MINES
Albi-Carmaux

QEMU/KVM

Présentation CAPITOUL

« La virtualisation »

15 Décembre 2016 – Ludovic Pouzenc

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
- […] !

- TinyCC
  - 9x vitesse de GCC
  - Full C support, memory bound checks
  - `#!/usr/local/bin/tcc -run`
- 2005 : Emetteur TNT avec une carte VGA
  - A custom polyphase filter is used to interpolate the baseband COFDM complex signal. Then it is translated to the 25.71 MHz frequency
  - => STAR TRECK !.
- Base station 4G LTE
  - 3GPP nightmares, LTEMBMSGW
- Pi
  - 2.7 TeraDigits
  - I7, 6Gio RAM, 103 jours
  - Implém. 20x plus efficace que précédente

QEMU

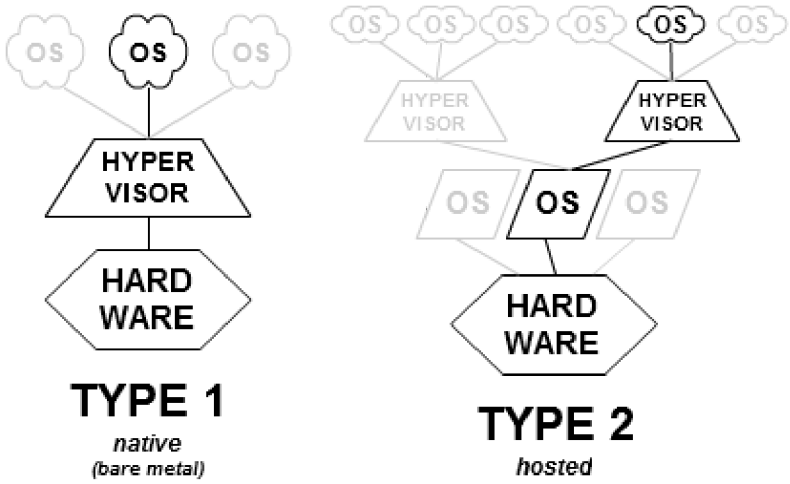
- QEMU is a generic and open source machine emulator and virtualizer (http://wiki.qemu.org)
- `apt-get source qemu ; sloccount qemu*`
  - generated using David A. Wheeler's 'SLOCCount'.
  - ansic: 806 697 (94.89%)
  - kvm côté kernel ~30 000 lignes
- Targets (architectures)
  - alpha arm cris i386 lm32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

Mode émulation, ex : VM ARM sur PC Intel (via translation dynamique)
Mode virtualisation, code guest sur CPU host

Hyperviseur = plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une même machine physique en même temps (Wikipedia)
Type 1 : natif (kernel hôte spécial hypervision, Xen...)
Type 2 : hosted (kernel hôte standard + process de virtualisation)

Qemu virtualise un **système**, pas juste un **cpu**

Virtualisation



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

QEMU = originellement type 2 « pur ».
XEN= type 1
ESX = pas clair IHMO
(process vmx, kernel custom)

QEMU/KVM n'est pas type 1 mais l'essentiel tourne en kernel mode, et pas dans un process user-space

QEMU / KQEMU / KVM

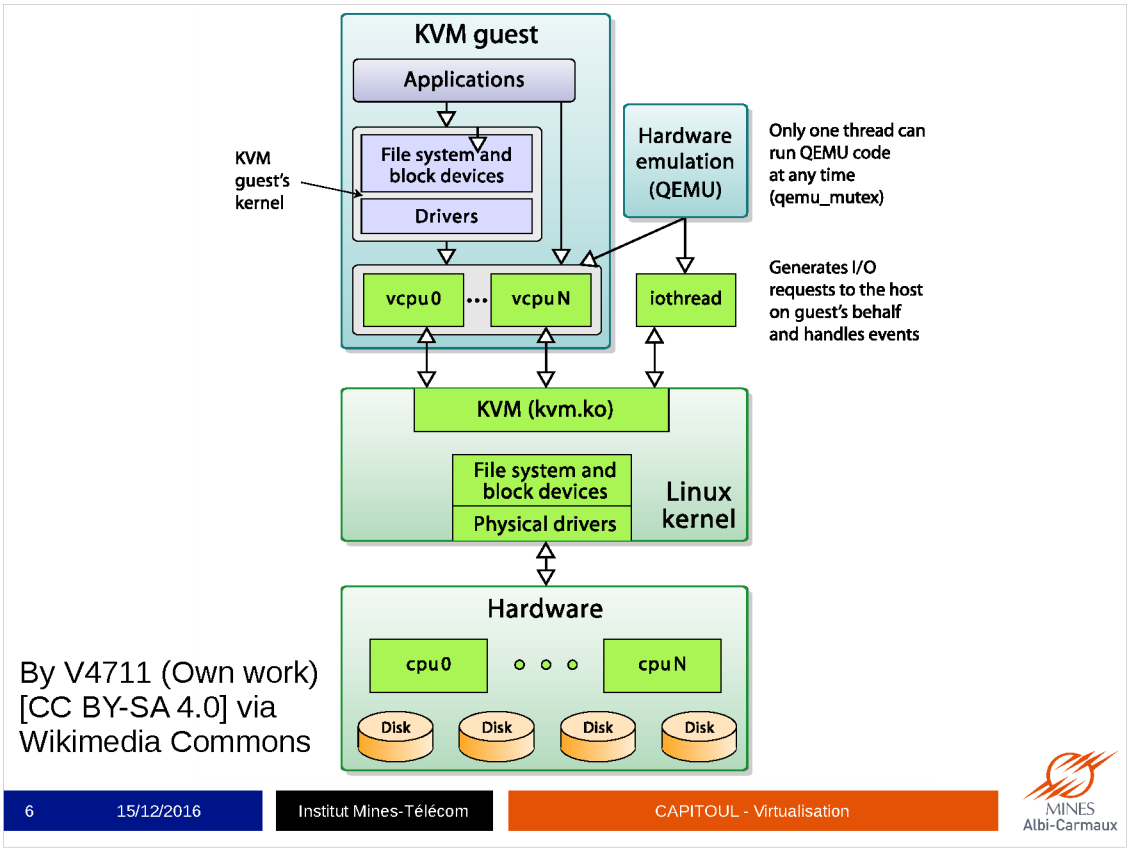
- qemu = 1 process sur kernel standard
  - User non privilégié si souhaité
  - Accès limité aux périphériques réels (impratique)
  - Syscall, copies mémoire kernelspace / userspace
- kqemu : 1er module kernel d'accélération (F.B., obsolète)
- kvm (Kernel Virtual Machine, Qumranet puis RedHat)
  - Exec 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

http://wiki.qemu.org/Documentation/KQemu

QEMU Accelerator (KQEMU) is an old driver allowing the QEMU PC emulator to run much faster when emulating a PC on an x86 host. Current versions of qemu (0.11 and up) has no support for kqemu anymore, focusing on kvm instead.

http://wiki.qemu.org/Features/KVM

KVM (Kernel Virtual Machine) is a Linux kernel module that allows a user space program to utilize the hardware virtualization features of various processors. Today, it supports recent Intel and AMD processors (x86 and x86_64), PPC 440, PPC 970, S/390, ARM (Cortex A15, AArch64), and MIPS32 processors.



Manque :

émulation des périphériques virtuels

Émulation machine

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

Pourquoi simuler des périphériques réels ?

=> Pour les systèmes où on a pas le choix sur les drivers disponibles

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 !

Virt-manager

- Hôtes locaux
- Hôtes LAN
- Hôtes via WAN+SSH

VM

- Divers OS OK
- SPICE
- Config VM à la Vbox
- virt-io, IDE…

Virsh dumpxml

Virt-manager

- New VM
- Import => existe outils conversion format

Extra : virsh commands pour monitoring, inventaire