Study Of Intel PML Effectiveness

Stella Bitchebe, Alain Tchana

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)

bstellaceleste@gmail.com | Alain.TCHANA@univ-cotedazur.fr

Laurent Réveillère

Laboratoire Bordelais de Recherche en Informatique (LaBRI)

laurent.reveillere@u-bordeaux.fr

<EURO/SYS'19>

PML : Page Modification Logging
WSS : Working Set Size

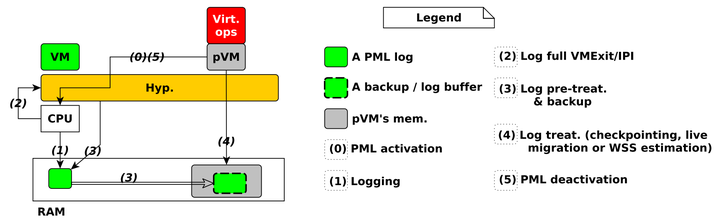
Context

- Since 2016, Intel (in collaboration with VMware) started to release processors with a novel virtualization technology feature referred to as Page Modification Logging (PML).
- We study the effectiveness of PML facing working page set analysis, checkpointing, and VM live migration.
- We validate the effectiveness of PML for live migration and checkpointing and we show that the current PML design does not allow an accurate WSS estimation and is unfair for the VM which WSS is estimated.
- We present an extension of PML called Page Reference Logging (PRL), which addresses PML limitations facing WSS estimation.

PML impact on VM Live Migration, Checkpointing and WSS estimation

- Reduction of live migration duration by $0.98\% - 10.18\%$: especially for read intensive applications.
- Reduction of the impact of live migration on users applications by $0.065\% - 0.95\%$: especially for write intensive applications.
- No improvement for checkpointing because the latter does not perform while the VM is running.
- Not efficient for estimating a VM WSS because of the several limitations included for this virtualization operation.

PML Architecture



Basic utilization of PML for improving a virtualization operation (live migration, checkpointing and WSS estimation).

PML Description

1. What is it?

- An Intel's hardware-assisted virtualization feature for monitoring a VM memory activity (Live migration, Checkpointing and WSS estimation).
- Relies on the Extended Page Table (EPT) mechanism.

2. How does it work? (See Figure on the left)

- The PML mechanism logs the guest-physical address (GPA) of every page modified by the VM (Virtual Machine) during its execution, inside a buffer.
- When the log buffer is full (512 addresses logged), the CPU that runs the VM raises a VMExit which traps inside the hypervisor.
- The handler of that VMExit does a certain task (e.g., copy the content of the PML logging buffer to a larger one), then the PML index is reset to 511 and the VM resumes.

PML Soft Limit: Unfair for cloud users and their VMs

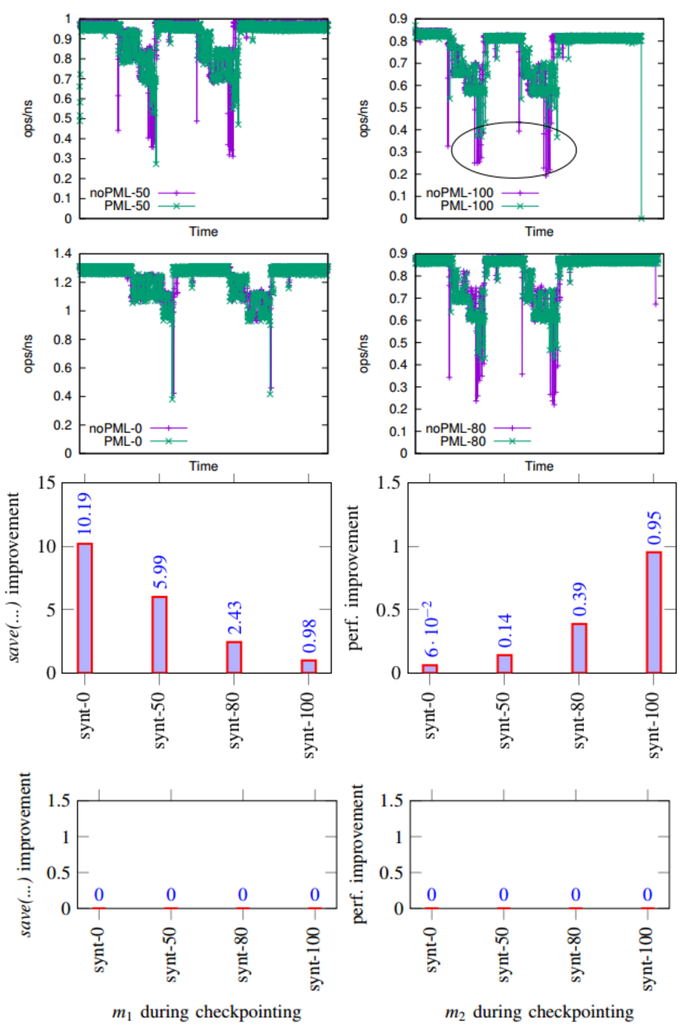
- Because of the VMExits imputed to the VM when the *PML logging buffer* is full.

PML Hard Limits: Inaccurate estimation of VM WSS

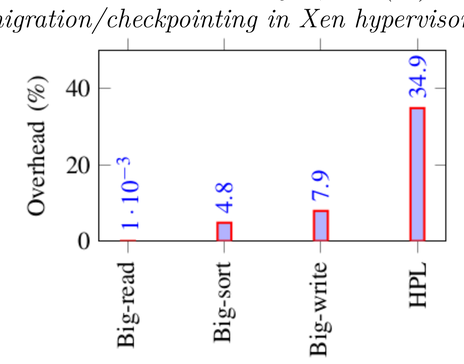
Because:

- Accessed pages are not logged
- Hot pages cannot be tracked

Experimental Results

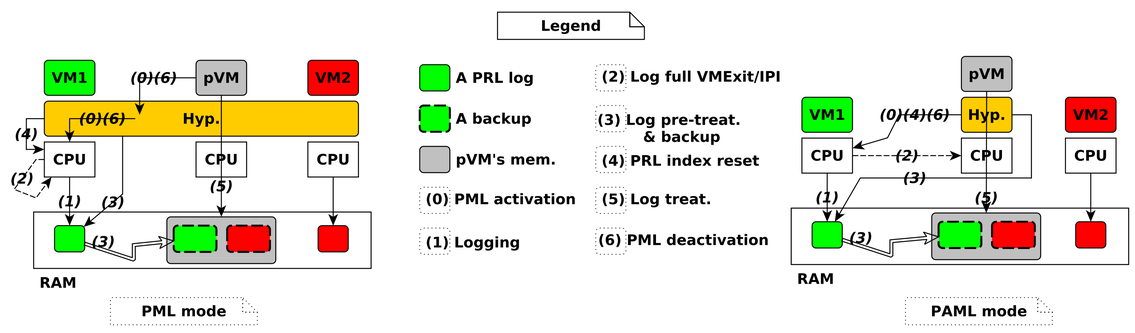


PML benefits during live migration and checkpointing. The first two lines show for the synthetic application ops/nsec (operations per nanoseconds) during live migration for different write intensity values. The third line summarizes live migration results. The last line summarizes checkpointing results (*synt-N means synthetic application with write intensity N, save(...) is the method performing migration/checkpointing in Xen hypervisor*).



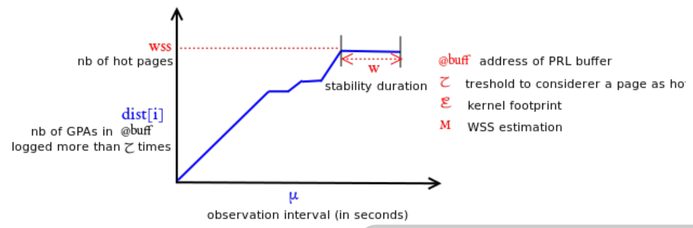
PML impact (for HPL Linpack the problem size is 29184 while for BigdataBench applications, the dataset is 10GB).

Hardware Contribution: An extension of PML, PRL (Page Reference Logging)



PRL, the new design that we propose. It works in two exclusive modes: PML (left) and PAML (right). (*PAML: Page Access and Modification Logging*).

Software Contribution: A PRL-based WSS estimation algorithm




```
For each process estimating WSS do{
    while( (dist[i] - dist[i-1]) > 0 ) do{
        sleep(μ);
    }
}
```

$$M = wssxsize_o f_{ap} age + \mathfrak{L}$$

Ongoing Work

- Implementation of PRL under Gem5 simulator.
- Evaluation of PRL and the proposed algorithm.