LIA Lab 2.1 - Virtual Machine Monitors (Hypervisors)

individual assignment

The whole point of true Type I Hypervisors[1] is to bring up a VMM farm and eventually enable virtualization-level HA and automated orchestration. VMware called that respectively vSphere HA and DRS. You will join a team in a future lab for that matter, and will be dealing with network and storage accordingly.

Task 1 - XEN vs KVM

- Install either XEN or KVM on your machine.
- Setup the network manually: a bridge so your guests can reach the SNE LAN directly.
- Install at least 4 different types of guest systems or distributions[2]. Different means of installing the target system are appreciated. *Bonus: non-Linuces, historical or experimental systems*
  - For XEN, do at least one of those as PV (*bonus: use `pvgrub` instead of loading the XEN/PV kernel from the host*) and one as HVM (*bonus: with optimal HVM drivers*).
  - For KVM, install and report on the optimal drivers for at least one guest.
- Make sure you can reach the console of some guests from the host, being hardware-assisted or para-virtualized. For the systems who allow that, what configuration**s** allows for both, the kernel and the userland system to show up there? Eventually disable the graphical console.

Task 2 - VMM Features

Install and Operate VMMs — take a pick:

- [bleeding-edge] virtualization on ARM (two students top)
- [bleeding-edge] NetBSD VMM [3]
- [hard] Recompile XEN & tools and/or dom0 and/or domU kernels[4] to enable TMEM (*hint: /lib/modules on the guest file-system*), and validate that guests now have dynamic memory allocation. *Bonus: all latests versions kernels*
- [hard] Investigate and *eventually* enable TMEM on KVM.
- [hard] Play with rkt-stage1[5]
- [medium] NetBSD as XEN dom0
- [medium] VMware ESXi on an available HP machine (max two students, will become a team later).
- [medium] Microsoft HyperV on an available HP machine (max two students, will become a team later).
- [medium?] VMware ESXi as **nested** virtualization
- [medium?] Microsoft HyperV as **nested** virtualization

For the XEN & KVM choices, investigate features regarding CPU scheduling and enable TMEM. *Bonus: test CPU-hotplug (add/remove)*

For the ESXi & HyperV choices, if you want you can also use the orchestrators or managers already. Investigate features regarding CPU scheduling and dynamic memory allocation. Install at least two of the four systems there too, and validate the optimal guest drivers there as well. *Bonus: compare guests performance against the Hypervisor from Task 1.*

At least two students need to choose ESXi and same for HyperV.

[1] https://en.wikipedia.org/wiki/Hypervisor
[2] http://sne.lan:4567/pierre/xen/help.guest (DNS: 10.1.1.5)
[3] http://www.m00nbsd.net/4e0798b7f2620c965d0dd9d6a7a2f296.html
[4] http://sne.lan:4567/pierre/xen/help.kernel
[5] https://github.com/rkt/stage1-xen