QEMU M680x0 support

November 2020, Laurent Vivier

Who am I?

- Laurent Vivier,

  QEMU maintainer for M68K CPU emulation, Quadra 800 emulation, linux-user mode, ...

- Author of EMILE, the Macintosh Linux bootloader

  (Allow to create bootable CDROM for Mac)

- I work at Red Hat since 2015 in the Virtualization Team

Agenda

- M68K System QEMU and user-mode QEMU
- Introducing a pure M68K Virtual Machine
- Demo
- Q/A

M68K System QEMU and user-mode QEMU

QEMU and M68K

- Motorola Coldfire support since 2006 (Paul Brook)
- Motorola 680x0 since 2015
- NeXT cube, incomplete support since 2018 (GSoC 2011)
- Macintosh Quadra 800 since 2019

Why a Quadra 800?

- Easier to emulate than an Amiga or an Atari
- Built with standard component (SCSI, Serial, …) but
  - Endianness updates,
  - Pseudo DMA mode for SCSI
- Can re-use emulation parts from PowerMac emulation
- I already know how to by-pass the ROM to load the kernel
- I own a Quadra 800 so I can check with the reality

Why is Quadra 800 not enough?

- Quadra 800 emulation is limited by the real machine:
  - Limited to 1 GB (kernel, but specs say 136 MiB)
  - No PCI or other modern buses or interfaces
  - To emulate real device is CPU time consuming
  - No DMA

Why is QEMU user-mode emulation not enough?

- qemu-m68k is faster and allows to use all the CPU cores and memory of the host but:
  - All the data structures must be translated between the user side and the kernel side
  - Some syscalls cannot be emulated correctly
  - Need to be constantly updated to be in sync with the kernel
  - Apps detect the kernel configuration of the host arch.

Why a pure virtual machine is better?

- To overcome real hardware limitations (Memory, devices)
- To leverage all industry QEMU/KVM investments:
  - Virtio interfaces
  - Migration
  - To have access to more QEMU backends (encryption, RNG, vhost, GPU, …)
  - Run a m68k kernel

A pure M68K Virtual Machine

What is QEMU virt m68k machine?

- It’s a pure virtual machine with no real hardware emulation
  - A M680x0 processor (only 68040 for now)
  - 4 GiB available memory
  - 128 Virtio buses
  - Google Android Goldfish virtual devices for the basic functions (serial port, RTC, timer, PIC)
  - No Firmware

Virtio devices

- Ethernet device (virtio-net)
- Block device (virtio-blk and virtio-scsi)
- Serial port (virtio-serial and virtconsole)
- Graphic and input devices (virtio-gpu, virtio-mouse, virtio-tablet, virtio-keyboard)
- Crypto and entropy devices (virtio-crypto and virtio-rng)
- Host file system mapping (virtio-9p)

```
qemu-system-m68k -device help
```

Command line example (simple)

```
qemu-system-m68k \
   -M virt \
   -m 1G \
   -nographic \
   -serial mon:stdio \
   -device virtio-blk-device,drive=disk0 \
   -drive \
    file=system.qcow2,format=qcow2,if=none,id=disk0 \
   -kernel vmlinux \
   -append "console=ttyGF root=/dev/vda2 rw"
```

Command line example (More 1/3)

- Entropy device (for sshd):

```
-device virtio-rng-device
```

- Serial port:

```
-device virtio-serial-device \
-device virtconsole,chardev=char0
```

- Graphic display with inputs:

```
-device virtio-keyboard-device \
-device virtio-tablet-device \
-device virtio-gpu-device
```

Command line example (More 2/3)

- Shared filesystem:

```
-fsdev local,id=home0,path=/home,
                          security_model=passthrough \
-device virtio-9p-device,fsdev=home0,
                            mount_tag=home0
```

- Mount command:

```
mount -t 9p home0 /home \
      -o version=9p2000.L,
         posixacl,msize=104857600,cache=loose
```

Command line example (More 3/3)

- CDROM:

```
-device virtio-scsi-device \
-drive file=cdrom.iso,format=raw,if=none,id=cdrom0 \
-device scsi-cd,drive=cdrom0
```

- Networking:

```
-device virtio-net-device,netdev=hostnet0
```

- Networking backend:

```
-netdev user,id=hostnet0
```

or

```
-netdev bridge,id=hostnet0,br=virbr0,
        helper=/usr/libexec/qemu-bridge-helper
```

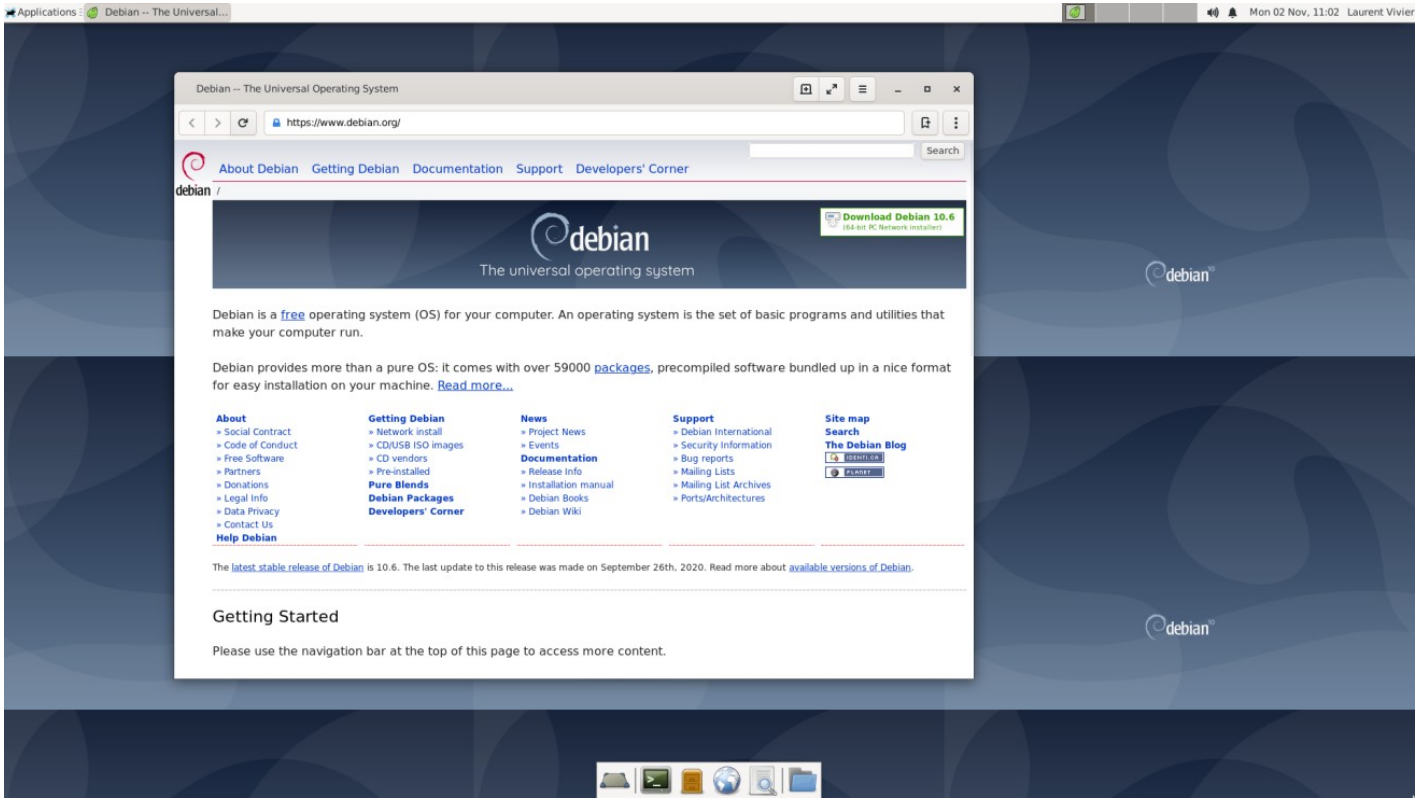
Sources (QEMU)

- QEMU:
  - `git clone git://github.com/vivier/qemu-m68k.git`
  - `git checkout m68k-virt`
  - `./configure -target-list=m68k-softmmu`
  - `make`

Sources (Linux)

- Linux:
  - `git clone git@github.com:vivier/linux.git`
  - `git checkout m68k-virt`
  - `export ARCH=m68k`
  - `export CROSS_COMPILE=m68k-linux-gnu-`
  - `make virt_defconfig`
  - `make vmlinux`

Demo


Applications
Debian -- The Universal...
Mon 02 Nov, 11:02 Laurent Vivier
Debian -- The Universal Operating System
https://www.debian.org/
Search
About Debian
Getting Debian
Documentation
Support
Developers' Corner
debian
Download Debian 10.6
(64-bit PC Network installer)
The universal operating system
Debian is a free operating system (OS) for your computer. An operating system is the set of basic programs and utilities that make your computer run.
Debian provides more than a pure OS: it comes with over 59000 packages, precompiled software bundled up in a nice format for easy installation on your machine. Read more...
About
» Social Contract
» Code of Conduct
» Free Software
» Partners
» Donations
» Legal Info
» Data Privacy
» Contact Us
Help Debian
Getting Debian
» Network install
» CD/USB ISO images
» CD vendors
» Pre-installed
Pure Blends
Debian Packages
Developers' Corner
News
» Project News
» Events
Documentation
» Release Info
» Installation manual
» Debian Books
» Debian Wiki
Support
» Debian International
» Security Information
» Bug reports
» Mailing Lists
» Mailing List Archives
» Ports/Architectures
Site map
Search
The Debian Blog
The latest stable release of Debian is 10.6. The last update to this release was made on September 26th, 2020. Read more about available versions of Debian.
Getting Started
Please use the navigation bar at the top of this page to access more content.

Questions?

Thank You

Backup

MacOS Boot