de la consommation électrique



Contexte


Utilisateurs
Hébergement
Reservé
Réel
Un faible taux de consolidation
Un rendement énergétique faible
Un gaspillage de la mémoire
Working set size
WSS
CPU
RAM
Perte d'argent
Ressource critique

Problématique

Problème d'estimation du WSS

Techniques existantes

- **Lenteur**
- **Indisponibilité momentanée**
- **Etc.**

Comment estimer le working set sans dégradations de performances?

Inconvénients

- **Imprécision : sur/sous estimation**
- **Nécessitent beaucoup de ressources : surcharge des processeurs**
- **Nécessitent la modification du système des serveurs virtuels : intrusives**

Motivations & Objectifs

P M L

Page Modification Logging

Faciliter l'obtention des statistiques de working set

But

Intérêt

- Solution basée sur le mécanisme du PML = Solution **matérielle**
- Solution existantes = Approches **logicielles**
- Aucun travail de recherche sur le PML

Objectifs

- Étudier l'architecture actuelle du PML et ressortir les limites
- Proposer une architecture qui sied mieux au problème
- Définir un algorithme s'appuyant sur cette nouvelle architecture
- Evaluer et comparer

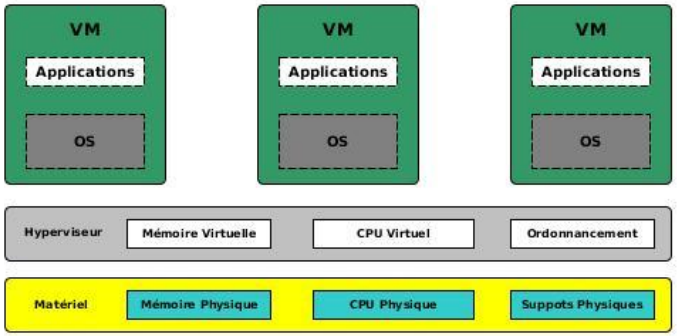
2

Background

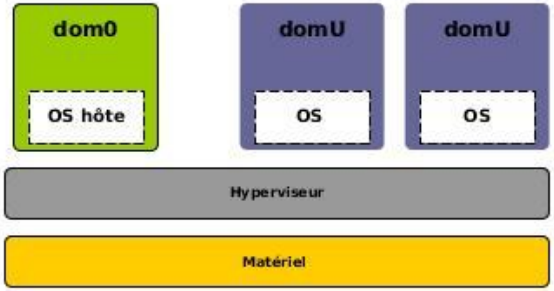
Présenter les concepts théoriques liés à la virtualisation et au PML

Généralités sur la virtualisation

Virtualisation = ensemble des techniques qui permettent de faire fonctionner simultanément sur une seule machine physique (machine hôte) plusieurs systèmes d'exploitation appelés machines virtuelles (VM)



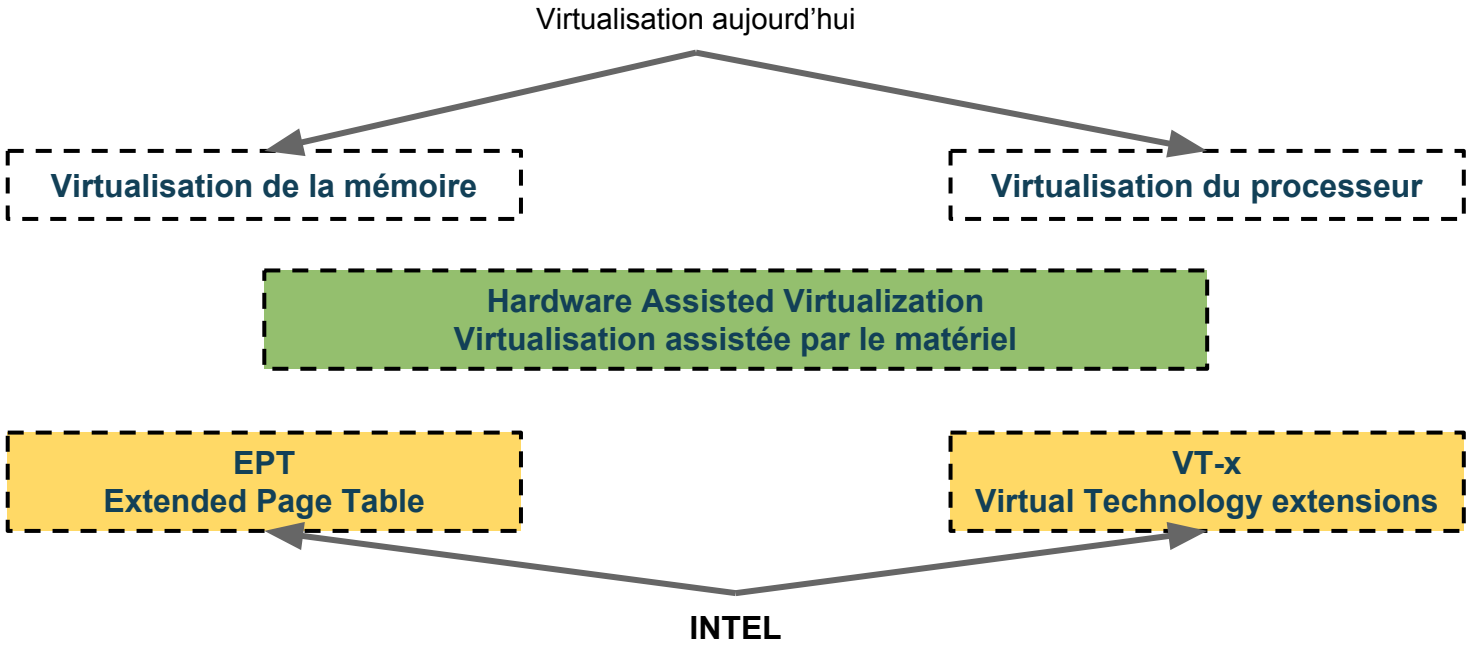
Outil logiciel qui permet de faire tourner les VMs sur la machine hôte



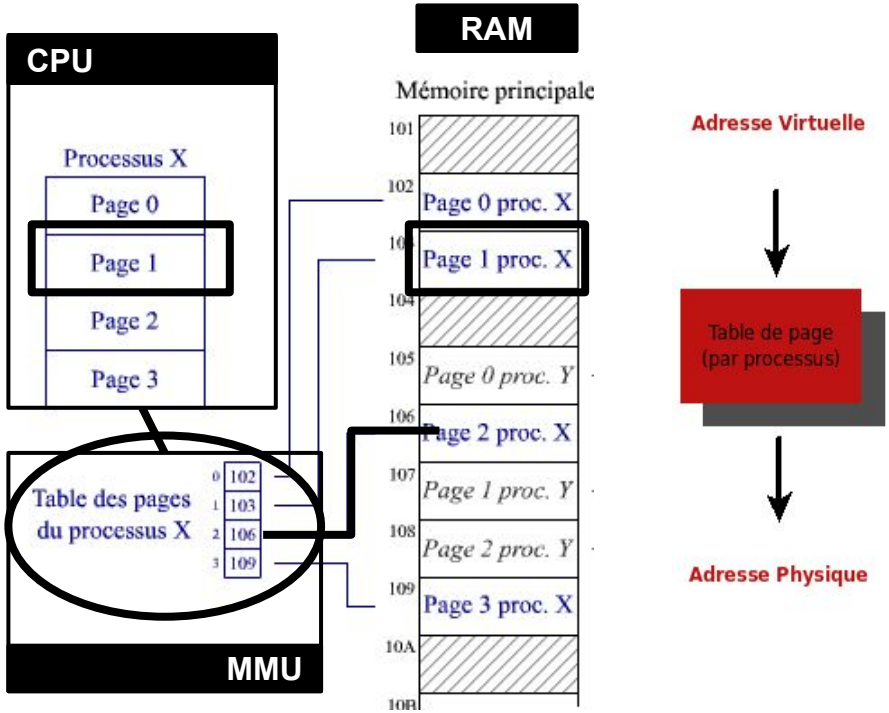
Xen

- → Assez populaire
- → Utilisé par Amazon dans ses datacenters
- → Code adapté pour le PML

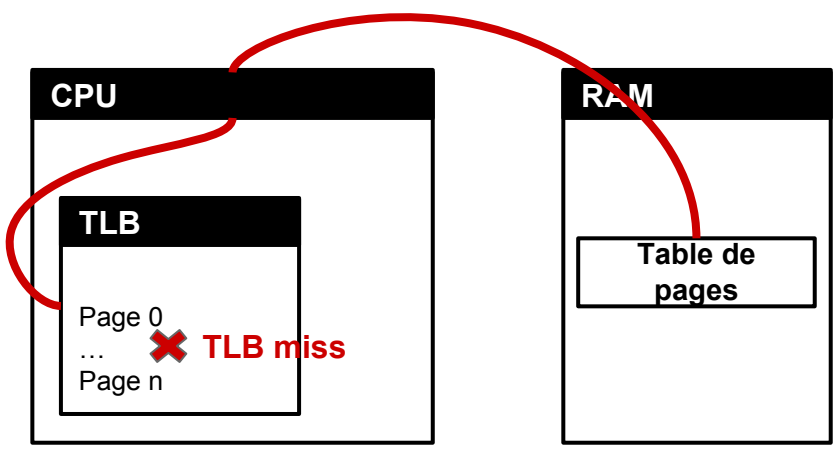
Virtualisation assistée par le matériel



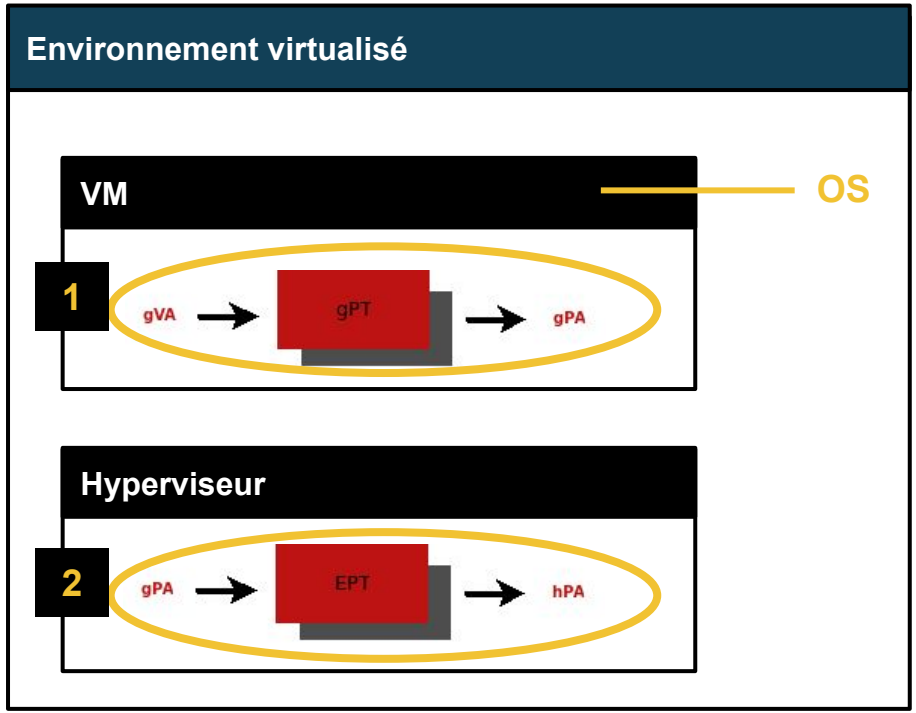
Virtualisation de la mémoire : Environnement natif



Translation Lookaside Buffer



Virtualisation de la mémoire : Environnement virtualisé (EPT)



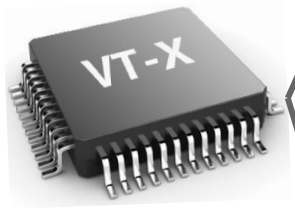
gPT : guest Page Table

gVA : guest Virtual Address

hPA : host Physical Address

gPA : guest Physical Address

Virtualisation du processeur : Intel VT-x



hyperviseur

vCPU
Virtual CPU

Hardware

CPU

VM

mode non-root

App

Guest OS

#vmexit

#vmlaunch
#vmresume

mode root

VMCS
Sauvegarde de l'état de la VM

hyperviseur

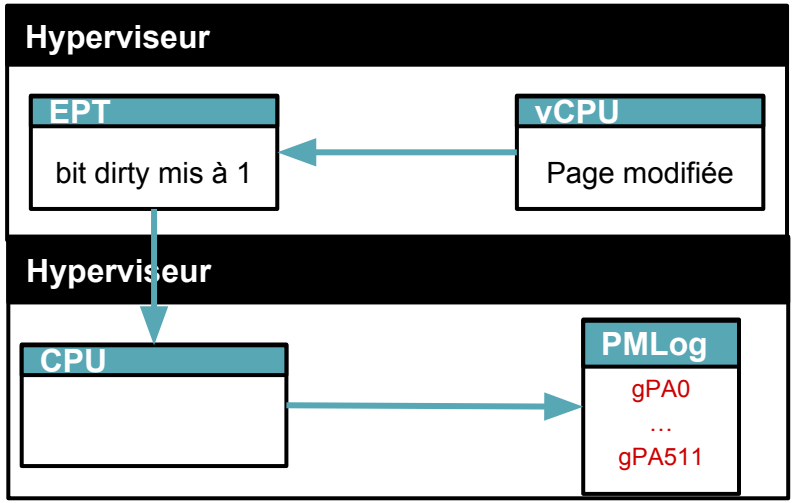
PML : Page Modification Logging

PML ⟹ virtualisation

Étend les capacités des systèmes de virtualisation utilisant le mécanisme d'EPT

Permet de traquer les pages mémoire que la VM modifie pendant son exécution

PML étend le procédé qui se produit lorsque les bits accessed et dirty sont mis à jour



Changements dans les VMCS

- **Enable PML** : lorsque le processeur supporte la fonctionnalité, il doit être mis à 1 pour activer le mécanisme.
- **Page modification log** : c'est la page mémoire dans laquelle sont enregistrées les gPAs. Elle contient 512 entrées de 64 bits.
- **PML Address** : adresse physique du page modification log.
- **PML index** : index de remplissage du page modification log. Il va de **511** à **0** .

3

État de l'art

Présenter les techniques existantes et leurs limites

Les métriques de l'estimation

Mémoire = **Ressource critique**

Allocation à la demande

- Collecter périodiquement les informations sur l'activité de la VM
- Estimer la quantité de mémoire dont la VM a besoin
- Ajuster la mémoire allouée à la machine

Q1 Comment observer la VM et collecter les informations sur son activité sachant que c'est une ***boîte noire***

- ➔ Méthode active : modifie le cours d'exécution de la VM
- ➔ Méthode intrusive : modifie la VM. Implémentée soit exclusivement à l'intérieur de la VM, soit répartie à travers la VM et l'hyperviseur ou le dom0. Nécessite l'accord du client.

Q2 Après avoir répondu à Q1, comment estimer le WSS de la VM à partir des données collectées

- ➔ Précision : sous-estimation ou sur-estimation
- ➔ Surcharge de la VM
- ➔ Surcharge de l'hyperviseur et/ou du dom0

Techniques existantes

Self-ballooning

Repose sur l'OS de la VM.

- **Réponse à Q1**

Valeur du ***Committed_As*** : nombre total de pages virtuelles allouées par un processus, même si elles ne correspondent pas forcément à des pages physiques en mémoire centrale :

cat /proc/mem/info

- **Réponse à Q2**

Incrémenter et décrémenter la valeur du ***Committed_As.***

- **Caractéristiques et Limites**

❏ Intrusive
❏ Imprécise

VMWare

S'appuie sur une approche d'échantillonnage.

- **Réponse à Q1**

Choisir aléatoirement et périodiquement n pages et les invalider (les marquer non-présentes ou en lecture seule).

- **Réponse à Q2**

Capture les exceptions et compter le nombre f de défauts de pages.

Estimer le wss par la formule : $(f/n)*m_{act}$

- **Caractéristiques et Limites**

❏ Non intrusive
❏ Active
❏ Surcharge de l'hyperviseur
❏ Imprécision si wss > m_{act}

Geiger

- **Réponse à Q1**

Observer les évictions et mises à jour éventuelles du cache.

- **Réponse à Q2**

Utiliser un buffer fantôme (mémoire supplémentaire allouée à la VM pour éviter qu'elle n'effectue des swaps) qui étend la mémoire physique de la VM.

Estimer le WSS par la formule :

$$m_{act} + m_{fantt} \text{ si } m_{act} > 0$$

- **Caractéristiques et Limites**

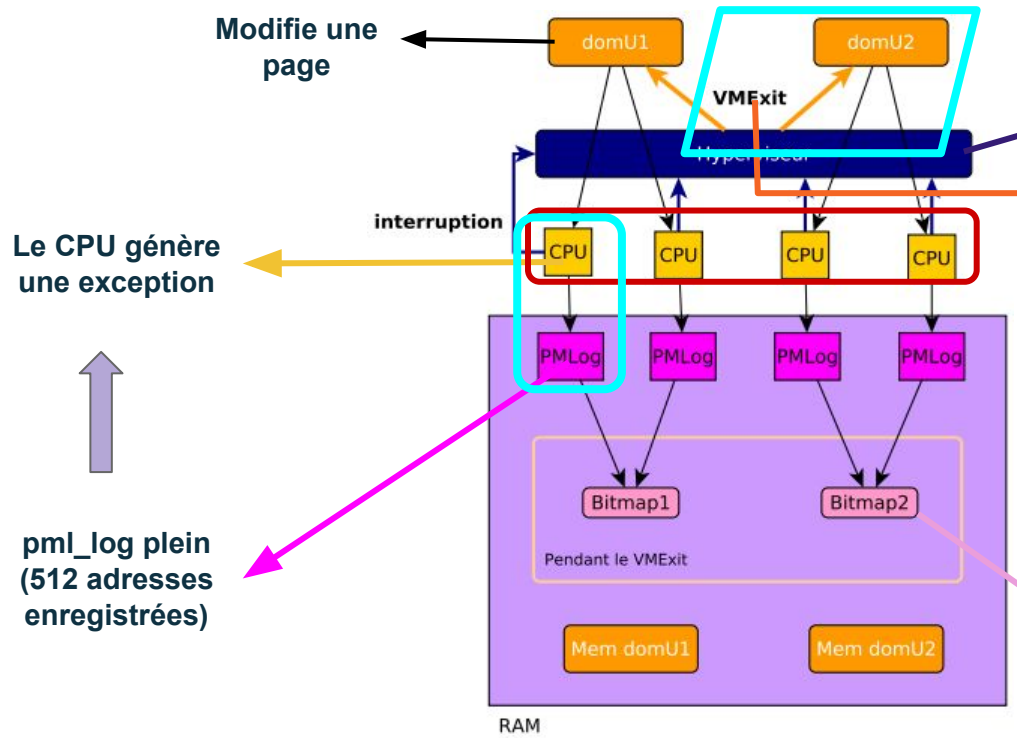
❏ Non intrusive
❏ Imprécise si $m_{act} <= 0$

4

Contributions

Présenter notre apport et comment la technique que nous proposons répond aux questions Q1 et Q2

Architecture actuelle du PML



L'exception est prise en main par l'hyperviseur et imputée à la VM sous la forme d'un VMExit

VMExit = passage du mode non-root au mode root
On passe de la VM à l'hyperviseur qui prend la main.
La VM arrête de s'exécuter, elle est comme en pause et donc plus aucune adresse n'est enregistrée

Pendant cette pause le pml_log est vidé : son index est remis à 511
Et la bitmap est mise à jour (constituée de bits représentant chacun une adresse, si elle a été modifiée le bit mis à 1)

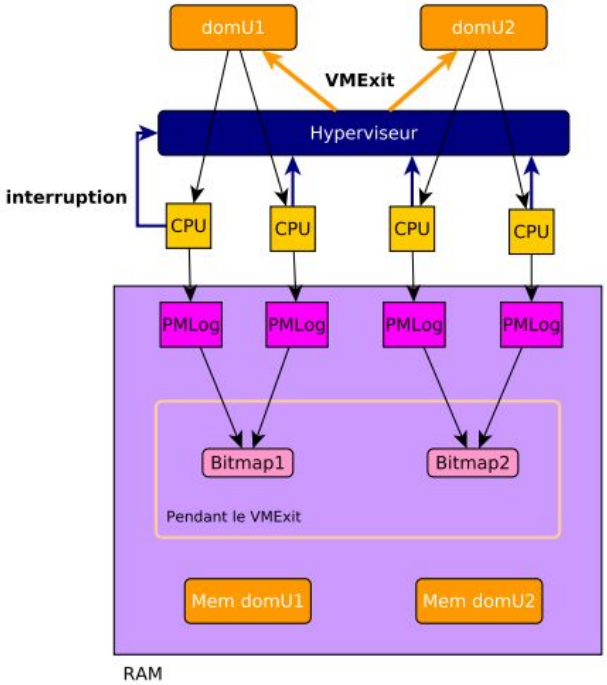
Limites de l'architecture actuelle

Limite 1 : *VMExit* imputé à la VM lorsque le *pml_log* est plein

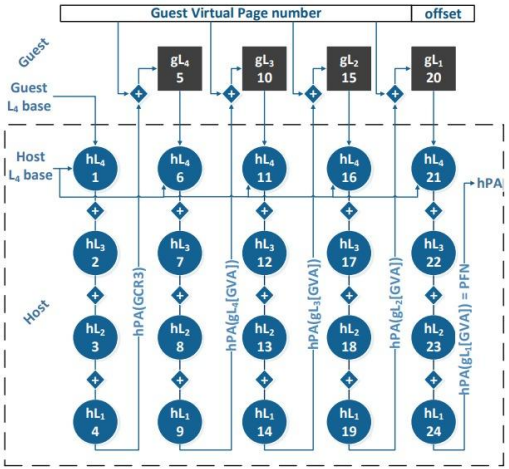
Limite 2 : taille du *pml_log* (4KB)

Limite 3 : uniquement les pages modifiées sont loguées

Limite 4 : la chaleur des pages n'est pas prise en compte



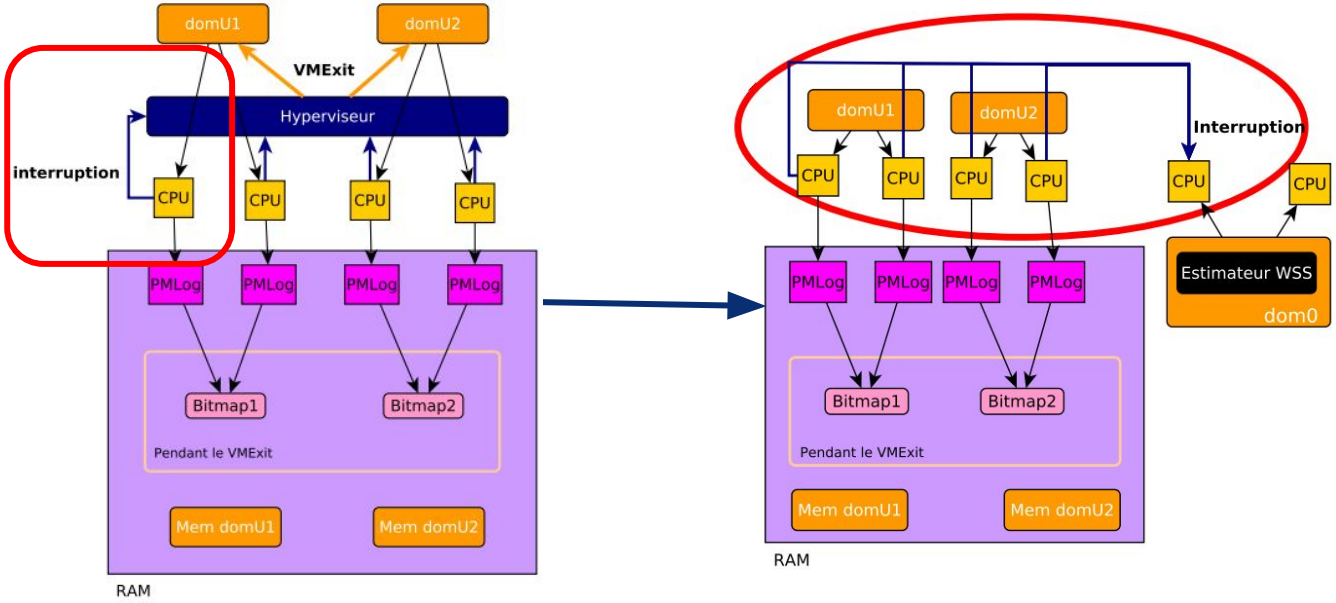
Limite 5 : les adresses des pages de la table de pages sont également enregistrées



Proposition d'une architecture améliorée

1. **Redirection des VMExits vers le dom0**

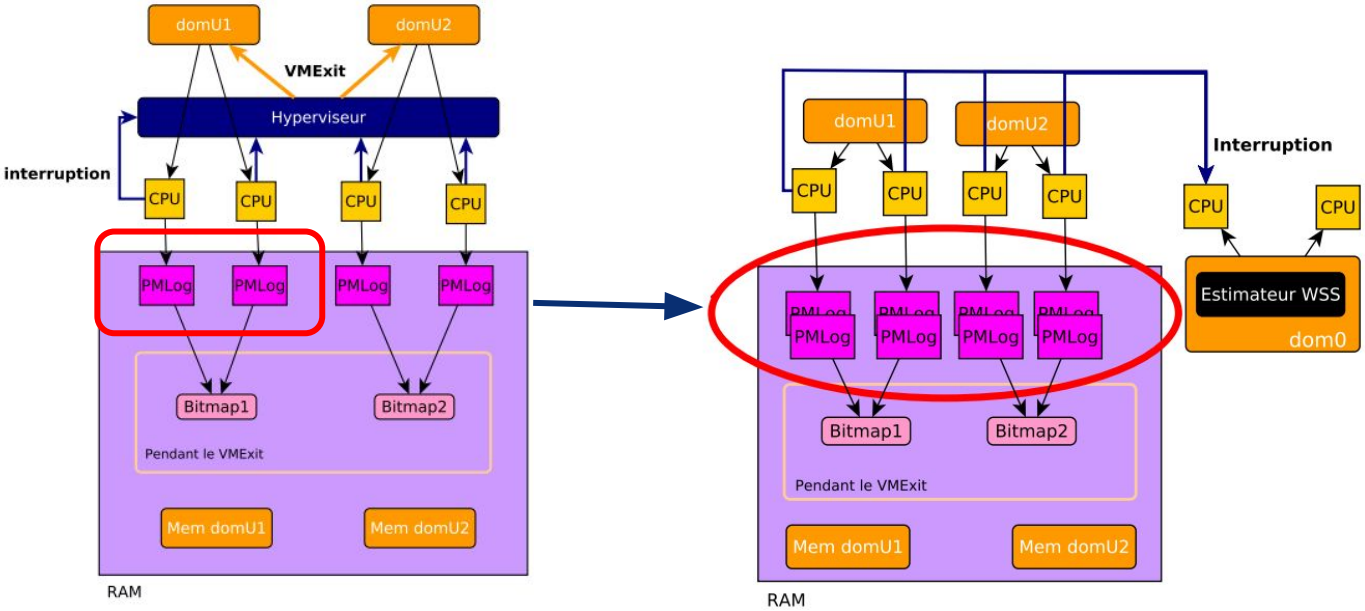
Limite 1 : *VMExit* imputé à la VM lorsque le *pml_log* est plein



Proposition d'une architecture améliorée

2. Introduction d'une 2ème page de log

Limite 2 : taille insuffisante du pml_log



Proposition d'une architecture améliorée

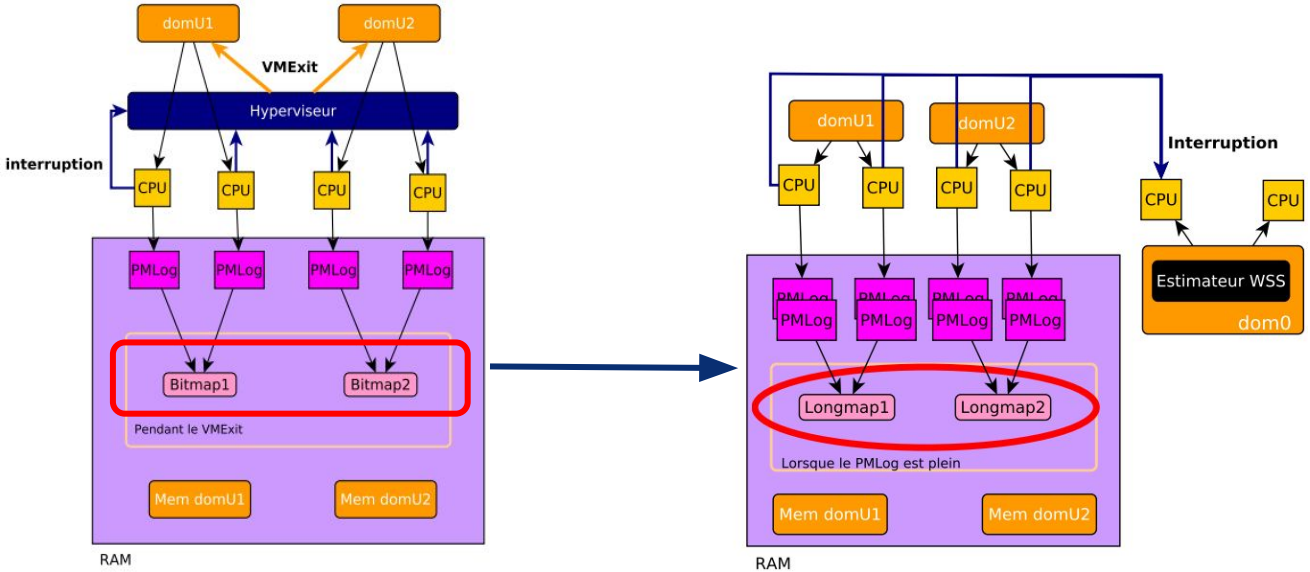
3. Remettre à 0 le bit dirty des pages

Limite 5 : les adresses des pages de la table de pages sont également enregistrées

4. Modification de la structure de données *bitmap*

Limite 3 : uniquement les pages modifiées sont loguées

Limite 4 : la chaleur des pages n'est pas prise en compte



Avantages et Inconvénients

Q1 **Comment observer la VM et collecter les informations sur son activité sachant que c'est une *boîte noire***

- Matériel collecte les informations sur l'activité de la VM
- Les consigne dans une structure de consolidation de logs

Inconvénient actuel = vmexit

Solution non intrusive

Solution = redirection vers le dom0

Pas de surcharge

Q2 **Après avoir répondu à Q1, comment estimer le WSS de la VM à partir des données collectées**

- n adresses dans le logs
- Taille d'une page = **4KB**

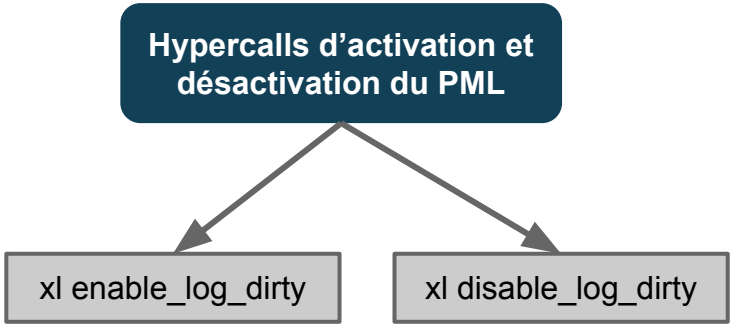
WSS = n*4KB

5

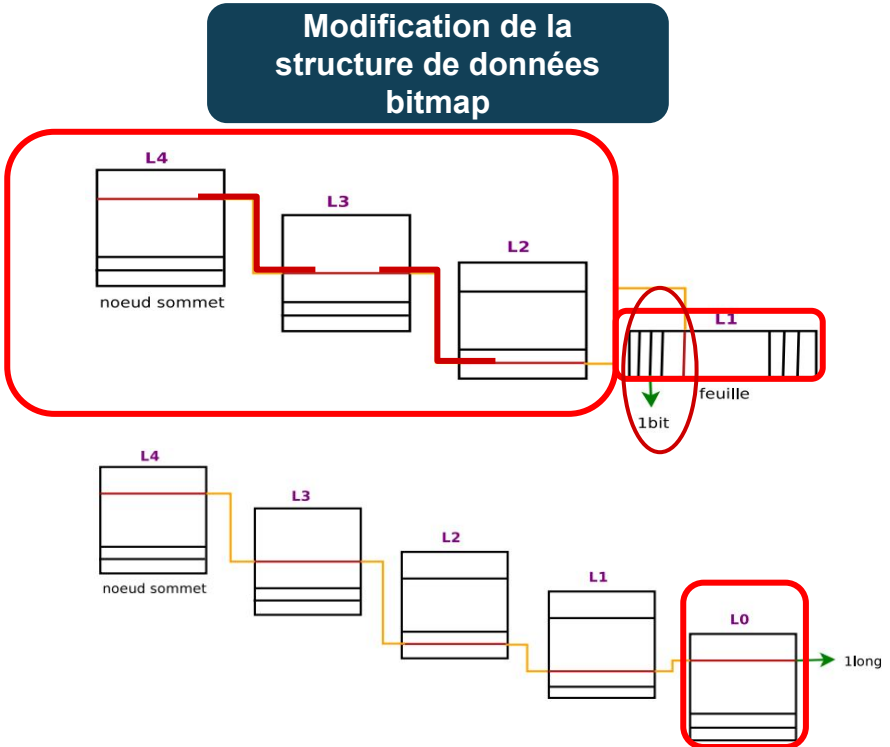
Implémentation

Présenter les détails de l'implémentation

Détails d'implémentation



- Le PML n'est pas activé par défaut
- XEN a modifié son code source source en accord avec le mécanisme
- Activation nécessaire au besoin

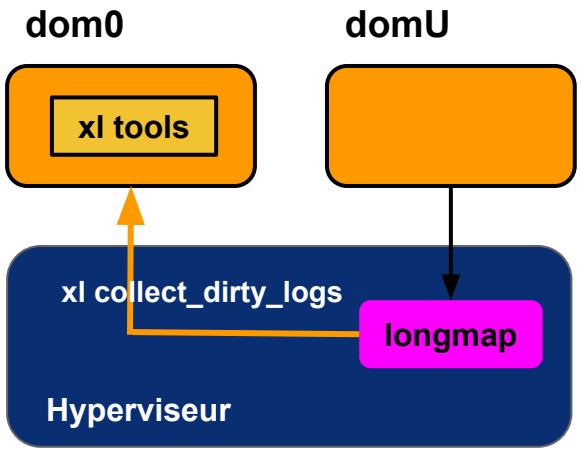


Détails d'implémentation

Modification du traitant *pml_buffer_full*

- En accord avec la nouvelle structure de consolidation de logs
- Lorsque le *pml_log* est plein notre algorithme parcourt la *longmap* pour retrouver l'entrée correspondant à l'adresse à enregistrée
- Si l'entrée est trouvée, le compteur est incrémenté sinon une nouvelle entrée est créée pour l'adresse et le compteur initialisé à 1

Hypercall de copie des logs consolidés de l'hyperviseur vers le dom0



6

Résultats

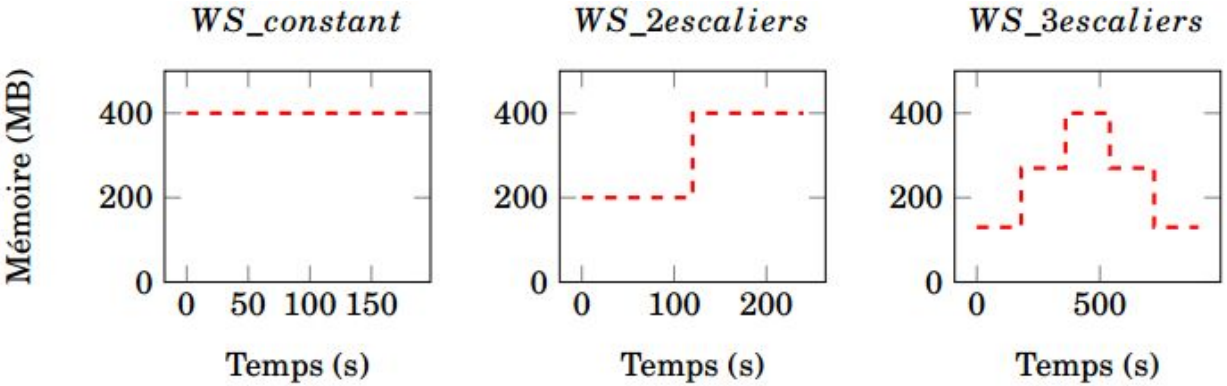
Présenter les des expérimentations menées

Expérimentations

Expérimentations menées avec des charges synthétiques

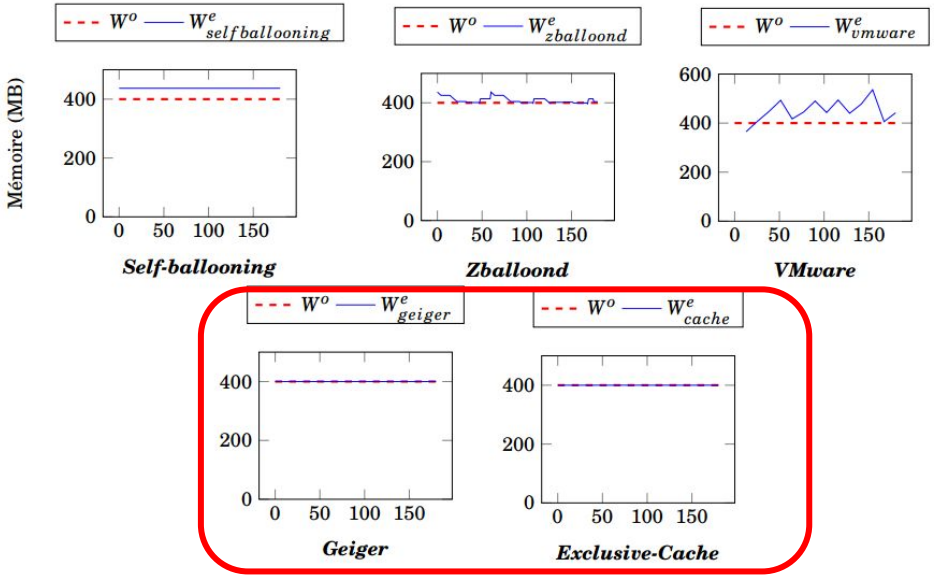
Charges manipulant 400MB de mémoire

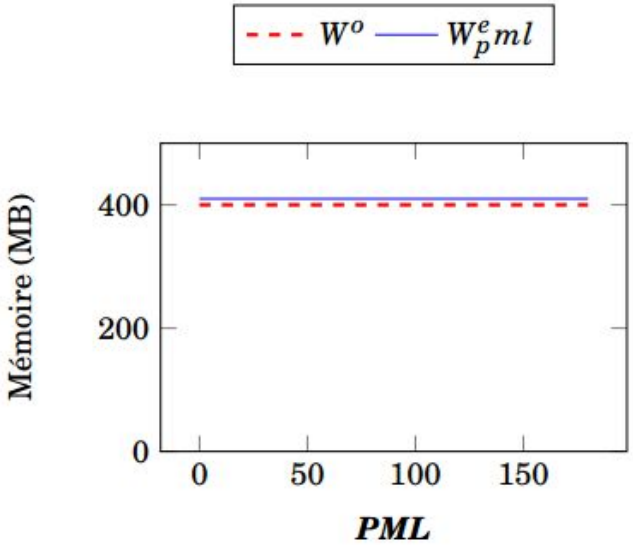
- **400MB = 400*1024 KB**
- **Taille d'une page = 4KB**
- **D'où 400MB → 400*1024/4 = 102400 pages**



Expérimentation 1

Charge synthétique constante

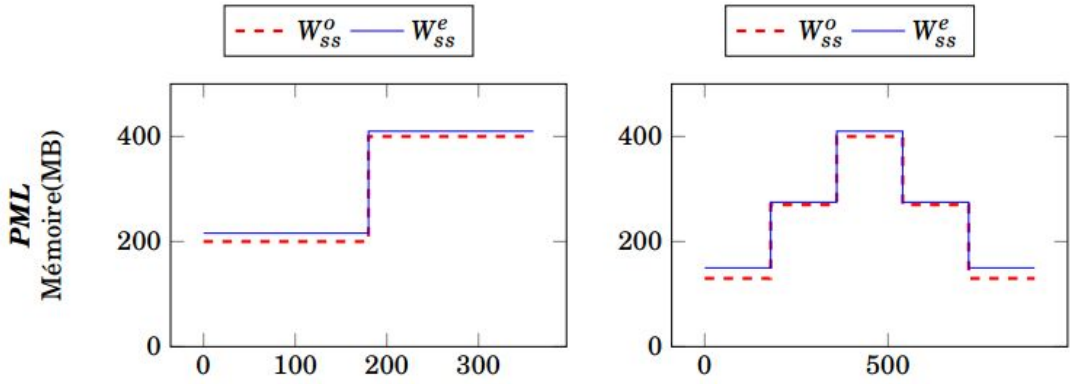


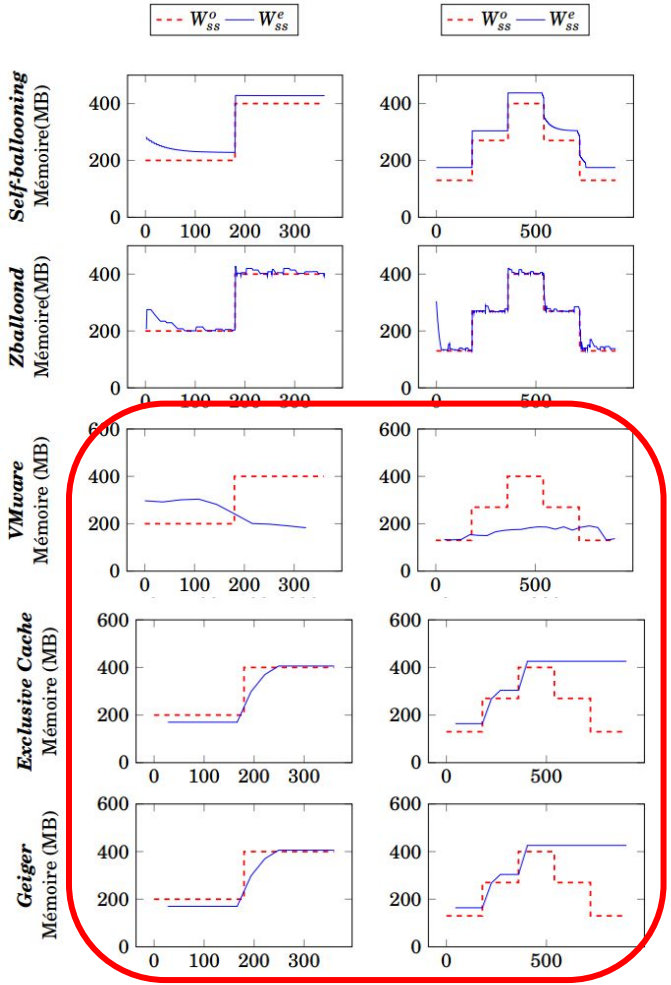


Expérimentation 2

Charges synthétiques variables

But : détecter les variations dans l'utilisation de la mémoire





7

Conclusion

Bilan et perspectives

Bilan

- ★ Nous avons conçu et implémenté une technique d'estimation du working set basée sur une amélioration matérielle des processeurs Intel : le PML (page modification logging)
- ★ Après avoir ressorti les limites de cette fonctionnalité dans le cadre de l'estimation du ws, nous avons proposé une nouvelle architecture qui répond mieux à ce problème
- ★ Les résultats des expérimentations que nous avons faites pour tester la technique implémentée dévoilent une estimation certes proche des valeurs attendues mais imprécises, ceci dues aux limites actuelles

Perspectives

- ★ Il est donc question par la suite d'implémenter un simulateur qui permettra de tester les performances de cette technique, car nécessitant des modifications matérielles elle ne peut être complètement implémenter dans un environnement réel

MERCI POUR VOTRE ATTENTION!

Des questions?

Technique d'estimation du *working set* basée sur le PML (Page Modification Logging)