Compas'2019

HARDWARE ASSISTED VIRTUAL MACHINE PAGE TRACKING

PhD Student : **Stella Bitchebe**[1]

Supervisor : **Pr. Alain Tchana**[1] & **Pr. Laurent Réveillère**[2]

[1]Université Cote d'Azur - I3S, [2]Université de Bordeaux - LaBRI

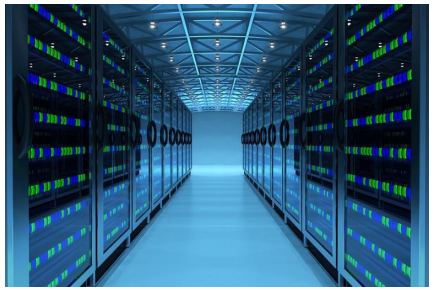
Anglet, June 2019

CONTEXT

CONTEXT

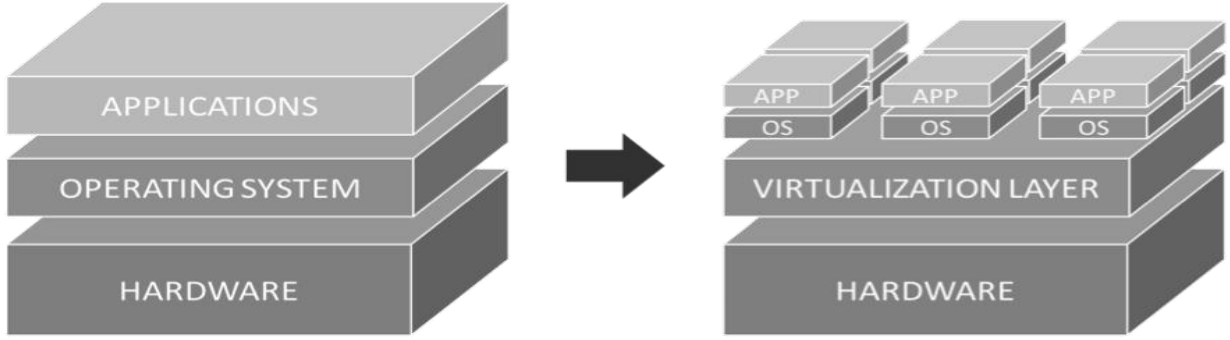
- Virtualization as the foundation of DataCenters.

Datacenter



50 - 70% of expenses

Energy saving



TRADITIONAL ARCHITECTURE

VIRTUAL ARCHITECTURE

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)
  - live migration (maintenance and dynamic packing)

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)
  - live migration (maintenance and dynamic packing)
  - working set size (WSS) estimation (overcommitment)

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)
  - live migration (maintenance and dynamic packing)
  - working set size (WSS) estimation (overcommitment)

- Widely used approaches for achieving page tracking are:
  - write protection
  - present bit invalidation

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)
  - live migration (maintenance and dynamic packing)
  - working set size (WSS) estimation (overcommitment)

- Widely used approaches for achieving page tracking are:
  - write protection
  - present bit invalidation

- Software-based lead to performance degradation.

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)
  - live migration (maintenance and dynamic packing)
  - working set size (WSS) estimation (overcommitment)

- Widely used approaches for achieving page tracking are:
  - write protection
  - present bit invalidation

- Software-based lead to performance degradation.

- Hardware Assisted Virtualization (HAV)

CONTEXT

- Virtualization as the foundation of DataCenters.

- Memory page tracking is a key demand for several essential tasks such as:
  - checkpointing (recovery after failure)
  - live migration (maintenance and dynamic packing)
  - working set size (WSS) estimation (overcommitment)

- Widely used approaches for achieving page tracking are:
  - write protection
  - present bit invalidation

- Software-based lead to performance degradation.

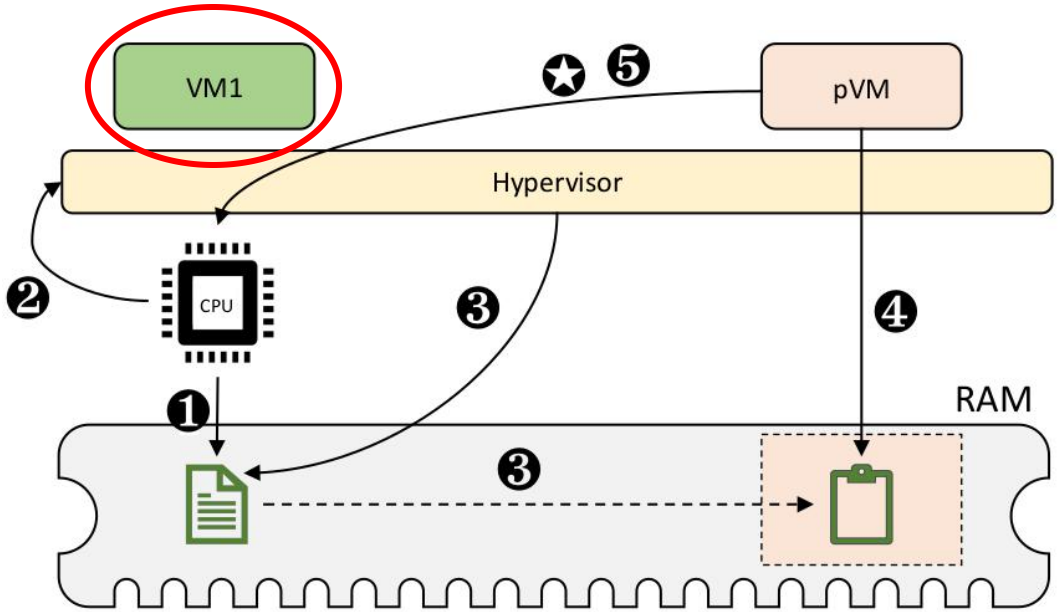
- Hardware Assisted Virtualization (HAV)

Page Modification Logging (PML)

PAGE MODIFICATION LOGGING

PAGE MODIFICATION LOGGING (PML)

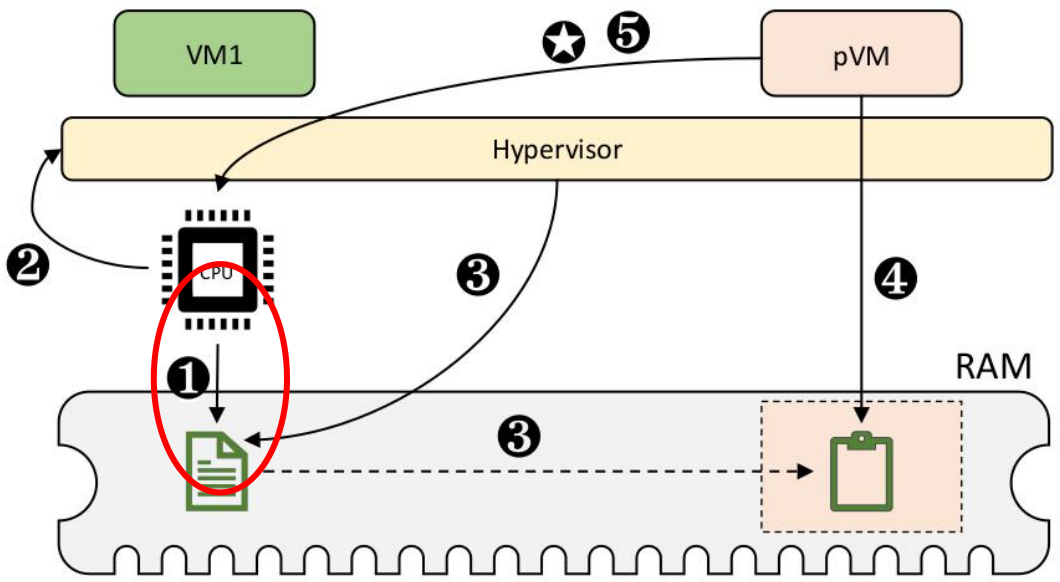
How does it work?



- PML log
- Backup / log buffer
- pVM's memory
- ✪ PML activation
- ❶ Logging
- ❷ Log full VMEXIT/IPI
- ❸ Log pre-treatment & backup
- ❹ Log treatment (checkpointing, live migration or WSS estimation)
- ❺ PML desactivation

PAGE MODIFICATION LOGGING (PML)

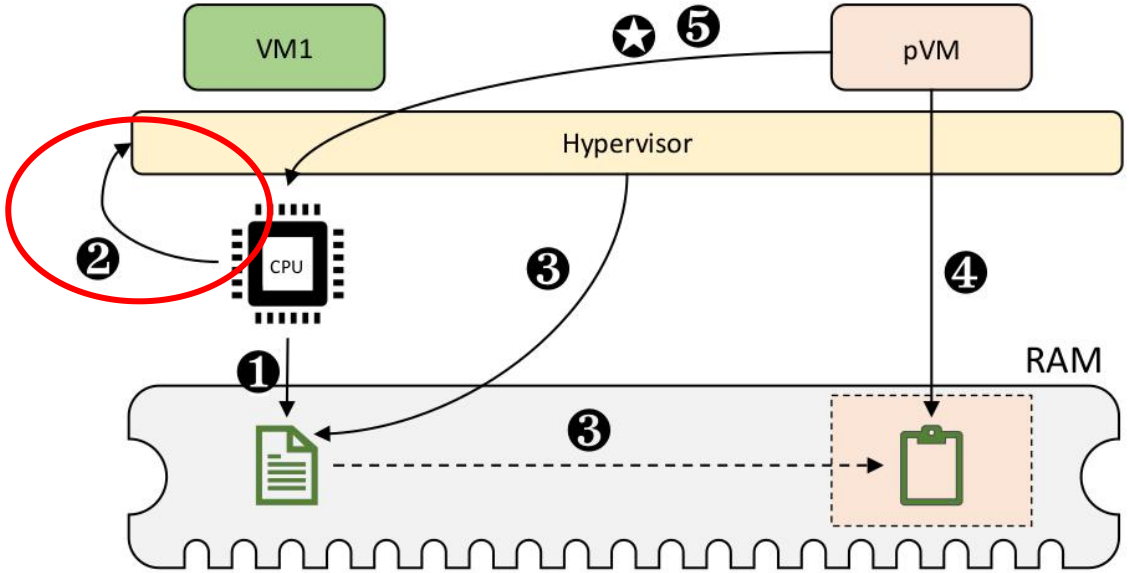
How does it work?



- PML log
- Backup / log buffer
- pVM's memory
- ✪ PML activation
- ❶ Logging
- ❷ Log full VMEXIT/IPI
- ❸ Log pre-treatment & backup
- ❹ Log treatment (checkpointing, live migration or WSS estimation)
- ❺ PML desactivation

PAGE MODIFICATION LOGGING (PML)

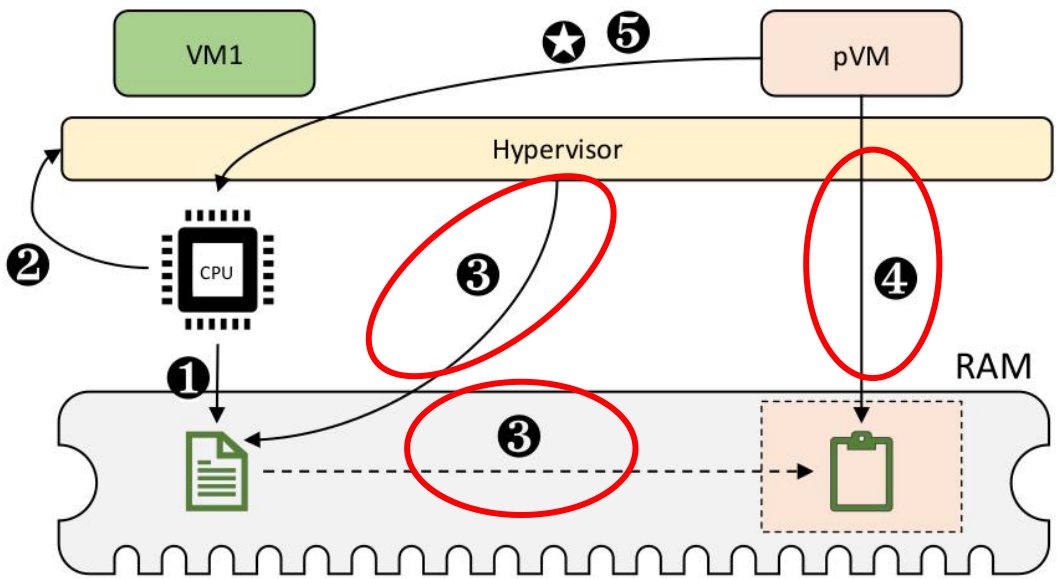
How does it work?



- PML log
- Backup / log buffer
- pVM's memory
- ✪ PML activation
- ❶ Logging
- ❷ Log full VMEXIT/IPI
- ❸ Log pre-treatment & backup
- ❹ Log treatment (checkpointing, live migration or WSS estimation)
- ❺ PML desactivation

PAGE MODIFICATION LOGGING (PML)

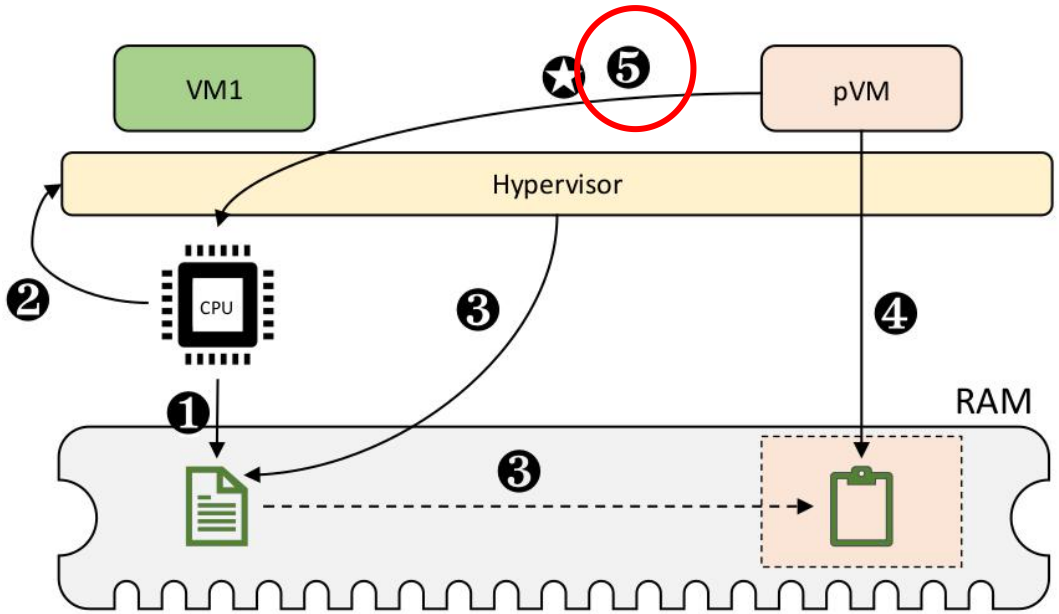
How does it work?



- PML log
- Backup / log buffer
- pVM's memory
- ✪ PML activation
- ❶ Logging
- ❷ Log full VMEXIT/IPI
- ❸ Log pre-treatment & backup
- ❹ Log treatment (checkpointing, live migration or WSS estimation)
- ❺ PML desactivation

PAGE MODIFICATION LOGGING (PML)

How does it work?



- PML log
- Backup / log buffer
- pVM's memory
- ✪ PML activation
- ❶ Logging
- ❷ Log full VMEXIT/IPI
- ❸ Log pre-treatment & backup
- ❹ Log treatment (checkpointing, live migration or WSS estimation)
- ❺ PML desactivation

PAGE MODIFICATION LOGGING (PML)

Study of PML from 3 angles

Energy consumption

Effectiveness

Performance overhead

PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Energy consumption

Workload: synthetic application that consists in parsing an array several time

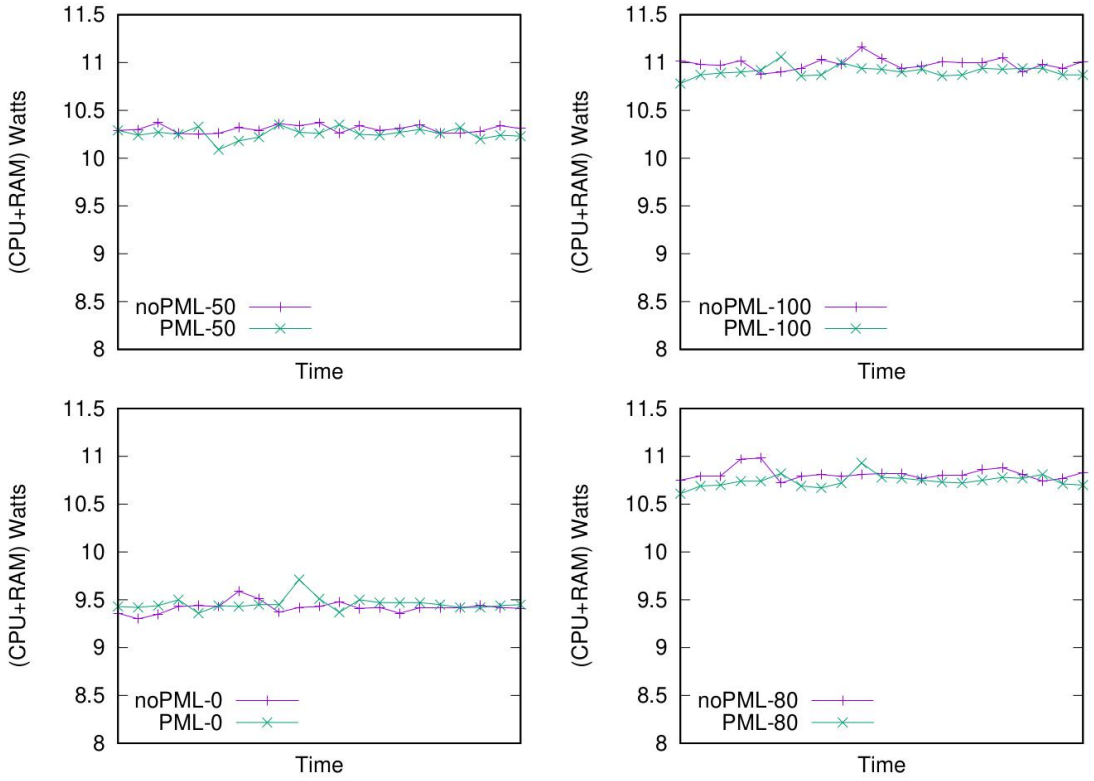
- Each entry points to a 4KB data structure (size of a memory page).
- The operation type: read or write.

PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Energy consumption

Workload: synthetic application that consists in parsing an array several time

- Each entry points to a 4KB data structure (size of a memory page).
- The operation type: read or write.

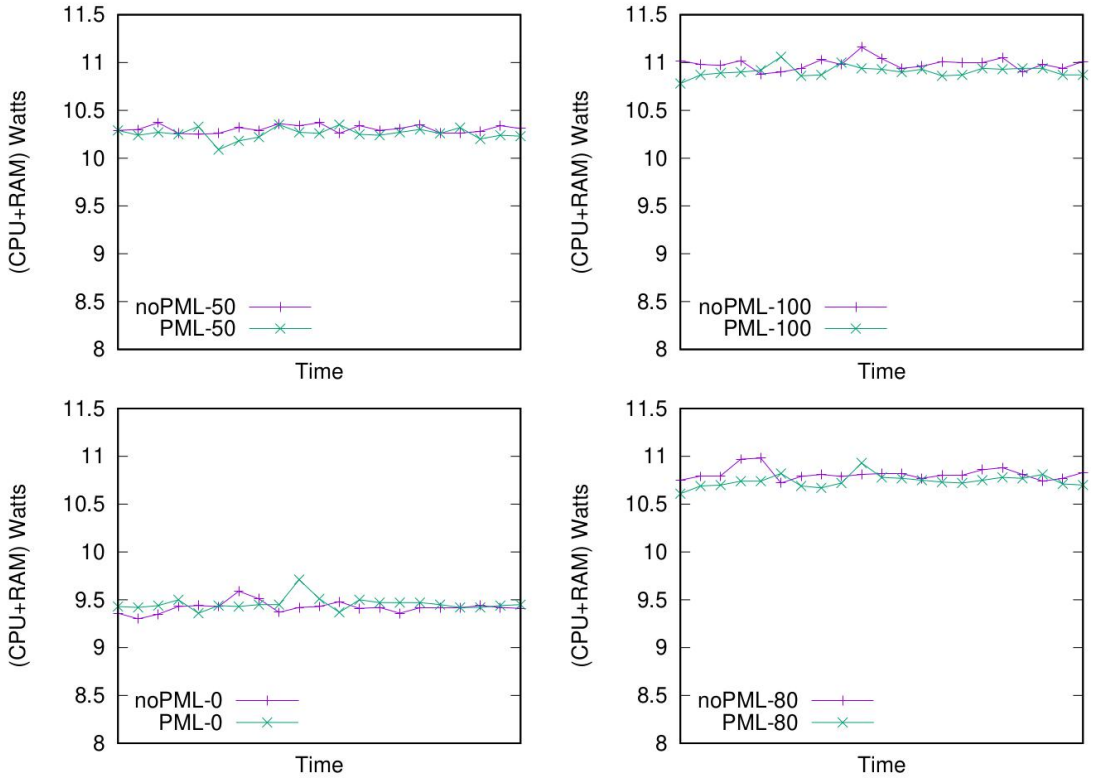


PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Energy consumption

Workload: synthetic application that consists in parsing an array several time

- Each entry points to a 4KB data structure (size of a memory page).
- The operation type: read or write.



✓ The energy consumption incurred by PML is almost nil.

✓ The energy concumption slightly reduces when the write intensity increases (up to 0.02%): write intensive workloads lead to much VMExits due to *PML log buffer full*, thus reducing the utilization of the CPU.

PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Effectiveness for live migration and checkpointing

- *static int save(…)* method: the intervention scope of PML in the XEN hypervisor.

PAGE MODIFICATION LOGGING (PML)

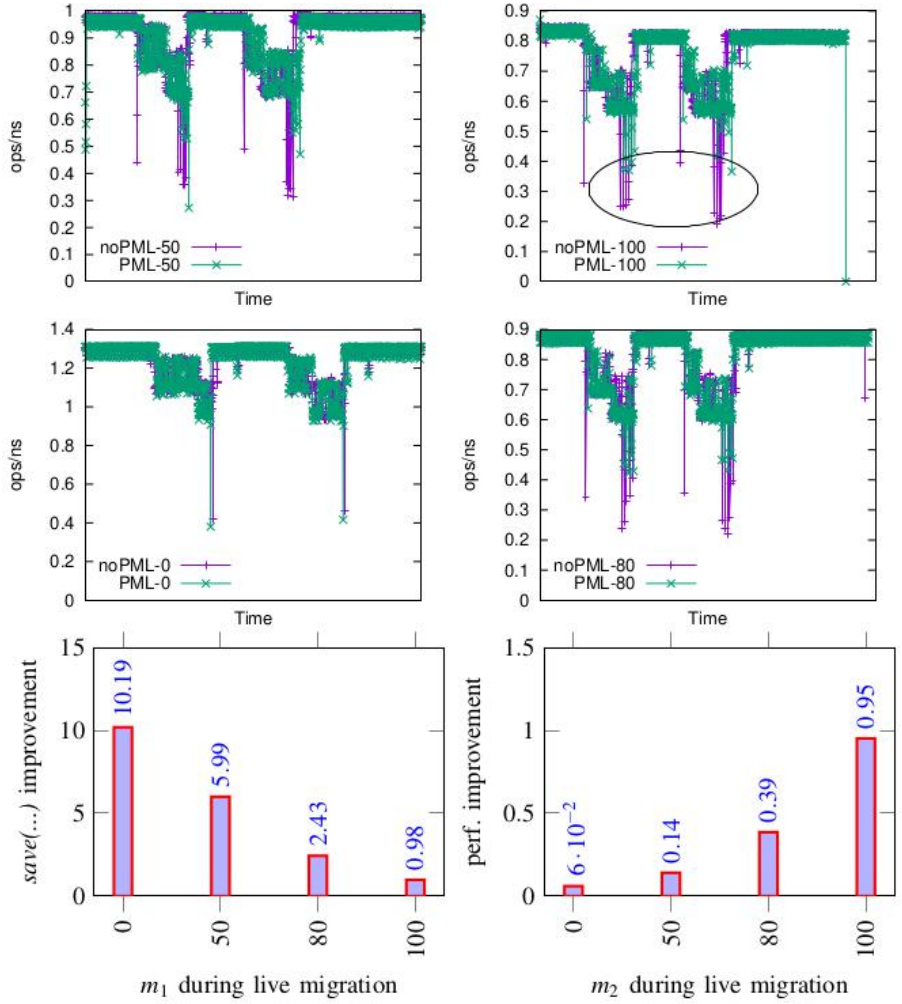
Study of Intel PML: Effectiveness for live migration and checkpointing

- *static int save(…)* method: the intervention scope of PML in the XEN hypervisor.
- Base line: the classical memory page tracking approach which consists in write protecting memory pages so that next writes lead to page faults.
- Metrics:
  - m_1 the execution time of save(…) method. Tells whether PML accelerates checkpointing/migration
  - m_2 the performance of the user application. Allows to check whether PML reduces or increases the negative impact of these operations on the user application.

PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Effectiveness for live migration and checkpointing

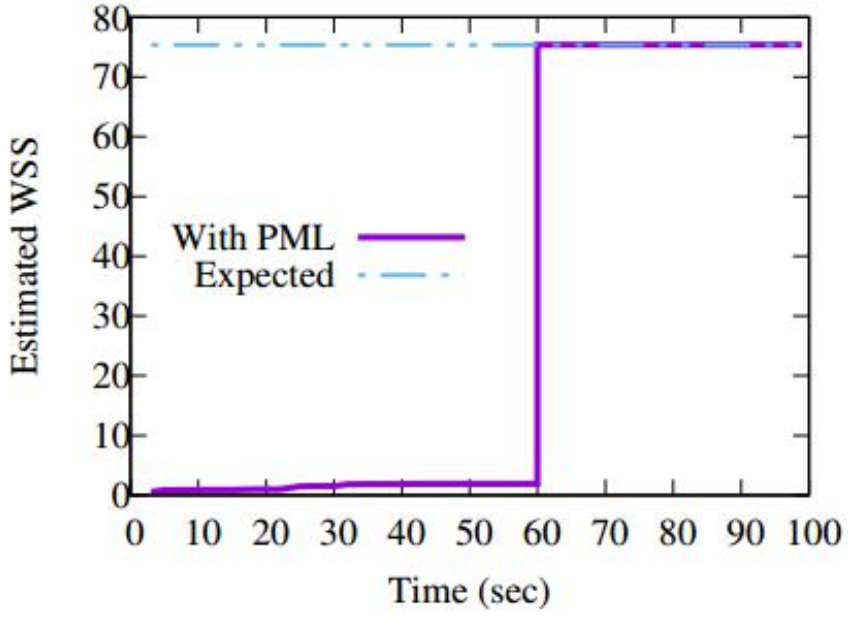
- *static int save(…)* method: the intervention scope of PML in the XEN hypervisor.
- Base line: the classical memory page tracking approach which consists in write protecting memory pages so that next writes lead to page faults.
- Metrics:
  - m_1 the execution time of save(…) method. Tells whether PML accelerates checkpointing/migration
  - m_2 the performance of the user application. Allows to check whether PML reduces or increases the negative impact of these operations on the user application.
- Observations:
  - **PML reduces the duration of save(...) method by 0.98%-10.18% (the leftmost curve in line three of the figure)**.
  - **PML slightly reduces down to 0.95% the negative impact of live migration (the rightmost curve in line three of the figure)**.



PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Effectiveness for WSS estimation

- WSS estimation system which relies on PML (in the XEN hypervisor).
- The implemented system cannot be accurate in respect with the current PML design.
- Two reasons for this issue:
  - Only modified pages are recorded. However, the WSS of a VM should include both read and write pages.
  - It is not possible to track only hot pages. In respect with the current PML design, a page is logged only once.



PAGE MODIFICATION LOGGING (PML)

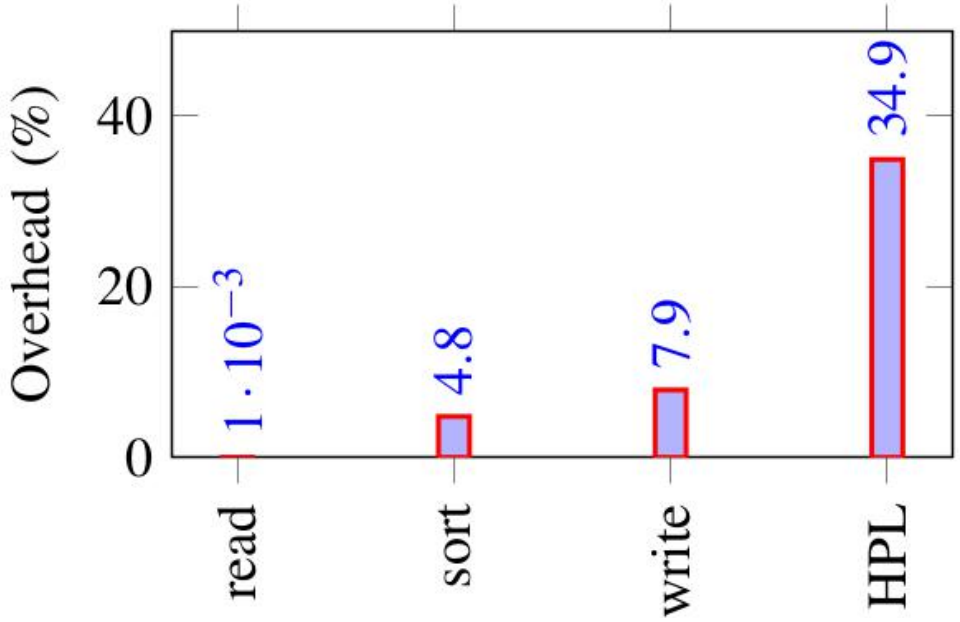
Study of Intel PML: Performance overhead

$$T = T_{PTW} + N \times (T_x + T_e + T_h):$$

- N: number of log full events. Depends on the workload type.
- T_{PTW}: overhead incurred by PML in the page table walker process.
- **T_x , T_e :** respectively the time needed for performing a VMExit, VMEnter. Are constant.
- **T_h**: time needed for performing *log full* handler. Depends on the virtualization operation for which PML is used and is very high.

PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Performance overhead



- Read intensive applications are not impacted (about 0.001% of overhead).
- For write intensive applications like HPL, the overhead is up to 34.8%.

PAGE MODIFICATION LOGGING (PML)

Study of Intel PML: Limits

The handling of PML *logging buffer full* events is done by the CPU that runs the VM.

The PML mechanism logs only modified pages.

Hot pages cannot be tracked.

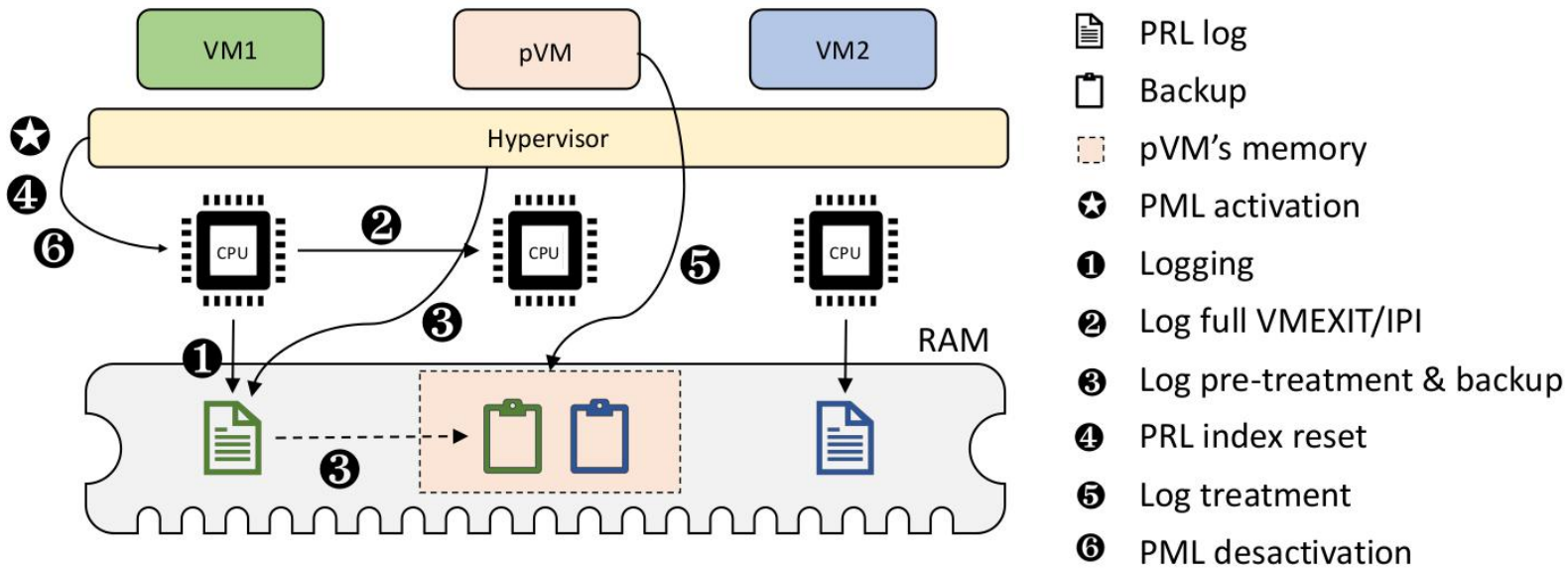
PAGE REFERENCE LOGGING

13/19

PAGE REFERENCE LOGGING (PRL)

Design implemented on Gem5

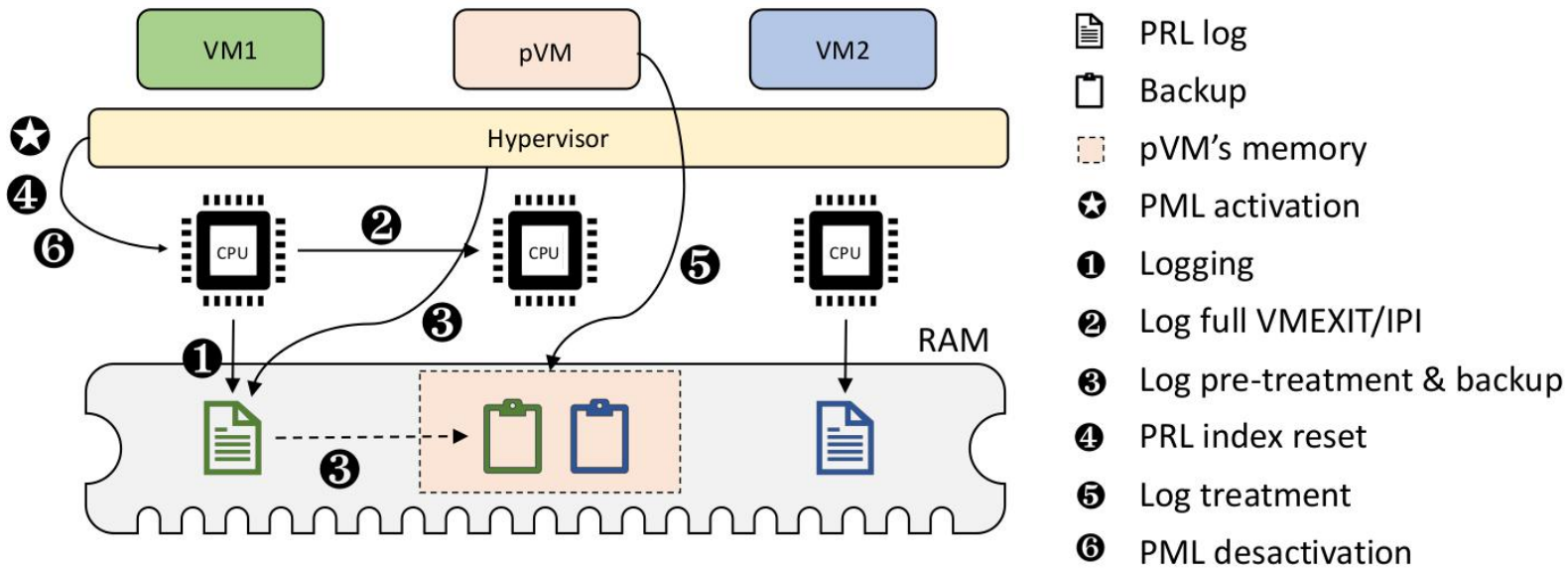
PRL is an extension of PML for making the latter usable for WSS estimation.



PAGE REFERENCE LOGGING (PRL)

Design implemented on Gem5

PRL is an extension of PML for making the latter usable for WSS estimation.



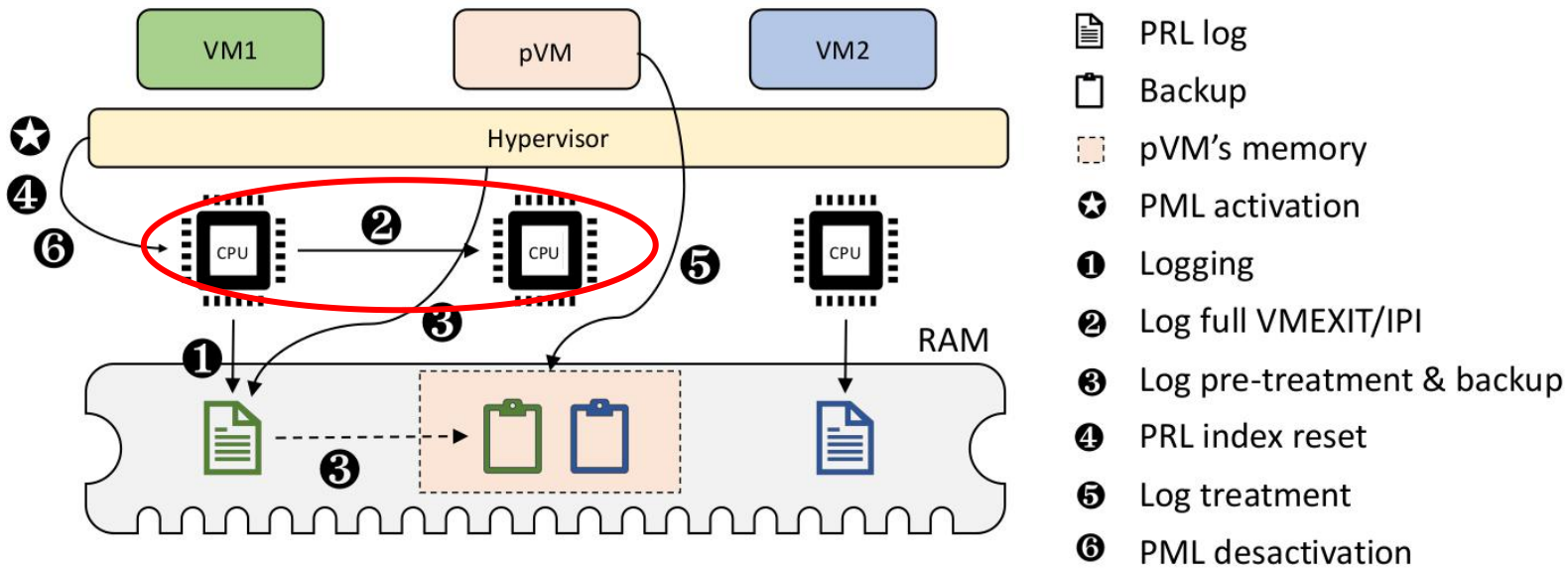
Conceptually, PRL includes two innovations:

- The capability to track both read and write pages.

PAGE REFERENCE LOGGING (PRL)

Design implemented on Gem5

PRL is an extension of PML for making the latter usable for WSS estimation.



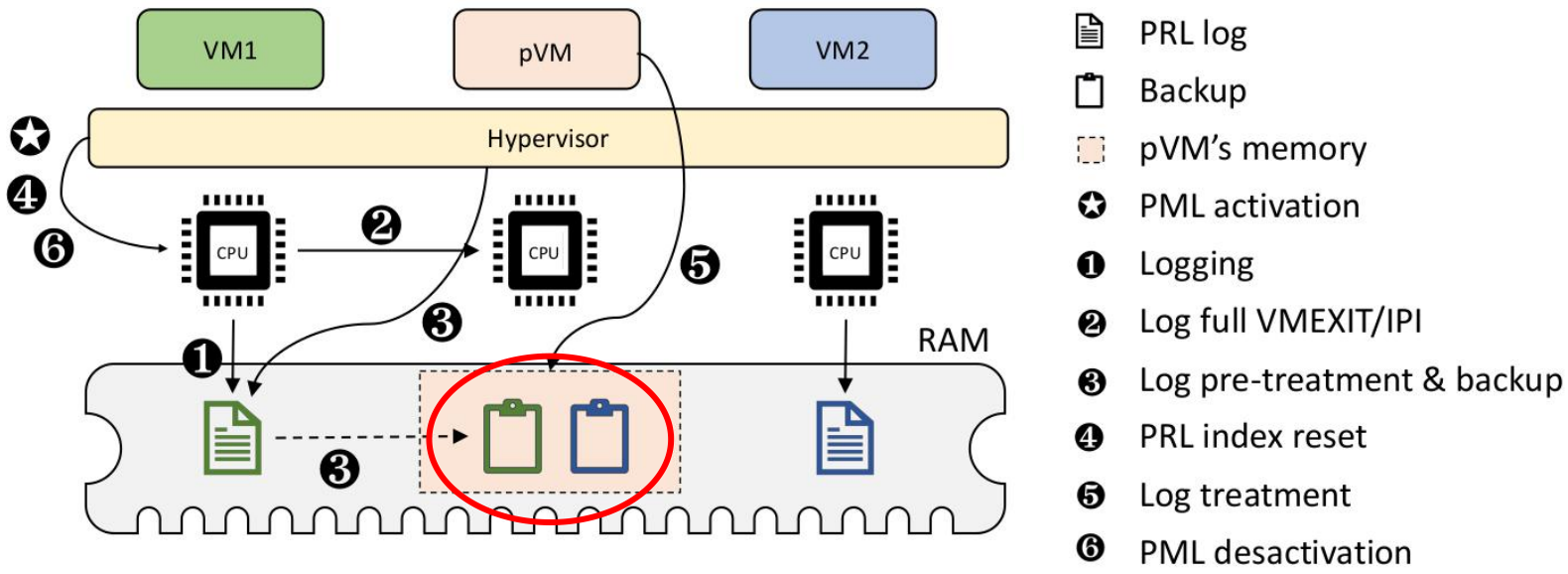
Conceptually, PRL includes two innovations:

- The capability to track both read and write pages.
- The redirection of log full events to pVM's CPUs, instead of user VMs.

PAGE REFERENCE LOGGING (PRL)

Design implemented on Gem5

PRL is an extension of PML for making the latter usable for WSS estimation.

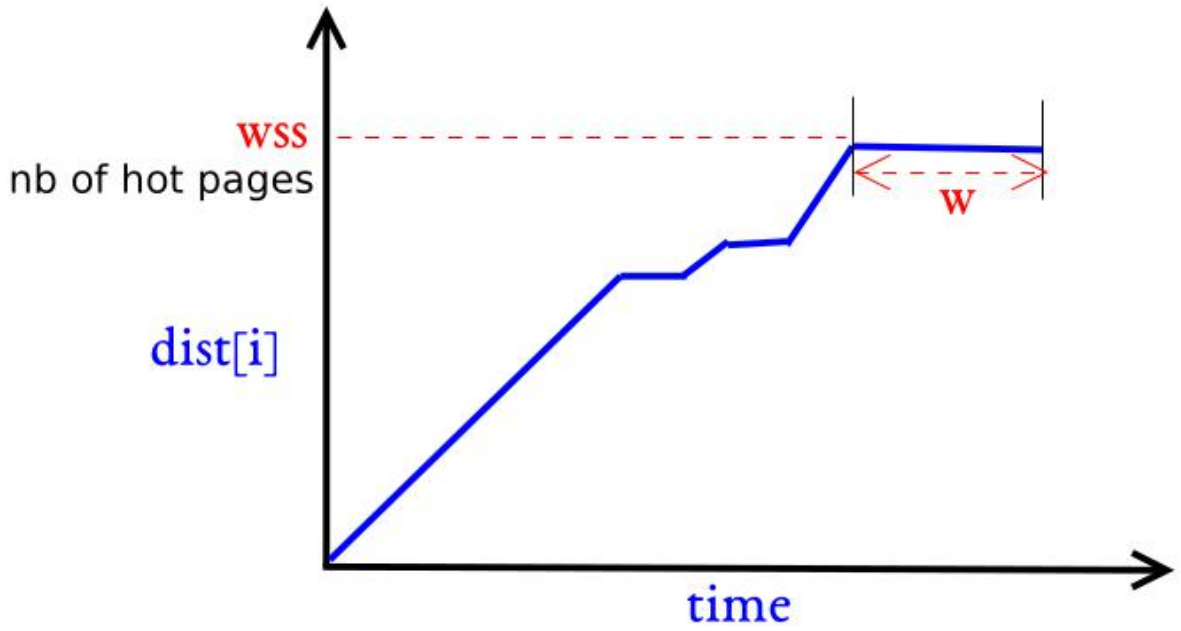


Conceptually, PRL includes two innovations:

- The capability to track both read and write pages.
- The redirection of log full events to pVM's CPUs, instead of user VMs.

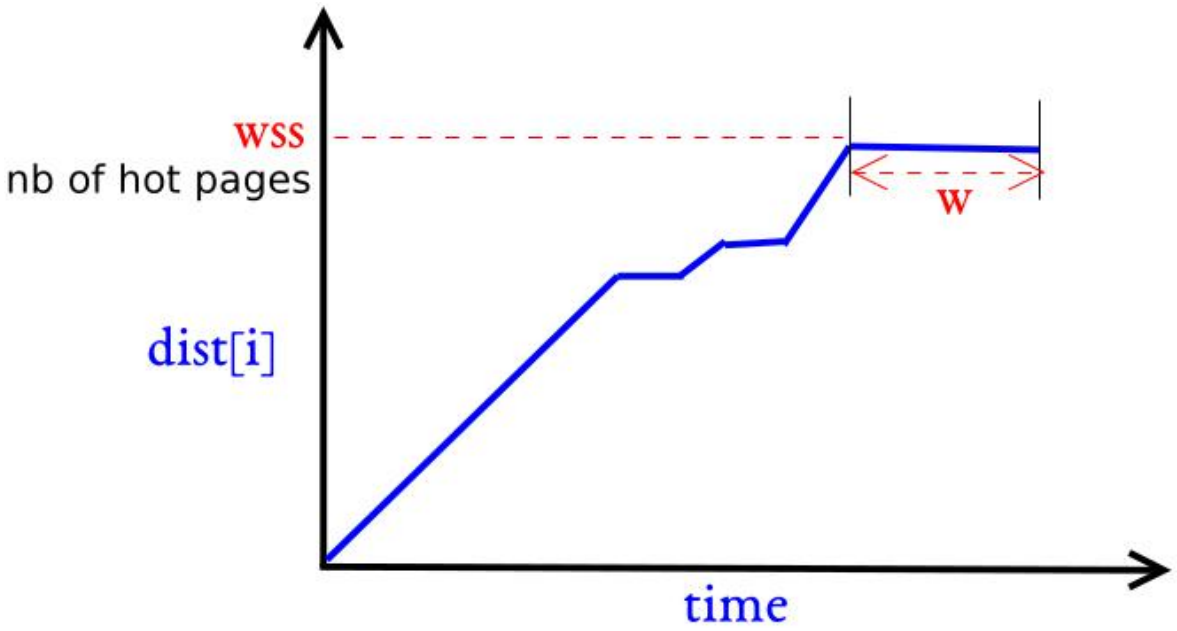
PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm



PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm

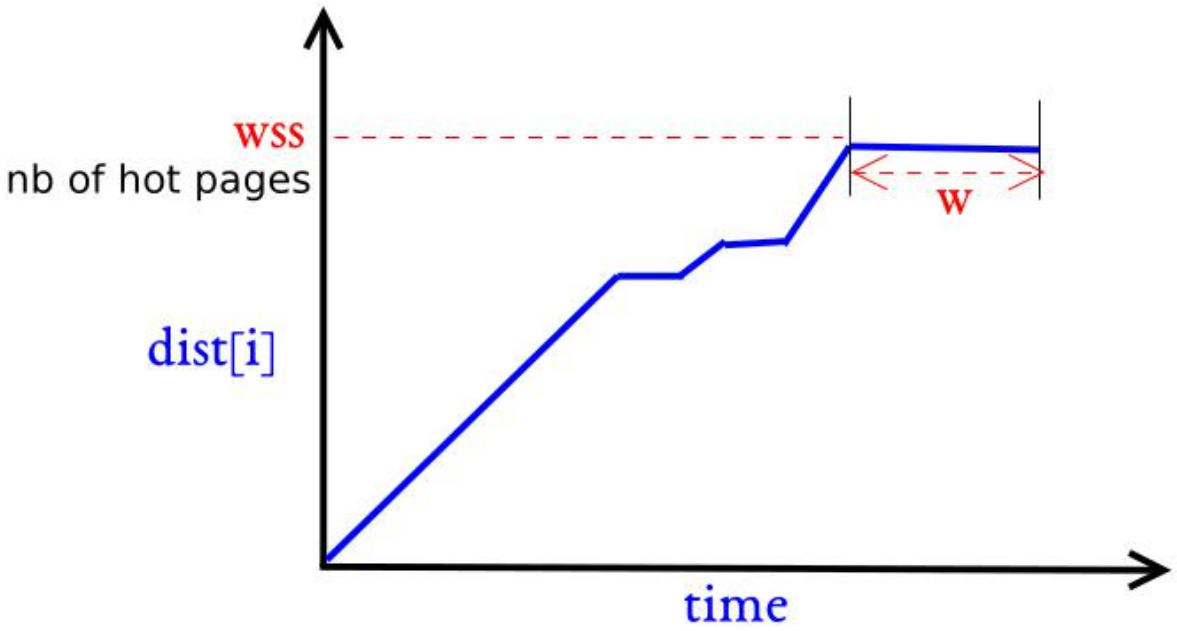


M = wss x sizeof_a_page + £

- **£:** kernel footprint.
- **sizeof_a_page:** size of a memory page.

PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm

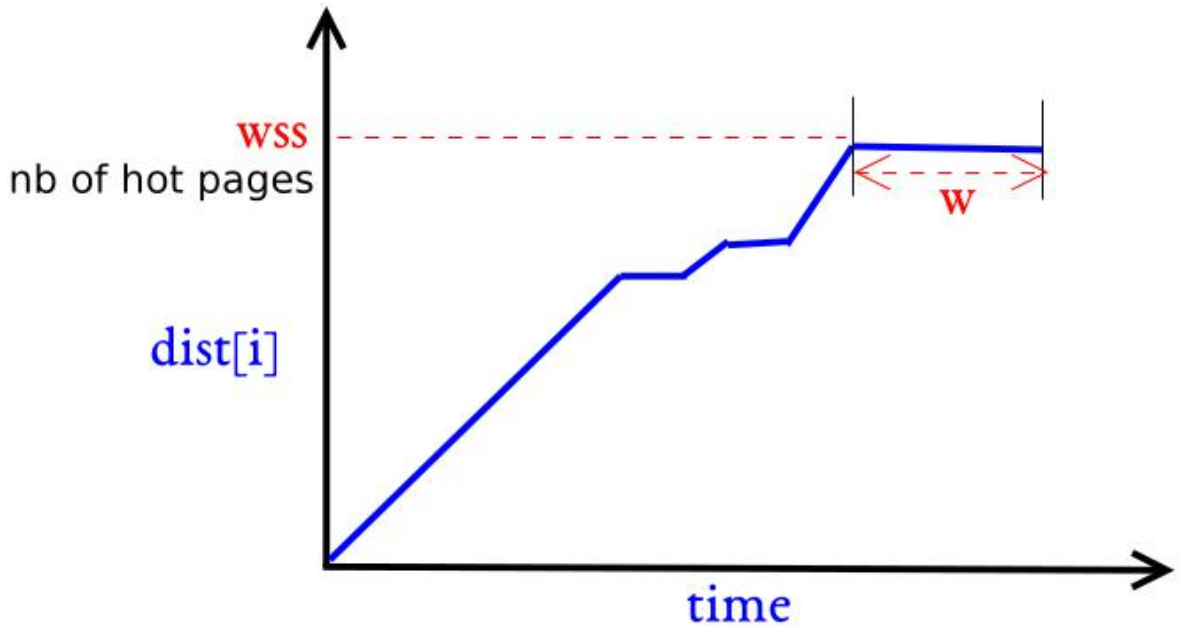


M = wss x sizeof_a_page + £

- **£:** kernel footprint.
- **sizeof_a_page:** size of a memory page.

PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm

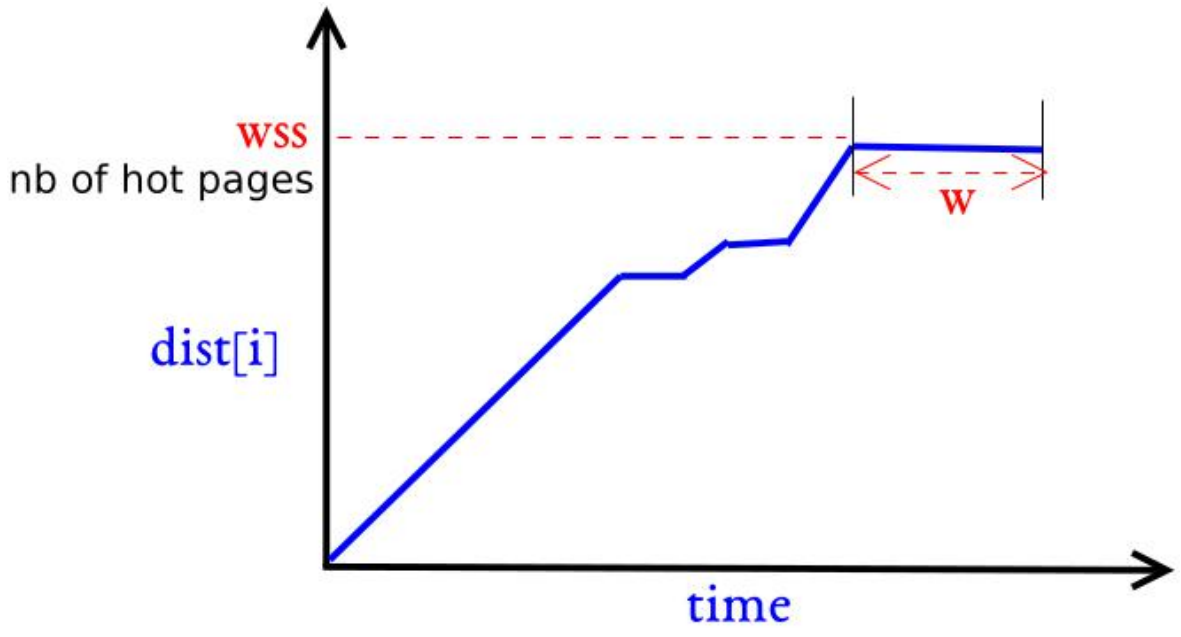


M = wss x sizeof_a_page + £

- **£:** kernel footprint.
- **sizeof_a_page:** size of a memory page.
- **wss:**
  - **τ**: treshold to consider a page as hot

PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm

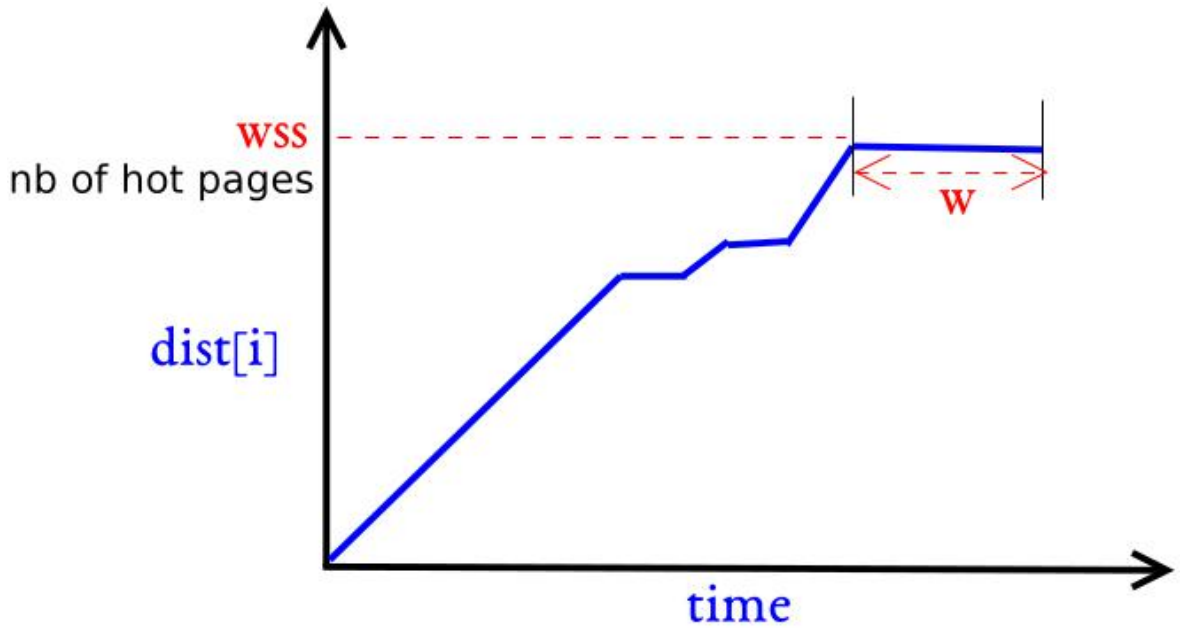


M = wss x sizeof_a_page + £

- **£:** kernel footprint.
- **sizeof_a_page:** size of a memory page.
- **wss:**
  - **τ**: treshold to consider a page as hot
  - **w**: stability duration

PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm

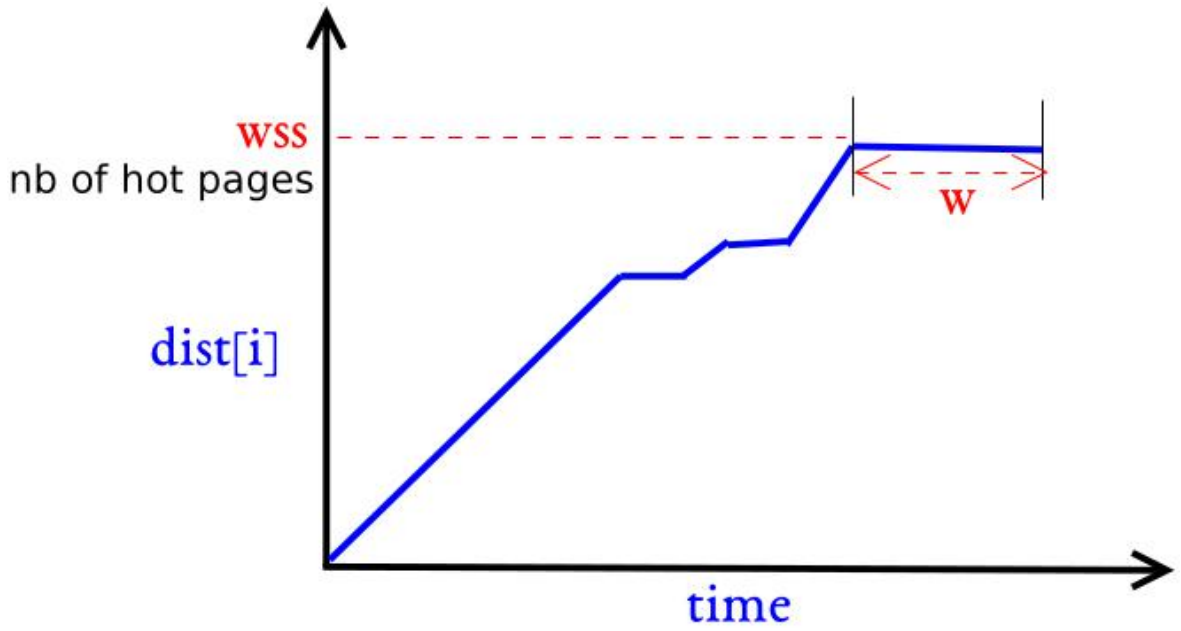


M = wss x sizeof_a_page + £

- **£:** kernel footprint.
- **sizeof_a_page:** size of a memory page.
- **wss:**
  - **τ**: treshold to consider a page as hot
  - **w**: stability duration
  - **time**: the observation interval

PAGE REFERENCE LOGGING (PRL)

PRL-based WSS estimation algorithm



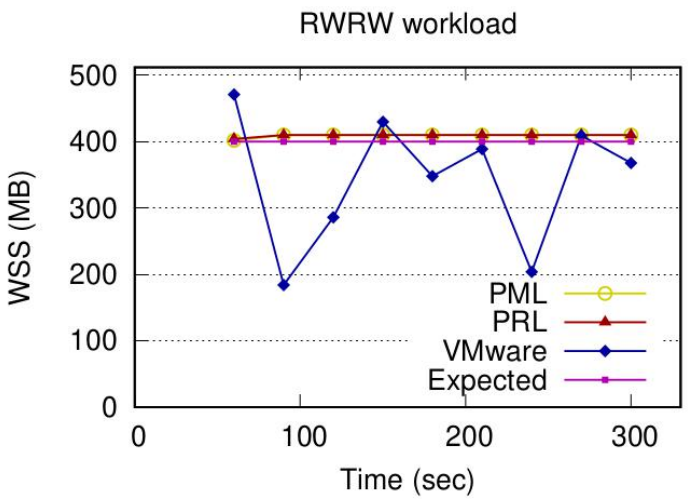
M = wss x sizeof_a_page + £

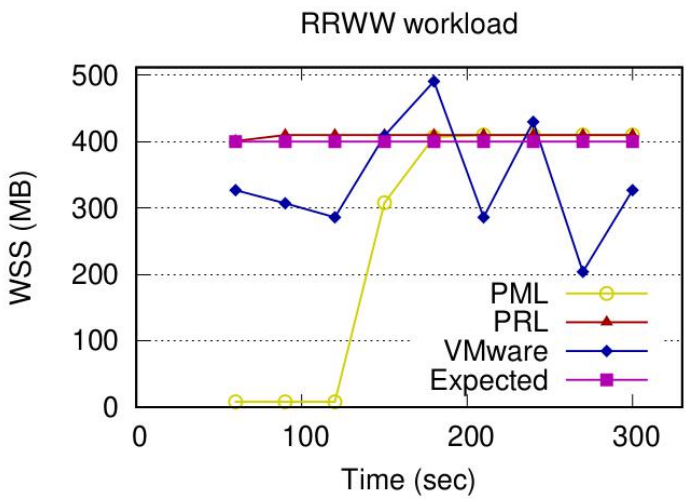
- **£:** kernel footprint.
- **sizeof_a_page:** size of a memory page.
- **wss:**
  - **τ**: treshold to consider a page as hot
  - **w**: stability duration
  - **time**: the observation interval
  - **dist[i]**: number of distincts GPAs present in the log which has been logged more than τ times

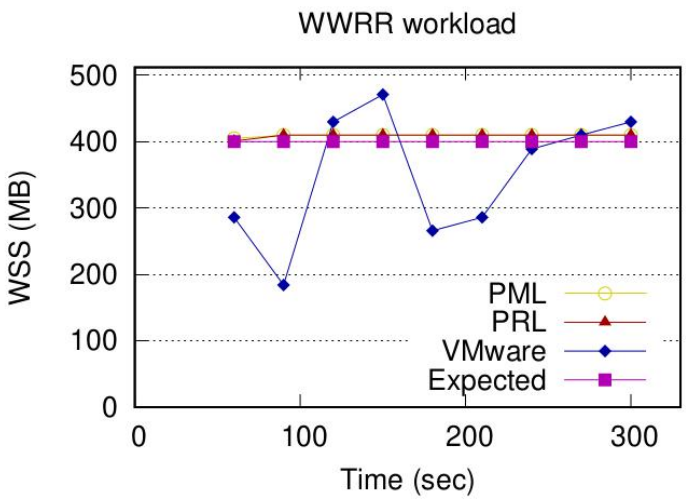
PAGE REFERENCE LOGGING (PRL)

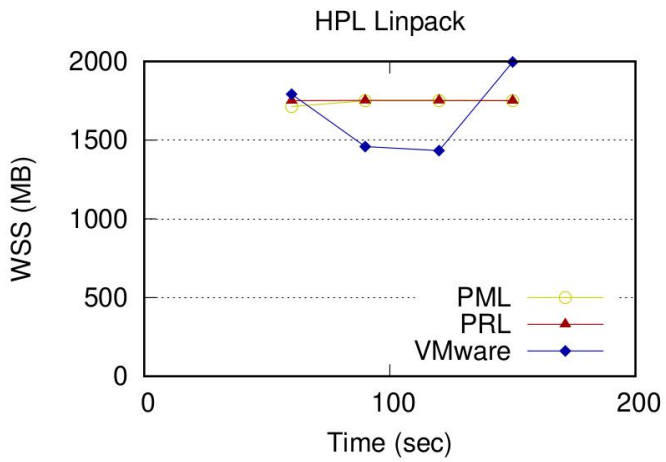
Evaluation of the accuracy

R: read W: write









CONCLUSION

CONCLUSION

- ✓ We presented a thorough analysis of Page Modification Logging (PML), a memory page tracking technology introduced by Intel and VMware as a key virtualization functionality.

CONCLUSION

- ✓ We presented a thorough analysis of Page Modification Logging (PML), a memory page tracking technology introduced by Intel and VMware as a key virtualization functionality.

- ✓ We show that although the current design of PML makes iteffective for VM live migration and checkpointing, it is notappropriate for working set estimation (WSS).

CONCLUSION

- ✓ We presented a thorough analysis of Page Modification Logging (PML), a memory page tracking technology introduced by Intel and VMware as a key virtualization functionality.
- ✓ We show that although the current design of PML makes iteffective for VM live migration and checkpointing, it is notappropriate for working set estimation (WSS).
- ✓ In the light of our analysis, we propose Page Reference Logging, an extensionof PML which makes it also effective for WSS estimation.

CONCLUSION

- ✓ We presented a thorough analysis of Page Modification Logging (PML), a memory page tracking technology introduced by Intel and VMware as a key virtualization functionality.
- ✓ We show that although the current design of PML makes iteffective for VM live migration and checkpointing, it is notappropriate for working set estimation (WSS).
- ✓ In the light of our analysis, we propose Page Reference Logging, an extensionof PML which makes it also effective for WSS estimation.
- ✓ We implemented PRL in Gem5, a popular hardware simulator and described a WSS estimation system which leverages PRL.

CONCLUSION

- ✓ We presented a thorough analysis of Page Modification Logging (PML), a memory page tracking technology introduced by Intel and VMware as a key virtualization functionality.
- ✓ We show that although the current design of PML makes iteffective for VM live migration and checkpointing, it is notappropriate for working set estimation (WSS).
- ✓ In the light of our analysis, we propose Page Reference Logging, an extensionof PML which makes it also effective for WSS estimation.
- ✓ We implemented PRL in Gem5, a popular hardware simulator and described a WSS estimation system which leverages PRL.
- ✓ We evaluated our solution using both real and synthetic applications, and compared it with VMware's WSS estimation solution. Our results demonstrate that, unlike VMware, our solution is both accurate and does not impact user VMs.

Compas'2019

HARDWARE ASSISTED VIRTUAL MACHINE PAGE TRACKING

ANY QUESTIONS?

PhD Student : **Stella Bitchebe**

Supervisors : **Pr. Alain Tchana & Pr. Laurent Réveillère**

Anglet, June 2019