Virtualization technologies: design and implementation

Alain Tchana - (alain.tchana@grenoble-inp.fr)

krakOS - France

VM live migration

- On the board

CPU virtualization

- Consists in scheduling virtual CPUs (vCPUs) onto physical CPUs (pCPUs)
- The same philosophy as in a native system
  - thread/process scheduling

Scheduling (recall)

- Activity which consists in choosing the next process/thread/vCPU which will use the CPU
- The component which realizes this task is called *scheduler*
  - includes within the operating system or the hypervisor
- On an SMP machine, each CPU runs a scheduler which is responsible to schedule processes/threads/vCPUs assigned to the CPU
- A scheduler should take as few time as possible to execute, the CPU should run applications most of the time

Scheduling (recall)

- Linux implements two scheduler types
  - Real time, processes which priority is 0-99
  - Completely Fair Scheduler (CFS), other processes (100-140)
- By default, processes are manged by CFS

Scheduling in a Xen virtualized machine

- Two scheduling levels
  - Inside the VM (Linux schedulers)
  - Inside the hypervisor Xen: SEDF, credit (default)

Xen credit scheduler

- The goal is to proportionally share CPU times among all VMs (via their vCPUs)
- Each VM is assigned a **weight** and a **cap** value
  - weight: indicates shares (1-65535, 256 by default)
  - cap: the maximum CPU time percentage (0-x)
- work-conserving
- Implemented in `xen/common/schedule.c`

Xen credit scheduler

- Each vCPU is assigned a priority between: idle, over, under, and boost (from the less to the most important priority)
- weight and cap are translated to credits, which are assigned to the active vCPUs of the VM
- The default quantum is 30ms, which is translated into credits
  - each VM's vCPU consumed credits periodically, every 10ms.

Xen credit scheduler

- If burnedCred < 0 then priority is over
- Else, the VM priority if under
- A CPU can find VMs with under priorities on another CPU

1. Lab
2. CPU virtualization
3. Labs

Play with weight and cap

- Run a CPU intensive application and compare execution times with different weights and caps

Scheduling and Virtual Interrupts

- See 2-interrupt-injection.pdf