QEMU/KVM

Présentation CAPITOUL

« La virtualisation »

15 Décembre 2016 – Ludovic Pouzenc

Introduction

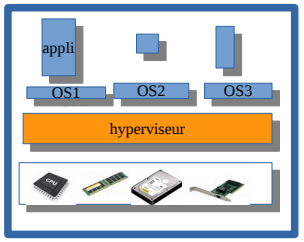
Du modèle physique au IAAS

Application
Data
Runtime
Middleware
OS
Servers
Storage
Network

Ludovic
KVM

Virtualization

Compute :
KVM
HyperV
Vmware ESXi

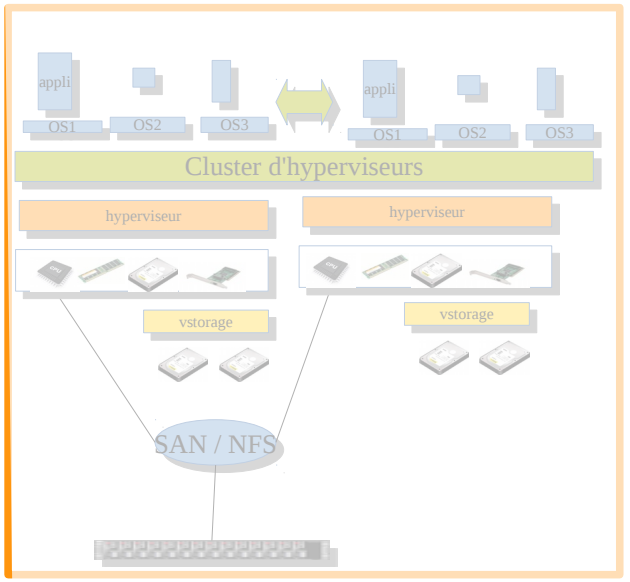


Virtualisation compute

Philippe O.
PROXMOX + CEPH

Clusters

Compute : KVM, ESXi
Storage : SAN, vSAN, CEPH
Network : vswitch (L2)

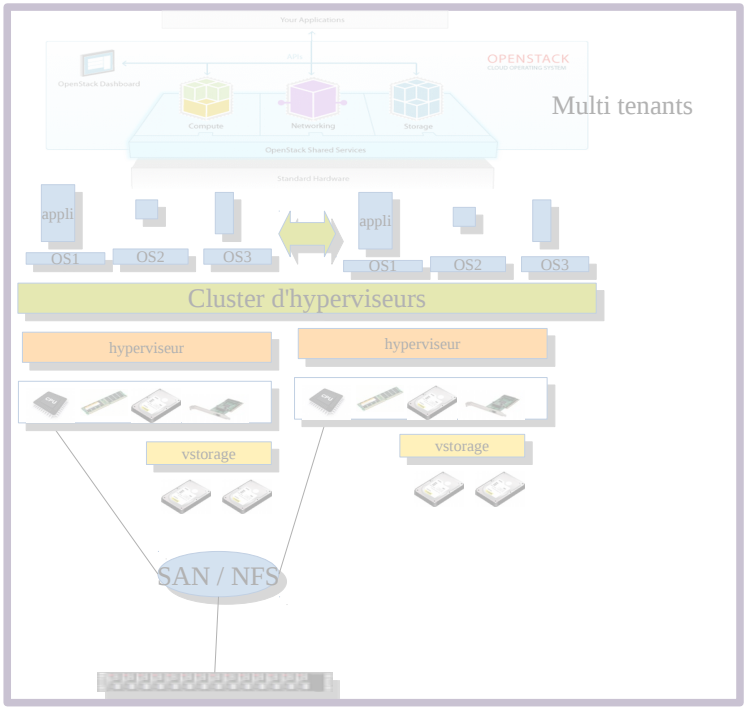


Virtualisation
compute (cluster)
± storage
± network L2
± LAN

Philippe S.
OPENSTACK + CEPH

IAAS

Compute : PROXMOX, Vmware VSPHERE
Storage : SAN, vSAN, CEPH
Network : Openvswitch (L2 + L3 + service)



Virtualisation
compute (cluster)
± storage
± network L2 + L3
⇒ contexte LAN, WAN

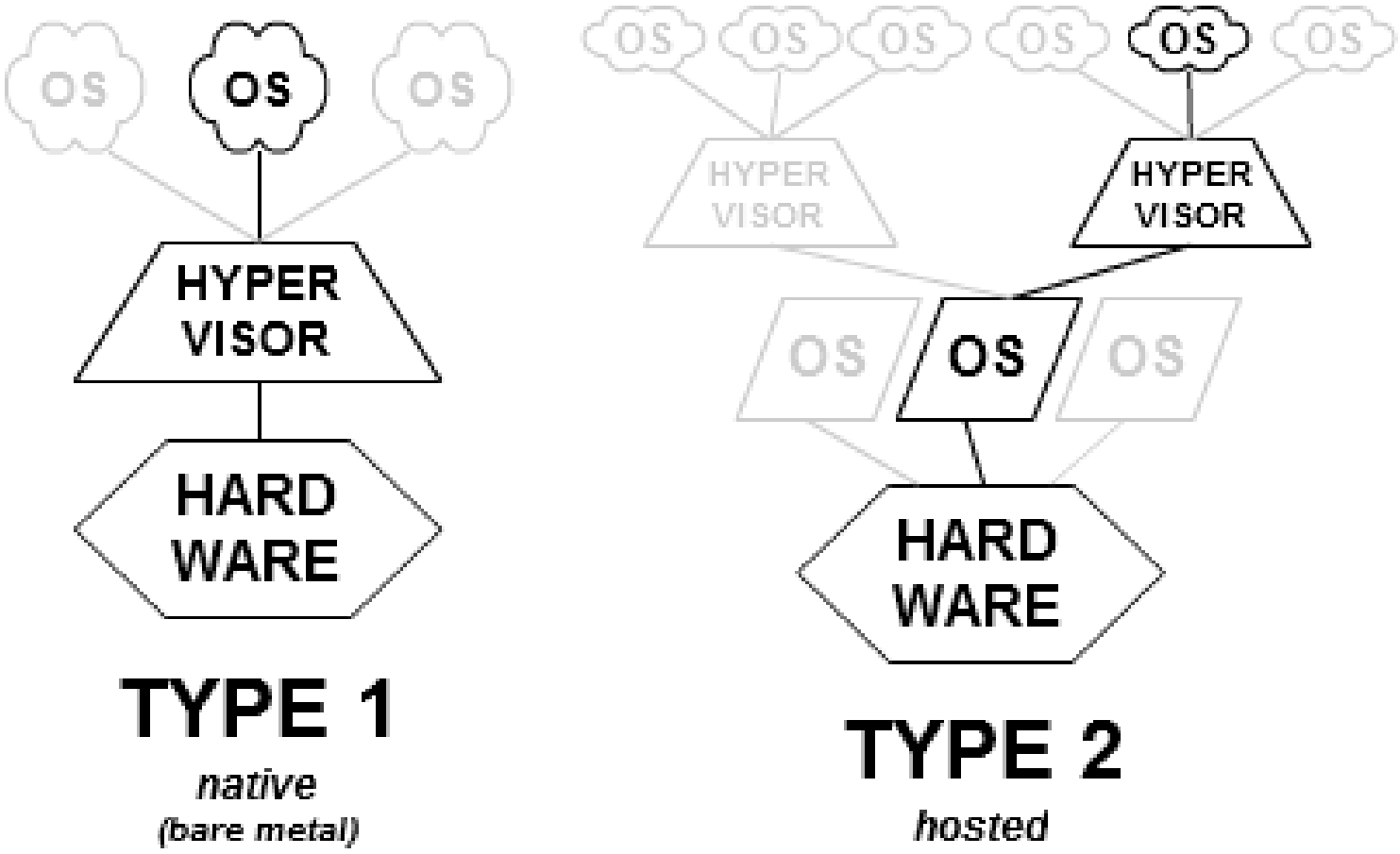
Fabrice Bellard

- http://bellard.org,
  né en 1972, Grenoble
- 1989 : LZEXE
- 2000 : FFMPEG
- ~2002 : TinyCC
- 2004 : tccboot
- 2005 : émetteur TNT avec une carte VGA
- 2007 : kqemu devient libre, qemu 0.9.0
- ~2009 : Base 4G LTE
- 2010 : record calcul Pi
- 2011 : jslinux (démo)
- 2014 : format BPG
- […] !

QEMU

- QEMU is a generic and open source machine emulator and virtualizer (http://wiki.qemu.org)
- `apt-get source qemu ; sloccount qemu*`
  - generated using David A. Wheeler's 'SLOCCount'.
  - ansic: 806 697 (94.89%)
  - kvm côté kernel ~30 000 lignes
- Targets (architectures)
  - alpha arm cris i386 lm32 m68k microblaze mips moxie openrisc ppc **riscv** s390x sh4 sparc unicore32 xtensa
- QEMU, partie virtualisation est de type 2 « hosted »

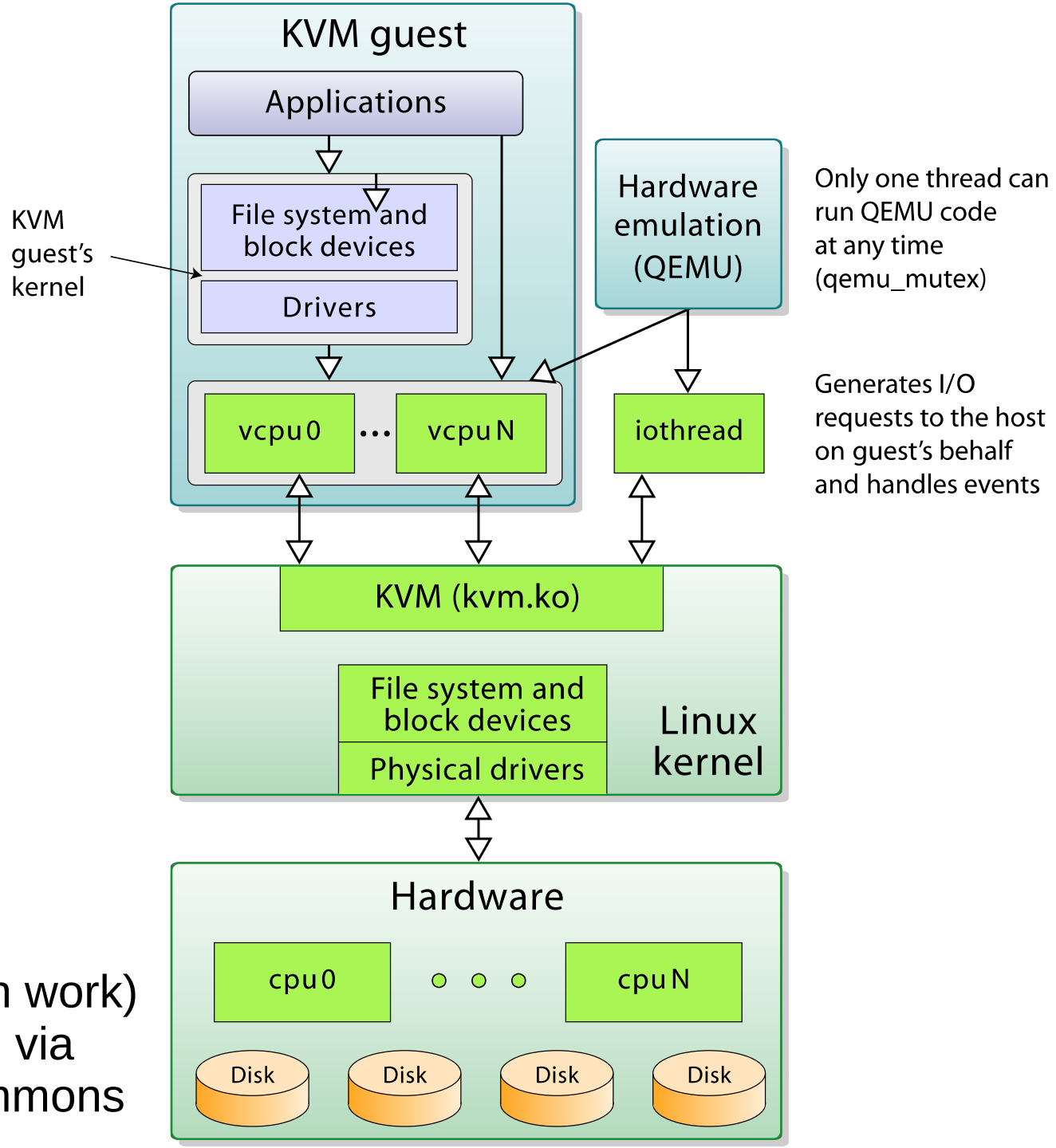
Virtualisation



– By Scsami (Own work) [CC0], via Wikimedia Commons

QEMU / KQEMU / KVM

- qemu = 1 process sur kernel standard
  - User non privilégié si souhaité
  - Accès limité aux périphériques réels (impratique)
  - Syscall, copies mémoire kernelspace / userspace
- kqemu : 1er module kernel d'accélération (F.B., obsolète)
- kvm (Kernel Virtual Machine, Qumranet puis RedHat)
  - Exec code CPU guest en mode kernel sur le host
  - Intel VT, AMD-V mais aussi équivalents ARM, PPC, MIPS...
  - qemu émule tout le reste du système



By V4711 (Own work) [CC BY-SA 4.0] via Wikimedia Commons

VIRT-IO

- OS guest vanilla => simuler des périphériques
- Périphériques réels complexes
  - Guest-side : driver e1000 linux : 11850 lignes de C
  - Host-side : simulation correcte d'une carte = difficile
  - ./qemu/debian/patches/
    - e1000-eliminate-infinite-loops-on-out-of-bounds-start-CVE-2016-1981.patch
    - e1000-avoid-infinite-loop-in-transmit-CVE-2015-6815.patch
- Pseudo-périphériques « proxy » : virt-io

Management

- Cas « minimal » : assemblage kvm / qemu / libvirt / virt-manager
- Production-ready pour :
  - Virtualisation PC individuel (à la Vbox)
  - Virtualisation infra PME 80 VM (à la XenSource)
- Peut-être aussi pour :
  - VDI petite échelle (magie SPICE, gnome-boxes)

Management

- Par rapport à ESX
  - Logiciel libre, drivers mainline linux
  - Pas d'orchestration à la VCenter
  - Migration à chaud possible, avec réserves
- Suffit déjà pour environnement SAN
  - Net : bond / bridge / vlan tagging
  - Disques : iSCSI multipath / NFS client

Démo

- Interagissez !