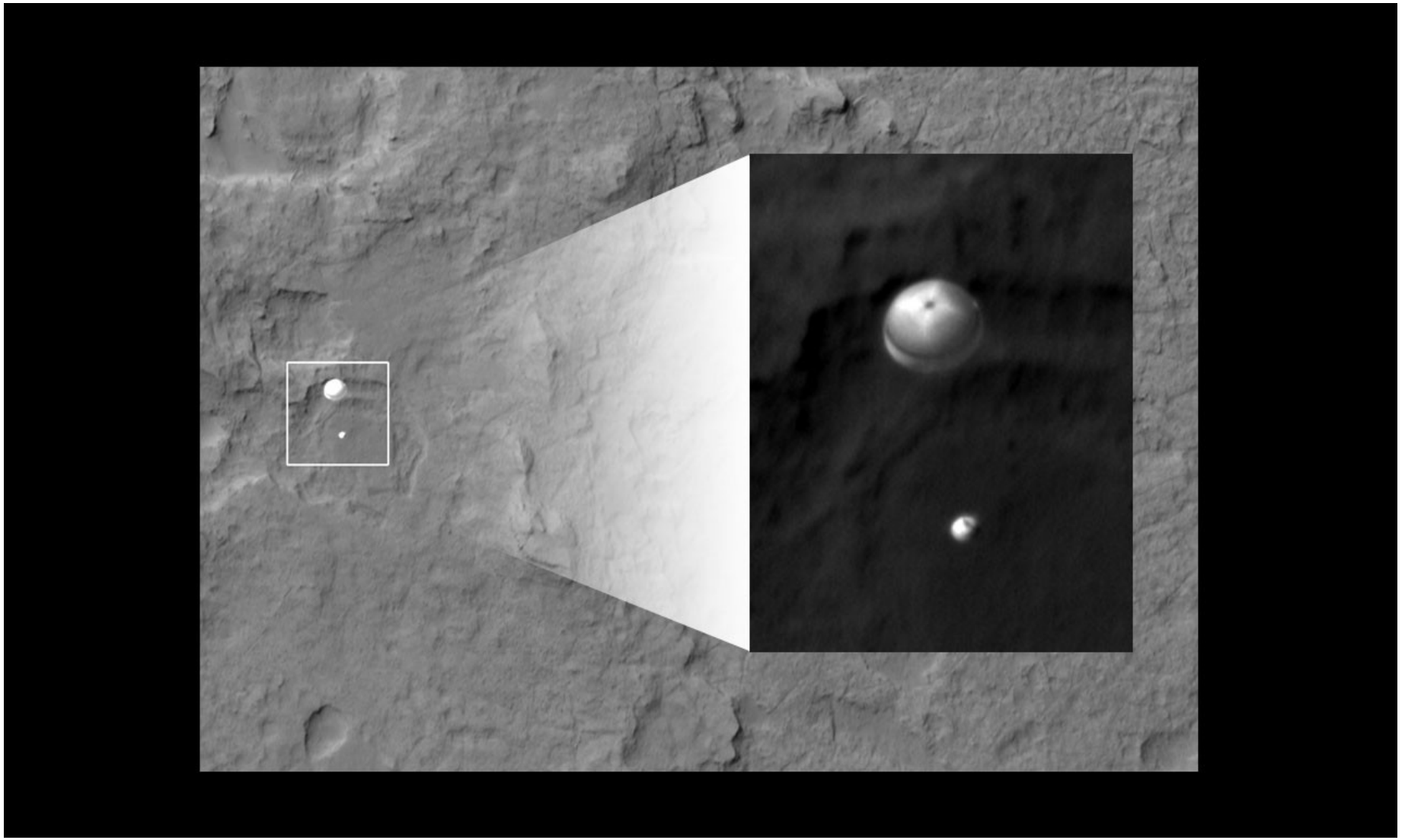
VFIO: Are we there yet?

Alex Williamson
alex.williamson@redhat.com
August 31, 2012

Almost...



Background - What is VFIO?

- **V**irtual **F**unction **I**/**O**
  - But not SR-IOV specific
- Userspace driver interface
  - KVM/Qemu VM ≈ userspace driver
- IOMMU required
- IOMMU group aware
- Full PCI interrupt support
- MMIO and I/O port access
- PCI config space access and virtualization

We talked about this last year, what's different?

- IOMMU groups
  - Allow the IOMMU driver to define device visibility
  - Unit of ownership
  - Devices accessible through groups
- Modular VFIO device and IOMMU drivers
  - vfio-pci allows PCI device access
  - vfio-iommu-type1 works with current IOMMU API and x86 style guest mapping
  - Open for other architectures and platforms to expose their specific IOMMU features and devices (POWER SPAPR support to follow soon)

Don't we already have PCI device assignment?

- KVM PCI device assignment
  - x86 only
  - KVM only
  - Doesn't understand IOMMU grouping
  - Relies on pci-sysfs
  - Not really a device driver
  - Why attach a PCI device to a hypervisor module?

“Careful. We don't want to learn from this.”
Bill Watterson (1958 -), "Calvin and Hobbes"

Where are we?

- VFIO in Linux v3.6
  - VFIO PCI driver
  - VFIO IOMMU driver
- VFIO in QEMU ~~1.2~~ 1.3

What's next?

- QEMU integration
- Legacy PCI interrupts
- libvirt support
- POWER platform support
- PowerPC?
- Error reporting
- Better page pinning
- PRI support
- Graphics support
- Migration?

Q&A, Discussion, etc...

- Where to get it:
  - VFIO kernel drivers – Linux v3.6-rc
  - QEMU - https://github.com/awilliam/qemu-vfio/tree/vfio-pci-for-qemu-1.2-v3
- How to use it:
  - Follow directions in Documentation/vfio.txt
  - Bind devices to vfio-pci to create viable groups
  - -device vfio-pci,host=1:10.0,id=hostdev0

Thanks

Differences from UIO

- Requires IOMMU with isolation support
  - Not available on bulk of platforms where UIO is used
  - Enables DMA access rather than PIO
- Provides access to all I/O spaces
  - Gated by above IOMMU requirement
  - Virtualized where necessary for security & convenience
- Flexible interrupt support
  - Makes use of eventfd/irqfd models
  - IOMMU protection required for MSI

IOMMU grouping examples