GDR and ASF Winter School on
Distributed Systems and Networks

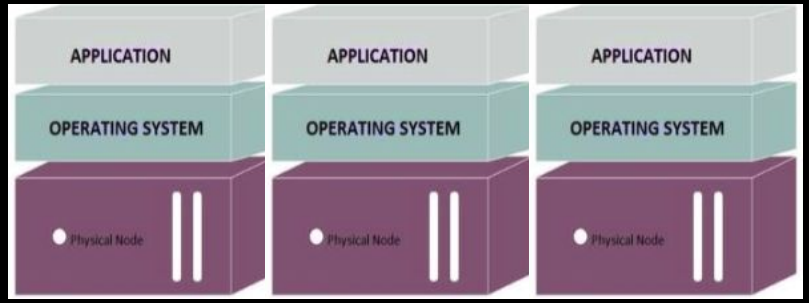
Hardware accelerators for virtualization

PhD Student - **Stella Bitchebe**
Supervisor - **Pr Alain Tchana**
Research team - **SCALE I3S**

Agenda

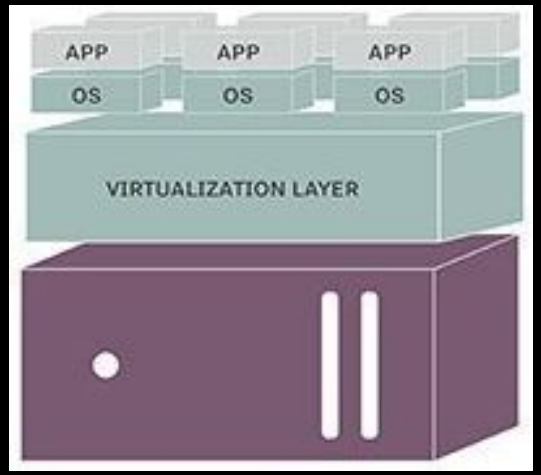
- Context
- State of the art
- Problem Statement
- Contribution
- Evaluation
- Conclusion

Context



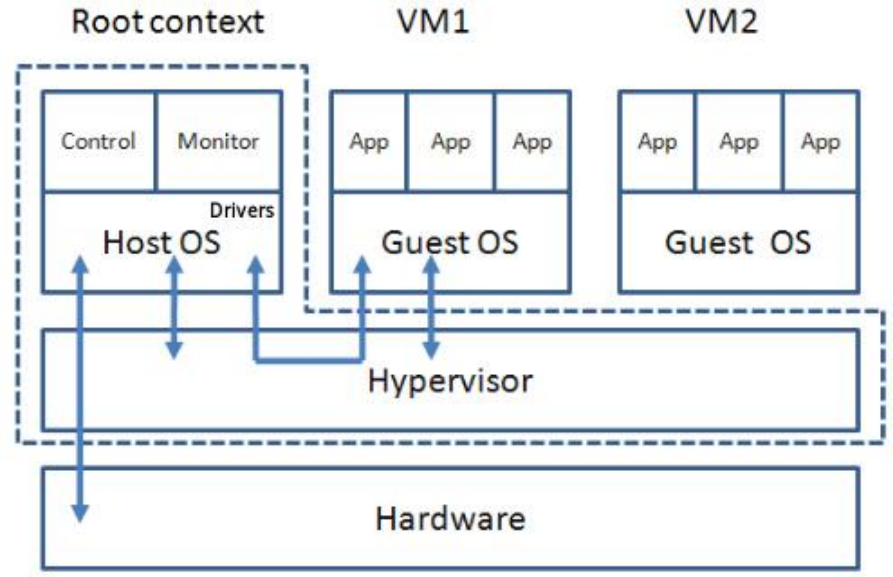
Traditionally

Virtualization



- Resources management and energy saving (vm consolidation)
- Scalability and easy deployment of applications
- Maintenance (vm migration) and fault tolerance
- etc.

Classic virtualization architecture



Root context Role

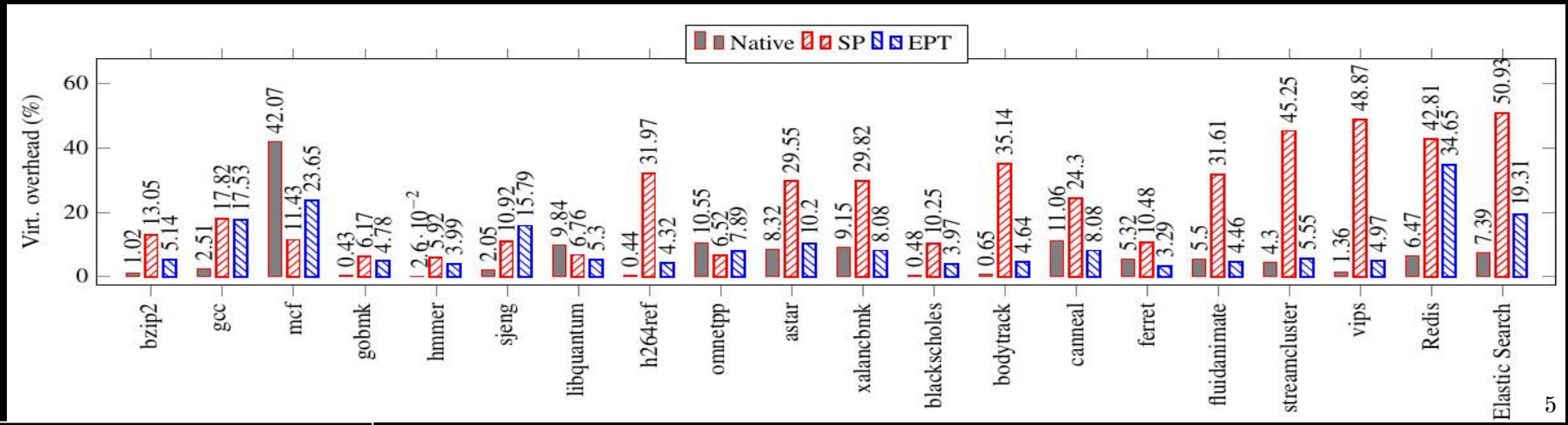
- VM management
- I/O drivers
- Monitoring tools
- etc.

Hypervisor Role

- Scheduler
- Memory allocator
- Interrupt manager
- etc.

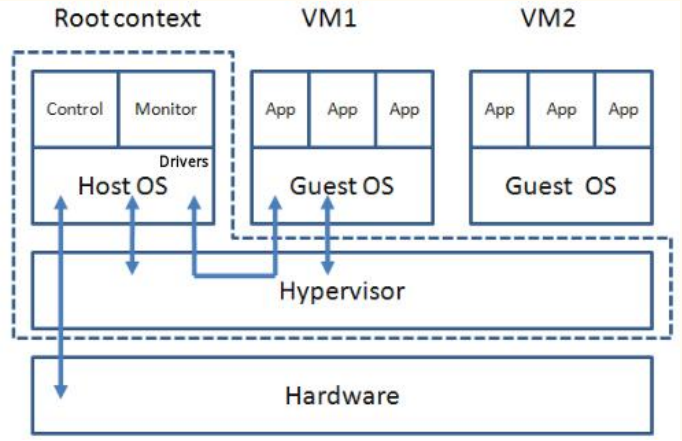
Virtualization side effects

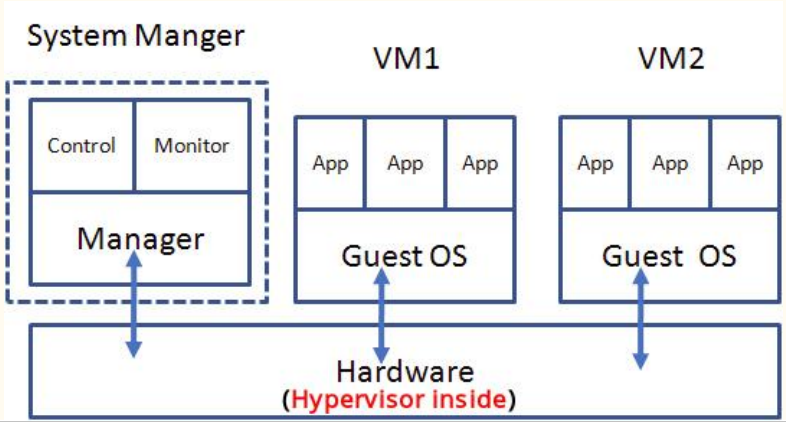
Limits of classical architecture	Promising approaches : Hardware Assisted Virtualization
• Hypervisor intervention (cpu time consumed by context switches) • Interference between VMs (pollution, e.g. cache pollution) • Hardware abstraction from the VM (VMs are black boxes) • etc.	• EPT • VT-d / VT-x • SRIOV • CAT • APICV • etc.



State of the art

Nohype (ISCA10), Nitro (AWS 2017)





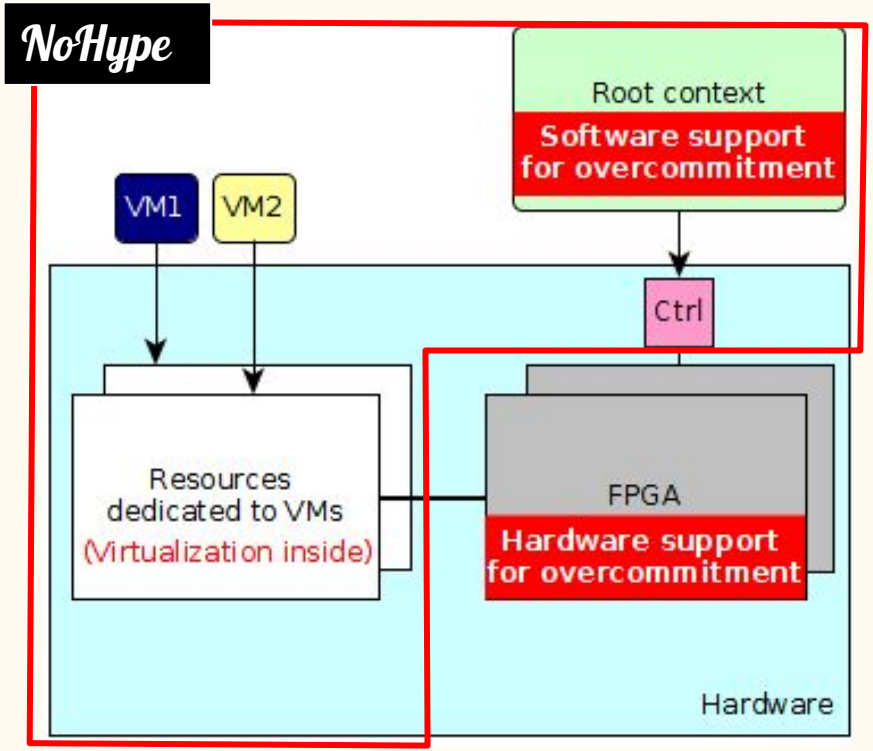
Limits of virtualization with hypervisor inside the hardware : rigidity of hypervisor

- In case of security issues (Meltdown & Spectre)
- In case of hypervisor updates (requires to change the hardware)
- Does not support Over Provisioning → PhD project
- etc.

Importance of over provisioning

- Workloads are not static
- Cloud users overestimate resources
- New workloads: Faas
- Cloud provider competition for costs reducing
- New hardware feature to facilitate overcommitment (e.g. PML)

Contribution



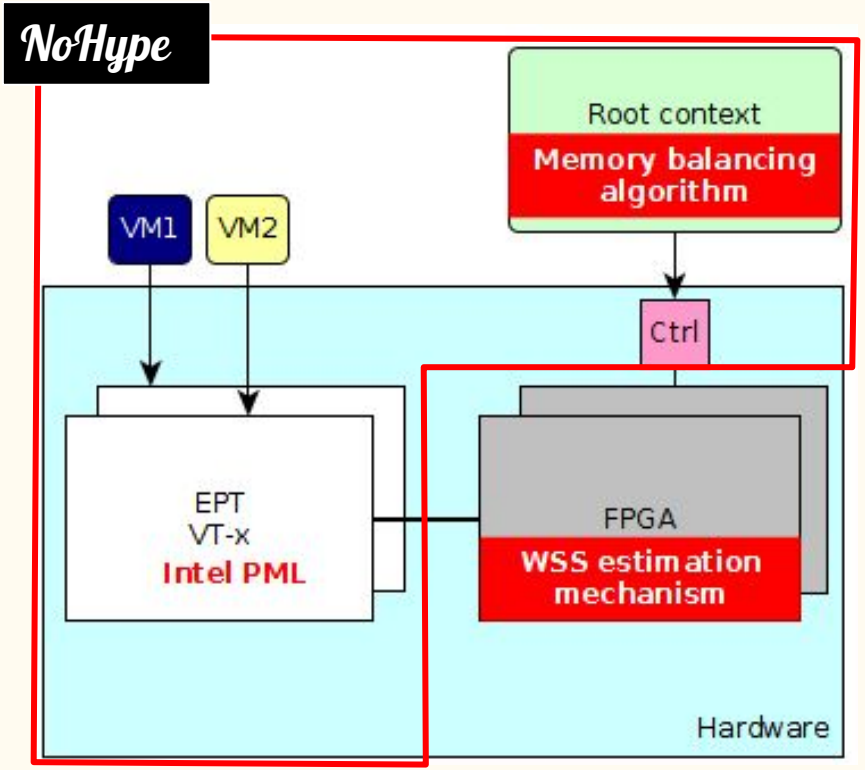
The goal of my thesis:

(Re)design hardware features and software supports (when needed) to improve overcommitment of:

- **CPU**
- **Memory (first step)**

Without impacting VMs perf.

Memory overcommitment



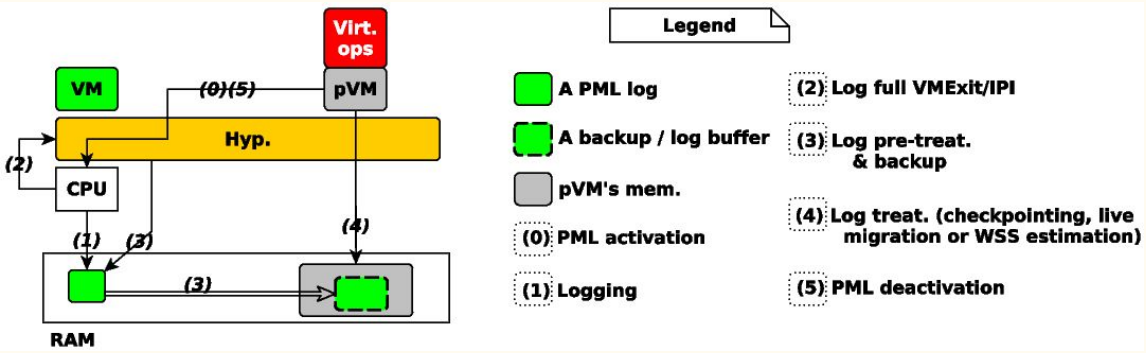
1. Working Set Size (WSS) estimation: determine the effective needs of the VMs
2. VM memory balancing

Existing methods: all software based → induce overheads

- Geiger [xxx'10]
- Exclusive cache [xxx'10]
- VMWare [xxx'10]
- etc.

Our solution: WSS estimation method based on Intel PML (hardware feature introduced in 2016 by Intel in collaboration with VMWare)

Intel Page Modification Logging (PML)



PML actual design

Description:

- Allows hypervisor to monitor the guest-physical pages modified by the VM

Aim:

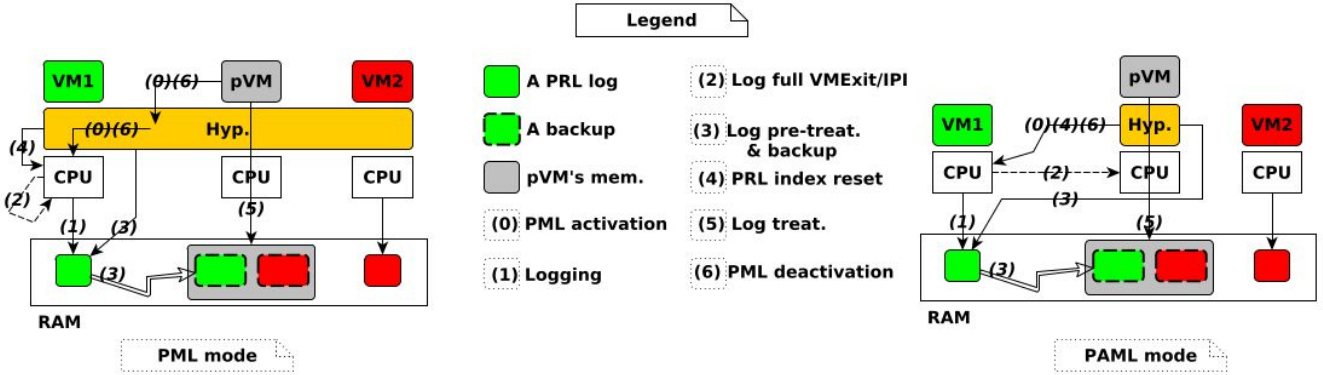
- Facilitate working set statistics during VM operations

Limits:

- Impact on VMs (VMExits)
- Overhead (CPU time used)

PRL (Page Reference Logging)

- a new PML design that we propose
- Adresses PML limits



Page Reference Logging (PRL), the new design that we propose.

Implementation and Evaluation methodology

1. Implementation and Evaluation frameworks
   a. Gem5, a computer architecture simulator
   b. FPGA boards [may be]
   c. Xen and KVM virtualization systems
2. Benchmarks
   a. SPEC benchmarks, CloudSuite, BigBench
3. Metrics
   a. Performance impact on VMs (#VMEXITs)
   b. Resource gain
   c. Energy gain (number of active servers, electricity consumption)
   d. Resource consumption (by our solution): #CPU and #memory
4. Comparison with state-of-the-art solutions
   a. see the previous slide

Questions?