Study Of Intel PML Effectiveness For Virtual Machine Working Set Size Estimation

PhD Student
Stella Bitchebe

SupervisorS
Pr. Alain Tchana
Pr. Laurent Réveillère

Research team
SCALE - I3S

Dresden-Germany March'19

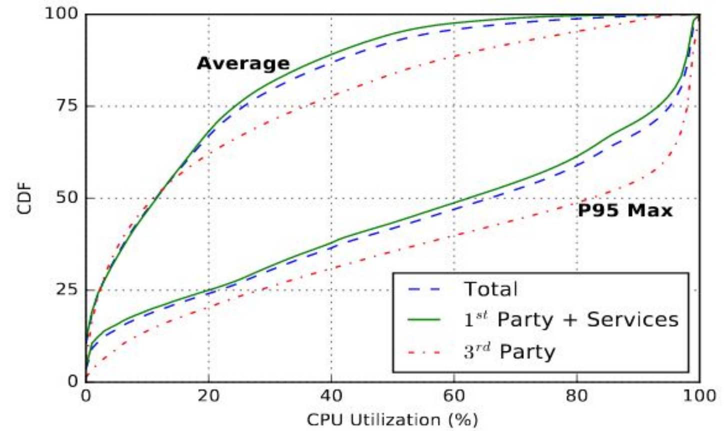
Introduction - Virtualization

- Basic building block in data centers
- Users reserve resources by booking VMs
- Overestimate VMs Sizes

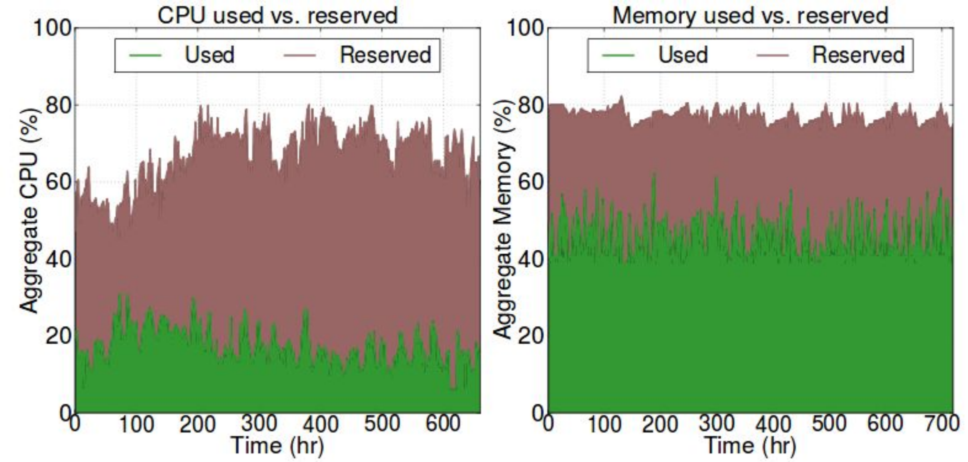
A large percentage of VMs require many fewer resources than their allocations

(SOSP'17)

There is a significant potential for oversubscribing physical resources



Twitter clusters CPU utilization [SOSP'17]



Azur clusters Memory utilization [ASPLOS'14]

On-demand allocation

Context

- To avoid resource waste
- Needs accurate and efficient estimation
- Especially for memory: limiting resource

Existing solutions

- All software base → Several drawbacks
- Examples
  - ❏ Self-Ballooning
  - ❏ Geiger
  - ❏ ZBalloon
  - ❏ VMWare

On-demand allocation

Context

- To avoid resource waste
- Needs accurate and efficient estimation
- Especially for memory: limiting resource

Existing solutions

- All software base → Several drawbacks
- Examples
  - ❑ Self-Ballooning
  - ❑ Geiger
  - ❑ ZBalloon
  - ❑ VMWare

VMWare Technique

- Periodically and randomly select n pages from the VM's memory and invalidate them
- Estimation is done by counting the number of pages which were subject to a non-present or read-only fault during the previous time interval

On-demand allocation

Context

- To avoid resource waste
- Needs accurate and efficient estimation
- Especially for memory: limiting resource

Existing solutions

- All software base → Several drawbacks
- Examples
  - ❑ Self-Ballooning
  - ❑ Geiger
  - ❑ ZBalloon
  - ❑ VMWare

VMWare Technique

- Periodically and randomly select n pages from the VM's memory and invalidate them
- Estimation is done by counting the number of pages which were subject to a non-present or read-only fault during the previous time interval

✗ VMWare not used in production Data Center (Confirmed by many cloud operators such as Nutanix, Eolas)

INTEL PML

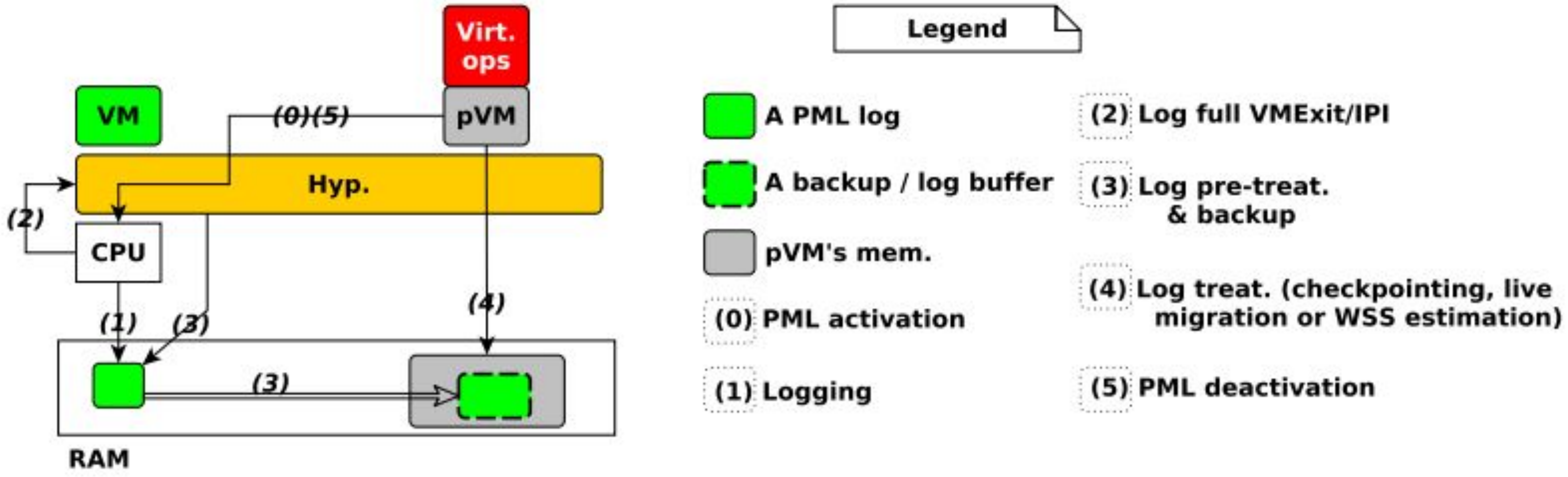
intel
vmware
intel
XEON
inside

Page Modification Logging

> “ *PML … provides a faster mechanism for a hypervisor to monitor all the memory pages that a guest VM modifies, in order to improve working page set analysis, checkpointing, and even VM migration.*

Richard Brunner, VMWare [https://www.theregister.co.uk/2016/05/11/dirty page logs coming to future vsphere release/]

PML Architecture



- Once PML is enabled, if the VM modifies a page the CPU logs its GPA (Guest-Physical address) inside a buffer (*PML logging buffer*)
- When the log is full (512 addresses logged), the CPU raises a VMExit which traps inside the hypervisor
- The handler of the VMEexit does certain task (e.g., copy the content of the *PML logging buffer* to a larger buffer which is shared with the pVM)
- Then the *PML index* is reset to 511 and the VM resumes

Contribution

1

Study of PML effectiveness for Live Migration, Checkpointing and Working Set Size estimation

- **Xen and KVM Implemented PML**
- **Xen use PML for Live migration**

Contribution

1

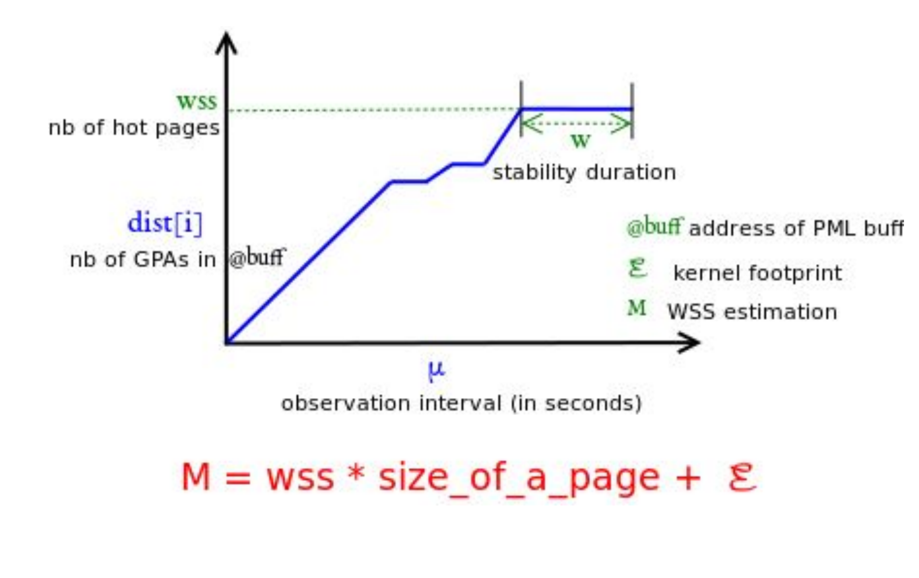
Study of PML effectiveness for Live Migration, Checkpointing and Working Set Size estimation

- **Xen and KVM Implemented PML**
- **Xen use PML for Live migration**

2

WSS estimation algorithm based on PML

- **With Xen Hypervisor**



$$M = wss * size\_of\_a\_page + \varepsilon$$

Testbed

On this machine:

PML is not yet

present on servers

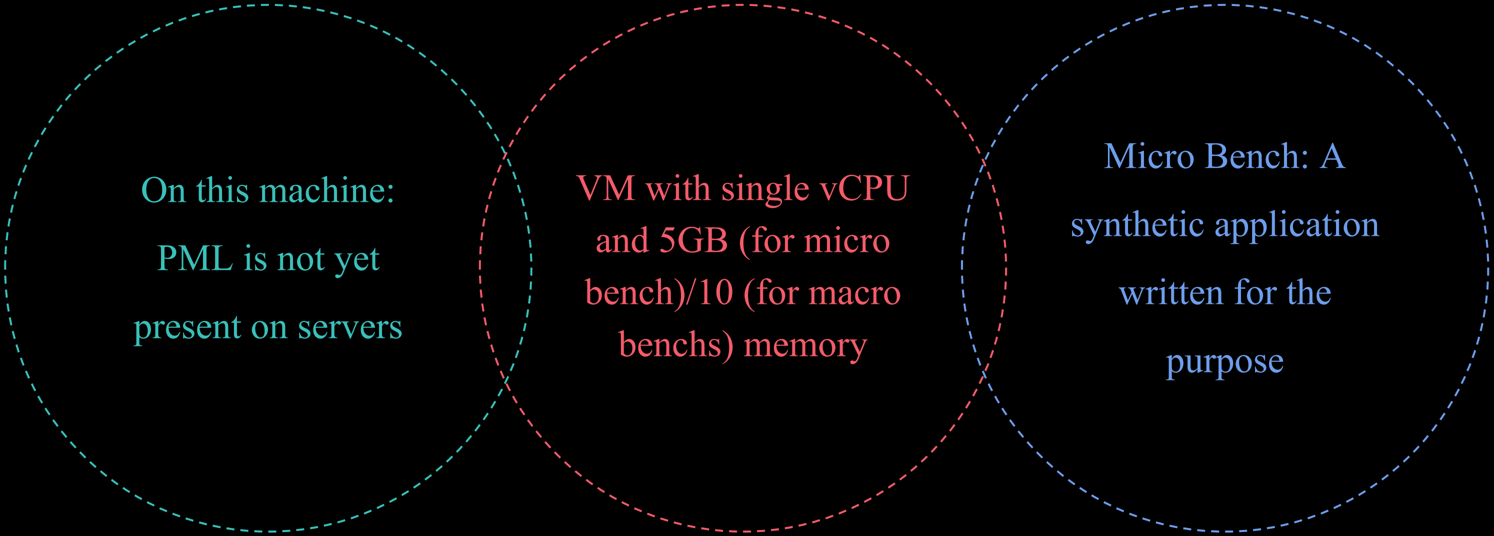
Testbed

On this machine:

PML is not yet present on servers

VM with single vCPU and 5GB (for micro bench)/10 (for macro benchs) memory

Testbed

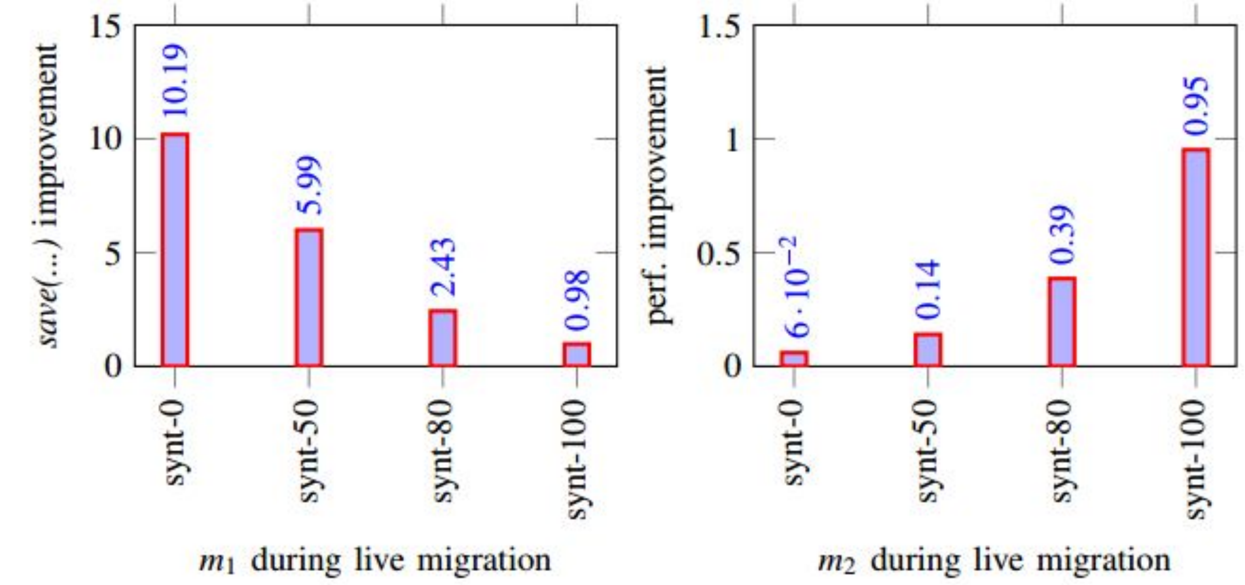


On this machine: PML is not yet present on servers

VM with single vCPU and 5GB (for micro bench)/10 (for macro benchs) memory

Micro Bench: A synthetic application written for the purpose

Results for Live migration

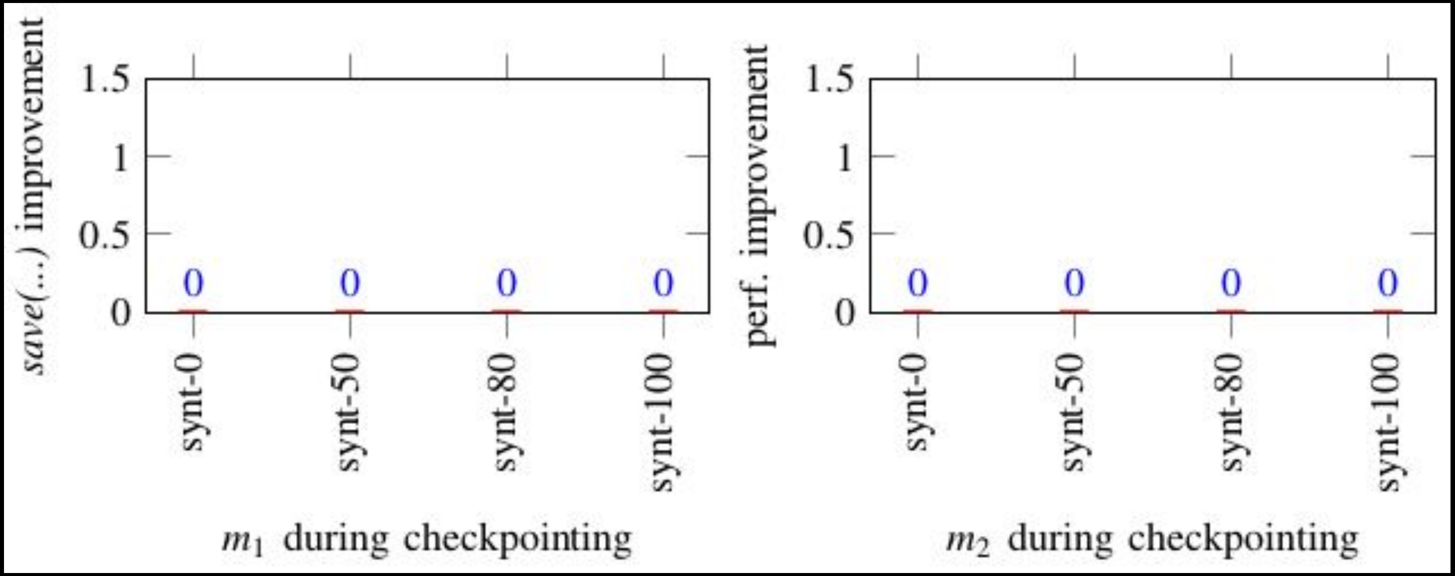


- Baseline: classical approach (write protecting)
- synt-x: x is the write intensity (100 means write only and 0 means read only)
- Average of 10 executions
- Metric 1 (*m1*): impact on live migration duration
- Metric 2 (*m2*): impact on user application performance

➔ PML reduces the duration of live migration by 0.98%-10.18%: especially for read intensive applications

➔ PML reduces the impact of live migration on user applications by 0.065%-0.95%: especially for write intensive applications

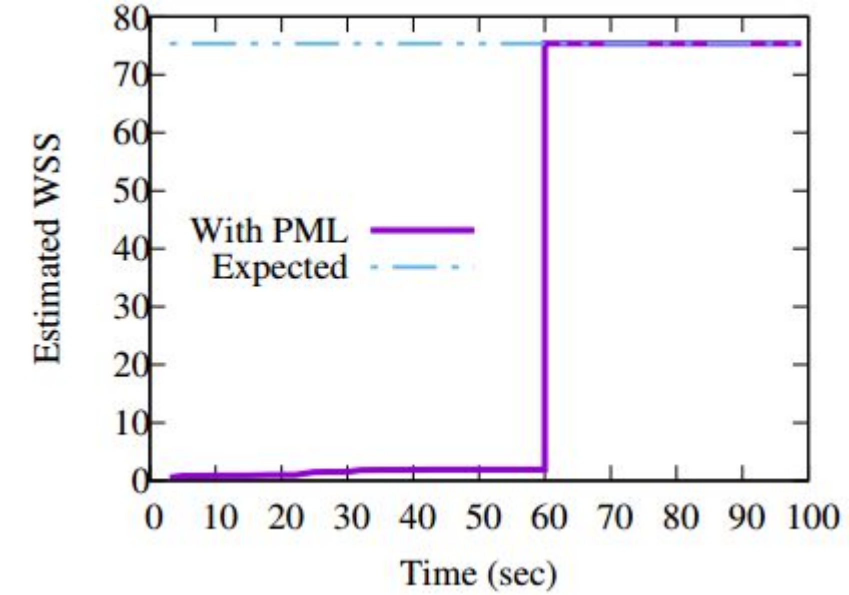
Results for Checkpointing



- ➔ **PML does not improve checkpointing**

- ➔ **Live Checkpointing is likely to take advantage of PML**

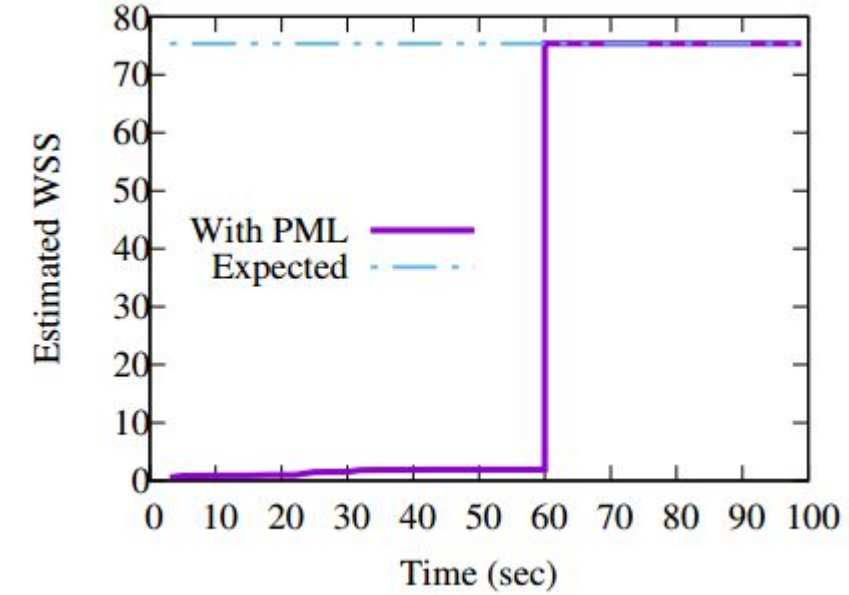
Results for WSS estimation

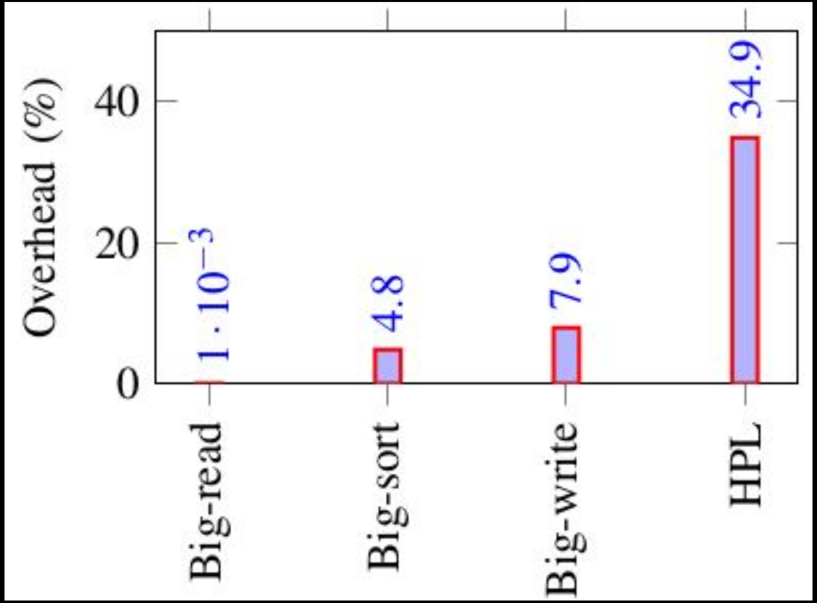


PML Hard Limit - PML is not able to accurately estimate a VM WSS

- ➔ PML does not log accessed pages
- ➔ With PML it is not possible to track hot pages

Results for WSS estimation





PML Hard Limit - PML is not able to accurately estimate a VM WSS

- ➔ PML does not log accessed pages
- ➔ With PML it is not possible to track hot pages

PML Soft Limit - PML is unfair for cloud users and their VMs

- ➔ The VMExit on *PML log buffer full* is handled by the CPU that runs the VM for which WSS is estimated

Conclusion - Ongoing Work

- Proposed Extension of PML: PRL (Page Reference Logged)
- PRL implementation under Gem5 simulator
- PRL-based WSS estimation algorithm

That was all!

ANY QUESTIONS?