

---

# Virtual Machines: Performance Optimization

## Lab. 3

# Outline

---

- Impact in performance and use of hardware assist
- Types of virtualizations with Xen

# Task1 (20%)

---

- Create an HVM installation disk domU with SPECCPU2006, SPECJBB2005 (and SPECWEB2005) installed
  - Use a “lv” of at least “15GB”
  - Set-up all the benchmarks that you plan to be used
  - Configure the domU for “console” mode
- Configure the HVM machine (and the logical volume) to support a PV boot
  - *Don't use snapshots (we will be short in disk otherwise)*
- Install and configure SpecCPU and SpecWeb
  - Choose either INT or FP
  - Choose either “support” or “banking”

# To “reuse” a disk image created in a HVM to boot a PV machine

---

- Copy hvm.cfg file to pv.cfg
  - change “kernel” and “ramdisk” by the existing Dom0 /boot
- Disk devices
  - Replace “xvda” by “hda” in “phy:”
  - Add root device to boot pv kernel (Assuming hvm installer used 1<sup>st</sup> partition)
    - root= /dev/xvda1 ro
- Fix console
  - Set extra options for pv kernel
    - extra="console=hvc0 "
  - Edit the /etc/inittab in image file and replace
    - 2:23:respawn:/sbin/getty 38400 tty2
    - By
    - 2:23:respawn:/sbin/getty 38400 hvc0
  - After that, both HVM and PV will work with a single image
  - **Remember do not boot simultaneously both VM!!**

# Task 2: CPU Bound Workloads (30%)

---

- Compare the performance of SpecCPU using:
  - Native
  - PV without VT-X
  - PV with VT-x
  - HVM-on-PV
- Where the native performance went? (in the application with the strongest effect)
  - To use perf we need to enable pmu in the domU
    - [http://events.linuxfoundation.org/sites/events/files/slides/XenPMU\\_XenSummit2013.pdf](http://events.linuxfoundation.org/sites/events/files/slides/XenPMU_XenSummit2013.pdf)
  - Experimental...

# Performance overheads analysis

---

- Unfortunately, PMU seems to be inaccessible to domU with current processors/Xen/kernels versions
  - `dmesg | grep -I pmu`
    - No driver found in kernel (?)
  - Little to no information on the webs...å
- Still the linux kernel has some interesting tracing support (and perf can access to it)
  - `perf list | grep xen`
    - xen:mc Hypercalls
      - Denoted as (mc) Multicalls due to xen “combine” multiple hypercalls in a single “multicall”
      - Key is “xen:xen\_mc\_entry”
    - xen.mmu Memory management
      - Includes shadow table, page allocation, page release, ...
      - Key are “xen:xen\_mmu\_flush\_tlb , xen:xen\_mmu\_\*\_ptpage”
    - xen.cpu Context switches
- Kernel tracing is much more powerful and complex
- Caveat: With “HVM” (even with PV drivers) the kernel don’t expose those tracepoints
  - Requires the use of “xentrace”

# Task 3: QoS for SpecJBB (20%)

---

- Configure two domU (using best virtualization) for running SpecJBB with latest Oracle JDK
  - Recompile the benchmark
- Configure CPU scheduling to achieve at least a 80% of native performance (SLA) in one of them (when executed concurrently)
  - Use remote ssh commands to synchronize execution
  - Make the configuration persistent
- (\*) If answer is “impossible”...
  - How much is the minimum performance that we can guarantee?

# Task 4: SpecWeb (30%)

---

- Compare the performance of SpecWeb2005
  - Native
  - HVM-on-PV
    - With separate domU for Clients+BeSim/Webserver
  - PV with PCI-Pass-through for the network
    - (\*) If HW supports it
  
- Multisystem run (PV-PCI-PT)
  - Configure besim and clients in a separate DomU
  - Run Dom0, and besim, clients in a single CPU (CPU0)
  - Migrate besim, client to another host (PCI-PT)