FTF | FREESCALE TECHNOLOGY FORUM POWERING INNOVATION

Virtualization Deep Dive on Freescale QorIQ Platforms

FTF-NET-F0597

Varun Sethi
Software Architect

Sudhanshu Mittal
Software Manager

August 2012

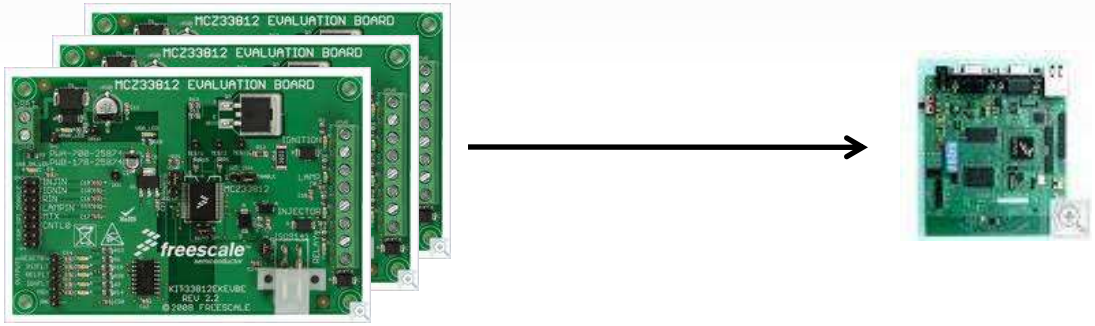
Agenda

- **Introduction to Partitioning and Virtualization**
- **Overview of Topaz**
- **Overview of KVM**
- **Performance Considerations**

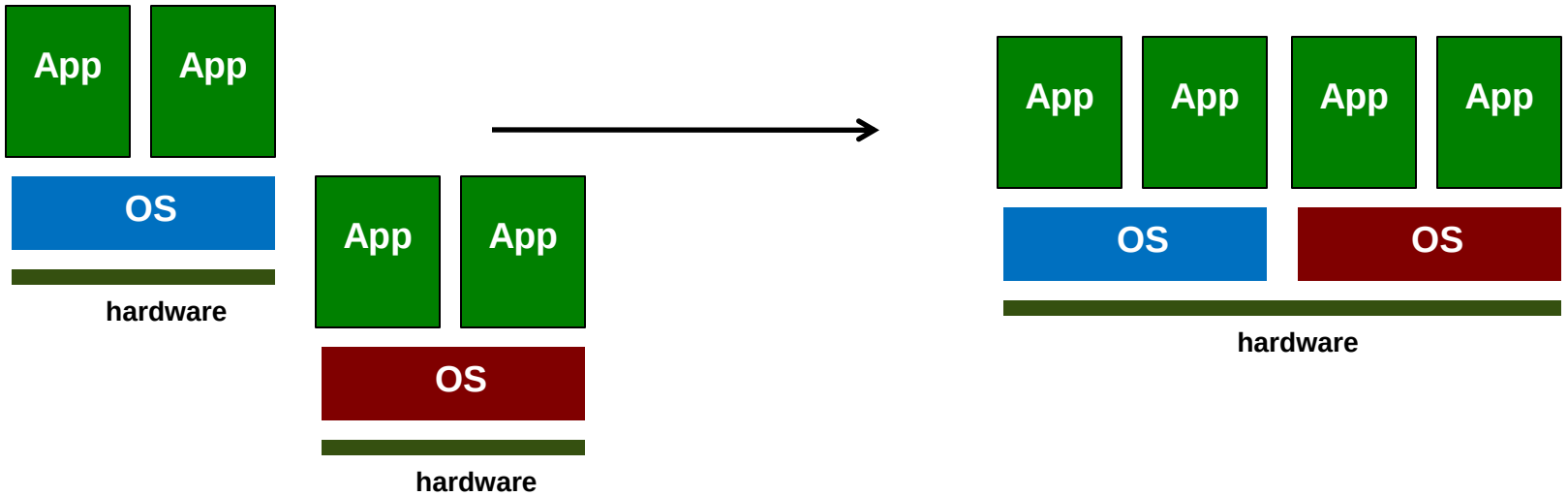
FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

Introduction to Partitioning and Virtualization

Consolidation on Multicore Processors

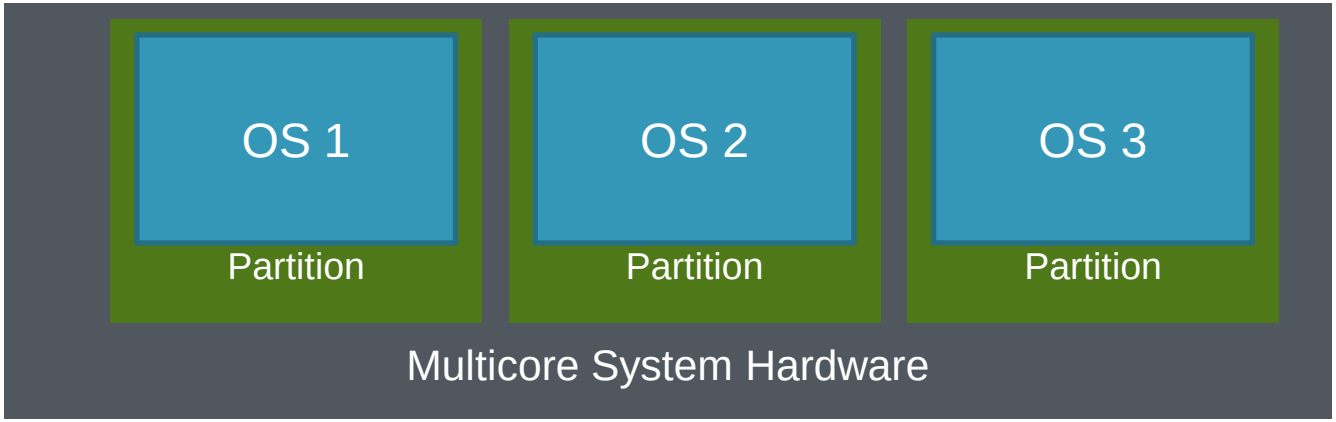


Benefit: Cost/power savings



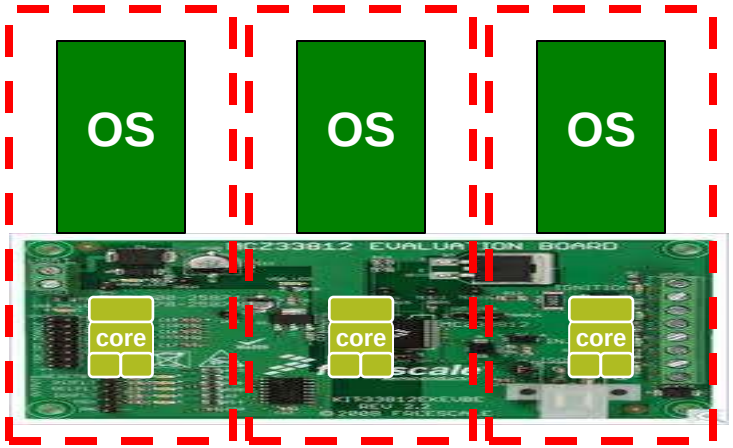
Partitions

- Enables consolidation
  - Multiple operating systems/partitions on a multicore chip
- Enables Secure operation of multiple Operating Systems
  - Isolation mechanisms are needed for safety, robustness



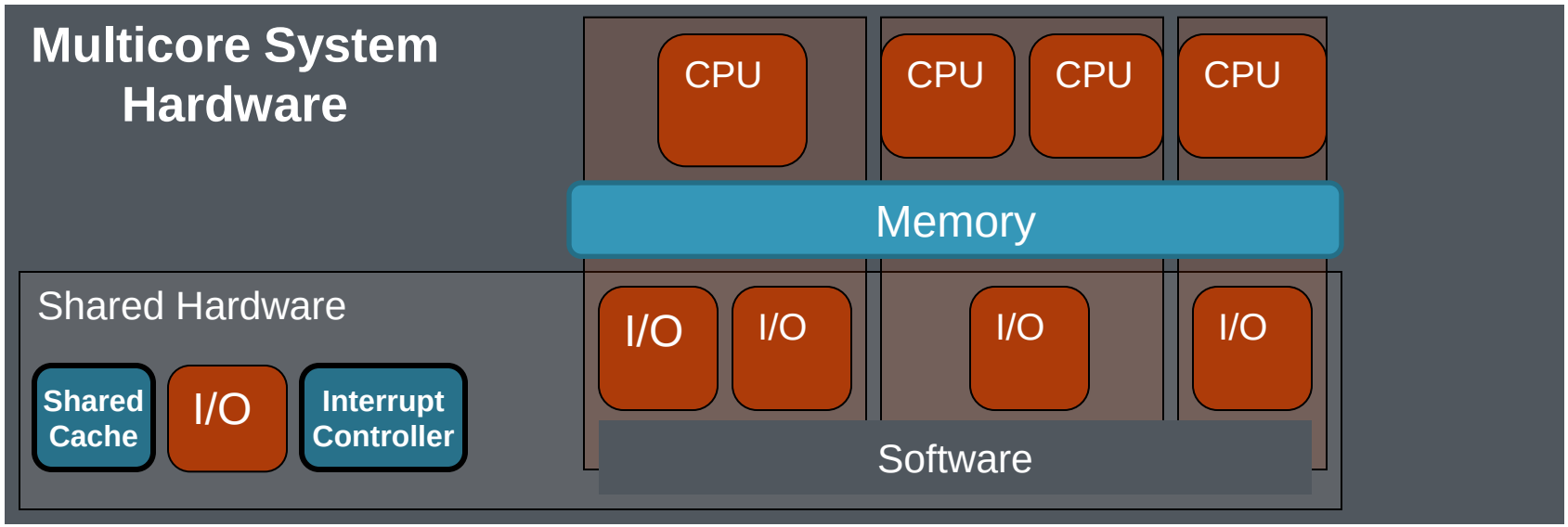
Unsupervised AMP (asymmetric multiprocessing)

- Security — no enforced isolation, cannot allow untrusted operating systems
- Requires cooperation among partitions
- How are global hardware resources managed?
  - Local access windows
  - Interrupt controller
  - Shared caches
  - IOMMU
- Boot sequence complexity
- Error management
- Resetting/rebooting partitions
- Debugging



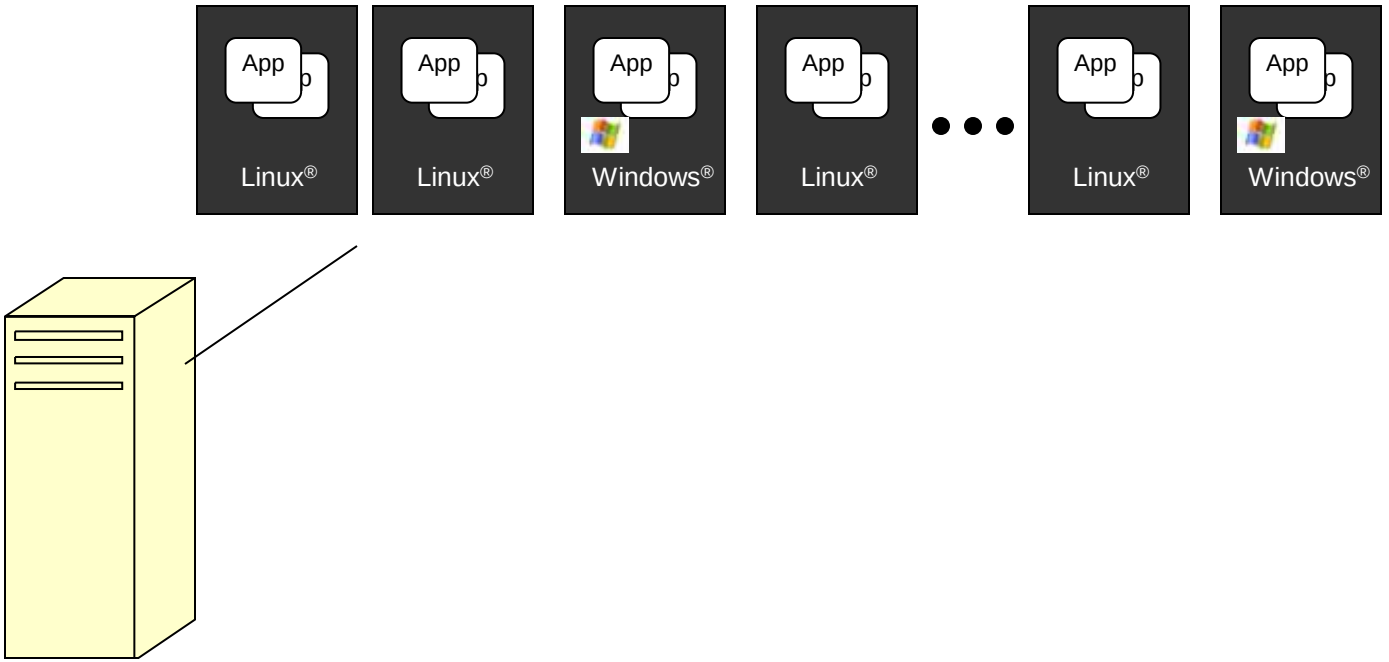
Enforcement of Partitioning

- Enforcement of separation can be done robustly with adequate hardware support.
- Partitions are enforced and managed by system software
  - Often a hypervisor

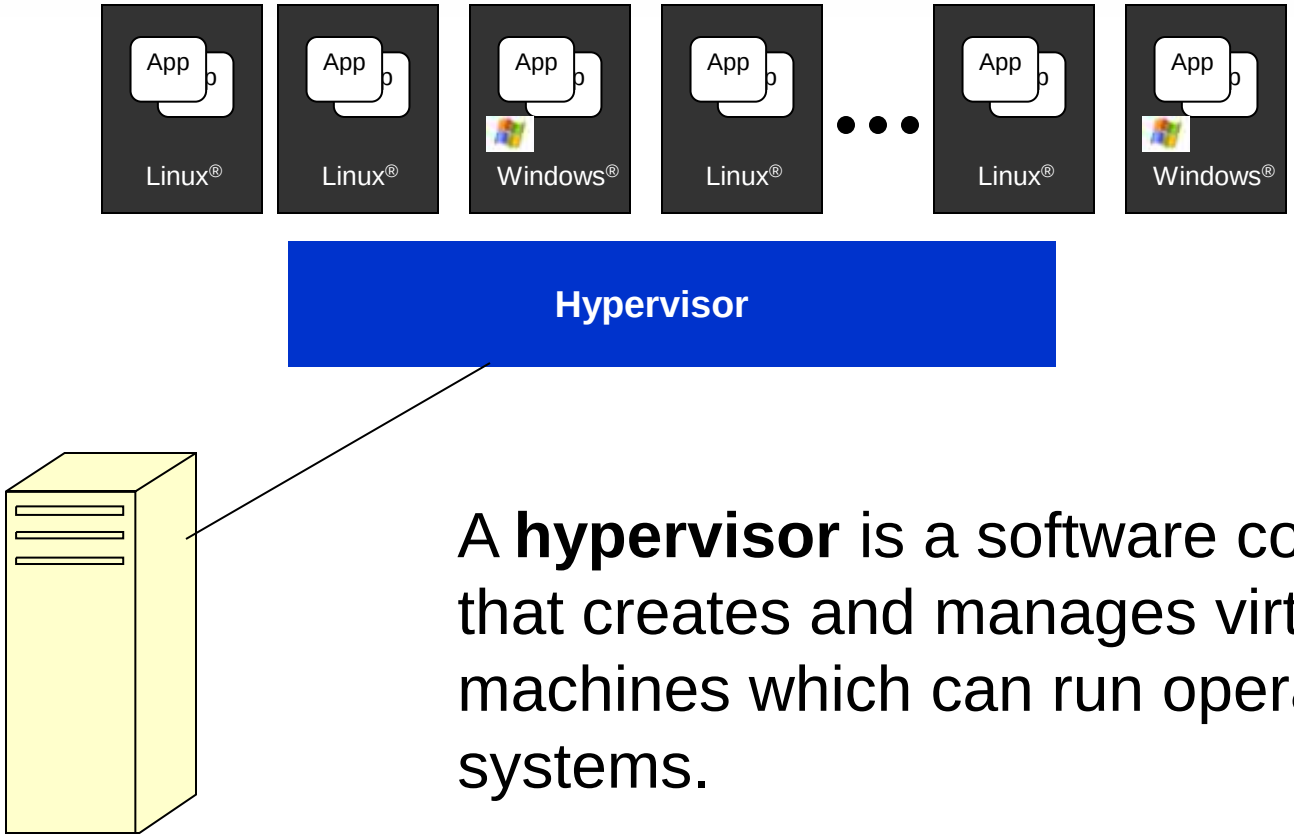


What is Virtualization?

Virtualization – Hardware and software technologies that provide an abstraction layer that enables running multiple operating systems on a single computer system



What is a hypervisor?

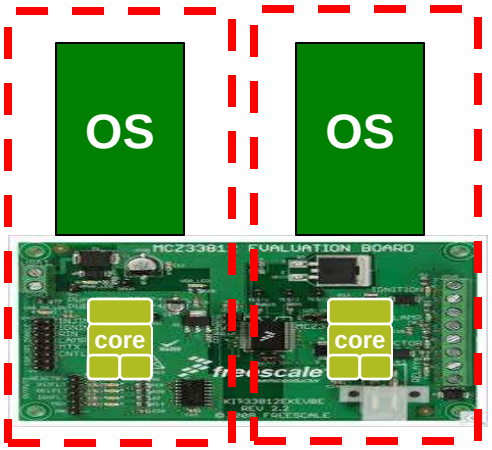


A **hypervisor** is a software component that creates and manages virtual machines which can run operating systems.

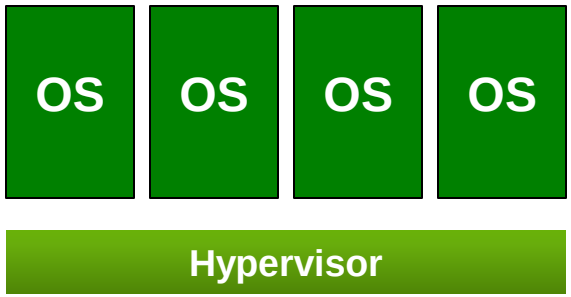
Partitioning and Virtualization

Partitioning ⟷ **Virtualization**

- Consolidation
- Direct hardware access
- Dedicated CPUs, I/O devices
- Minimal sharing



- Resource utilization
- Many virtual machines
- Resources are shared/virtualized
- Oversubscription – CPUs, I/O

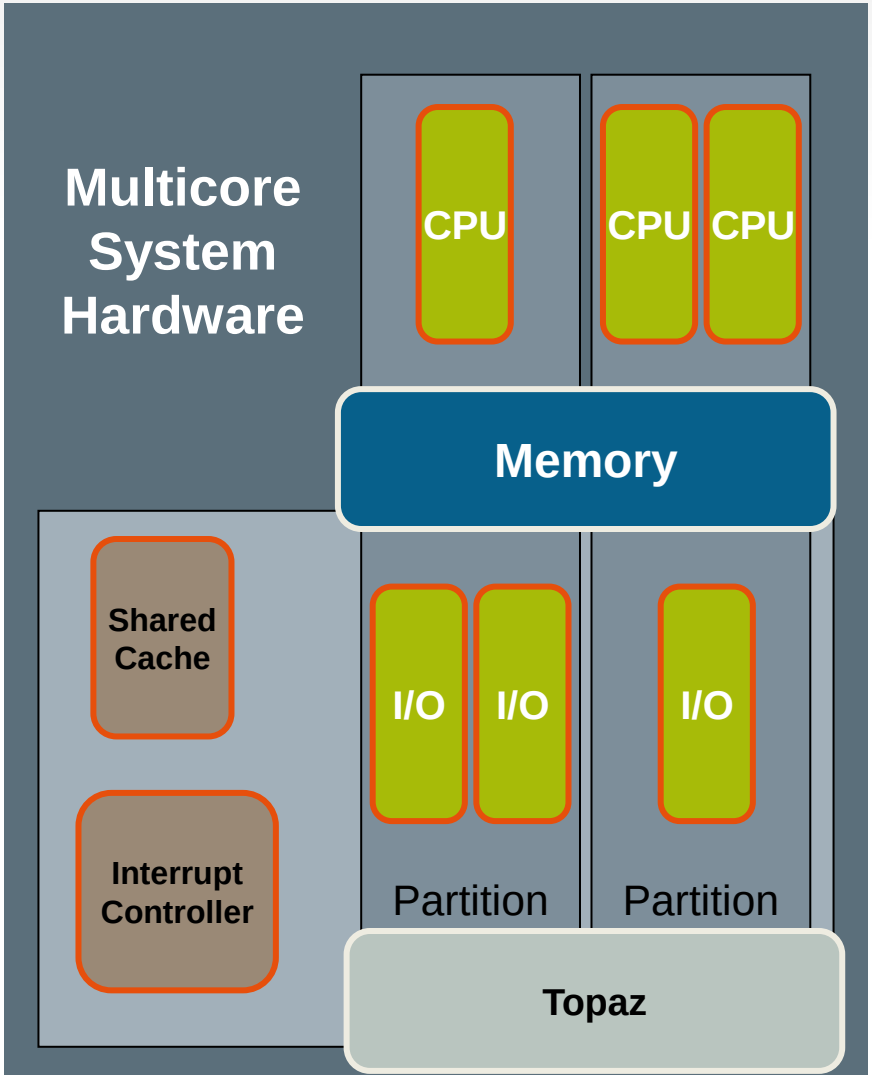


FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

Topaz Overview

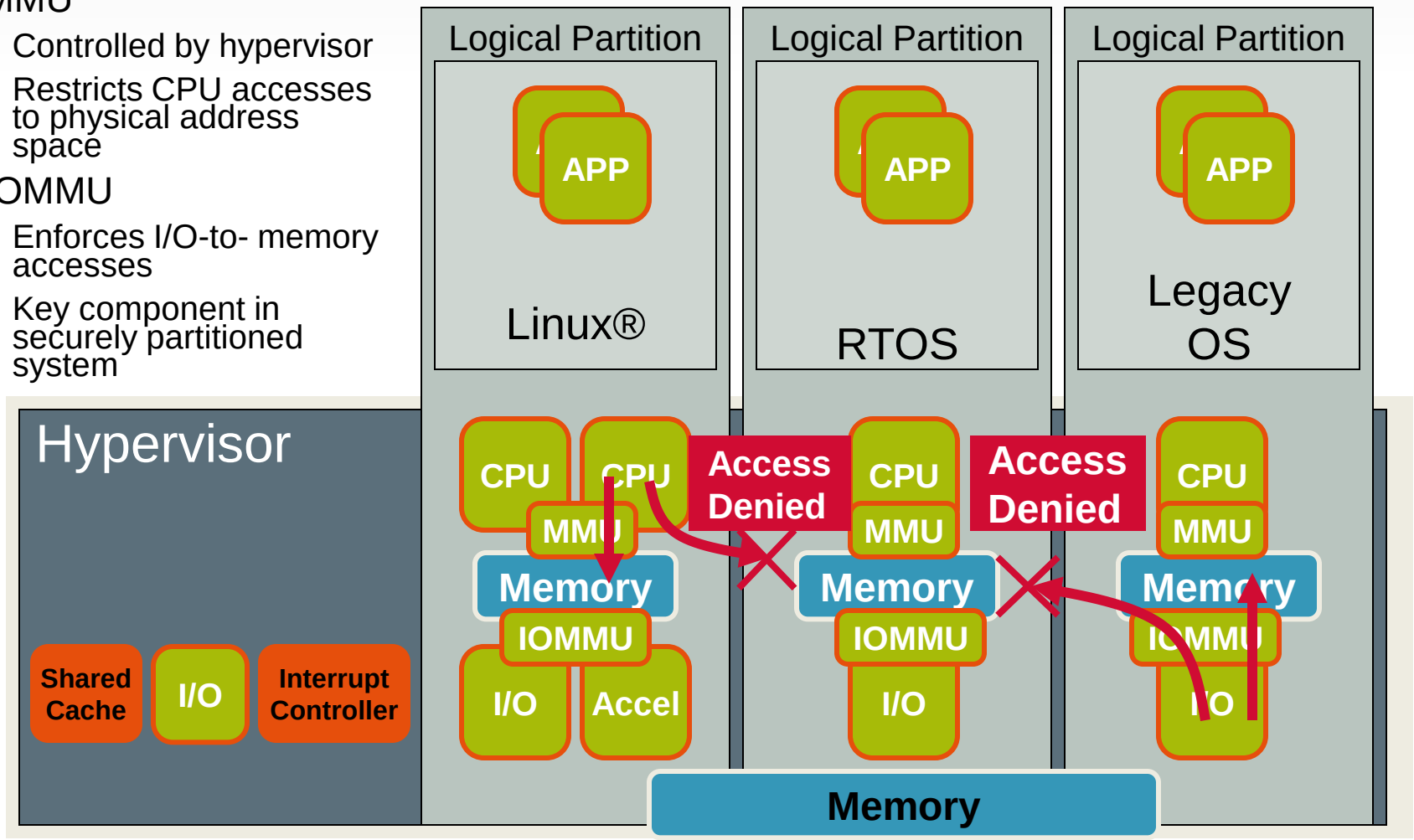
Topaz Implementation

- A “light weight” hypervisor for embedded systems
  - ePAPR compliant
- Primarily focuses on partitioning
  - CPUs, memory and I/O devices can be divided into logical partitions
  - Supports single guest per core
  - Direct device Assignment to guest
    - Limited virtualized I/O support, no virtio
- Designed to leverage E.HV features in the e500mc/e5500 cores
- Uses a combination of full-virtualization and para-virtualization



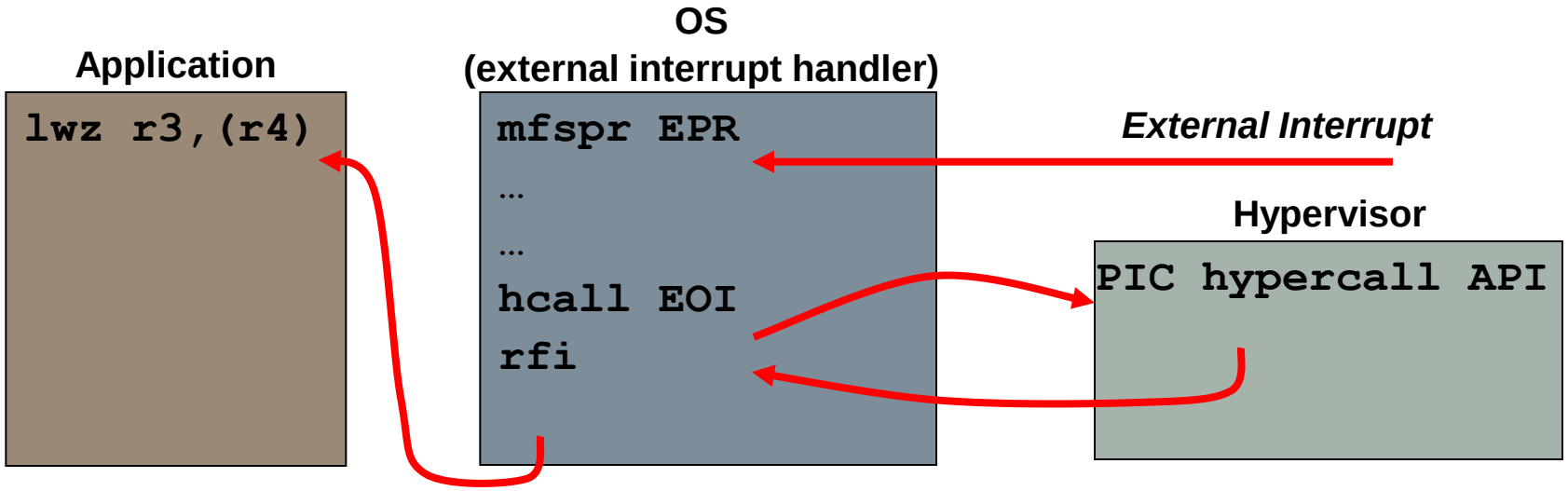
Memory Partitioning

- MMU
  - Controlled by hypervisor
  - Restricts CPU accesses to physical address space
- IOMMU
  - Enforces I/O-to- memory accesses
  - Key component in securely partitioned system



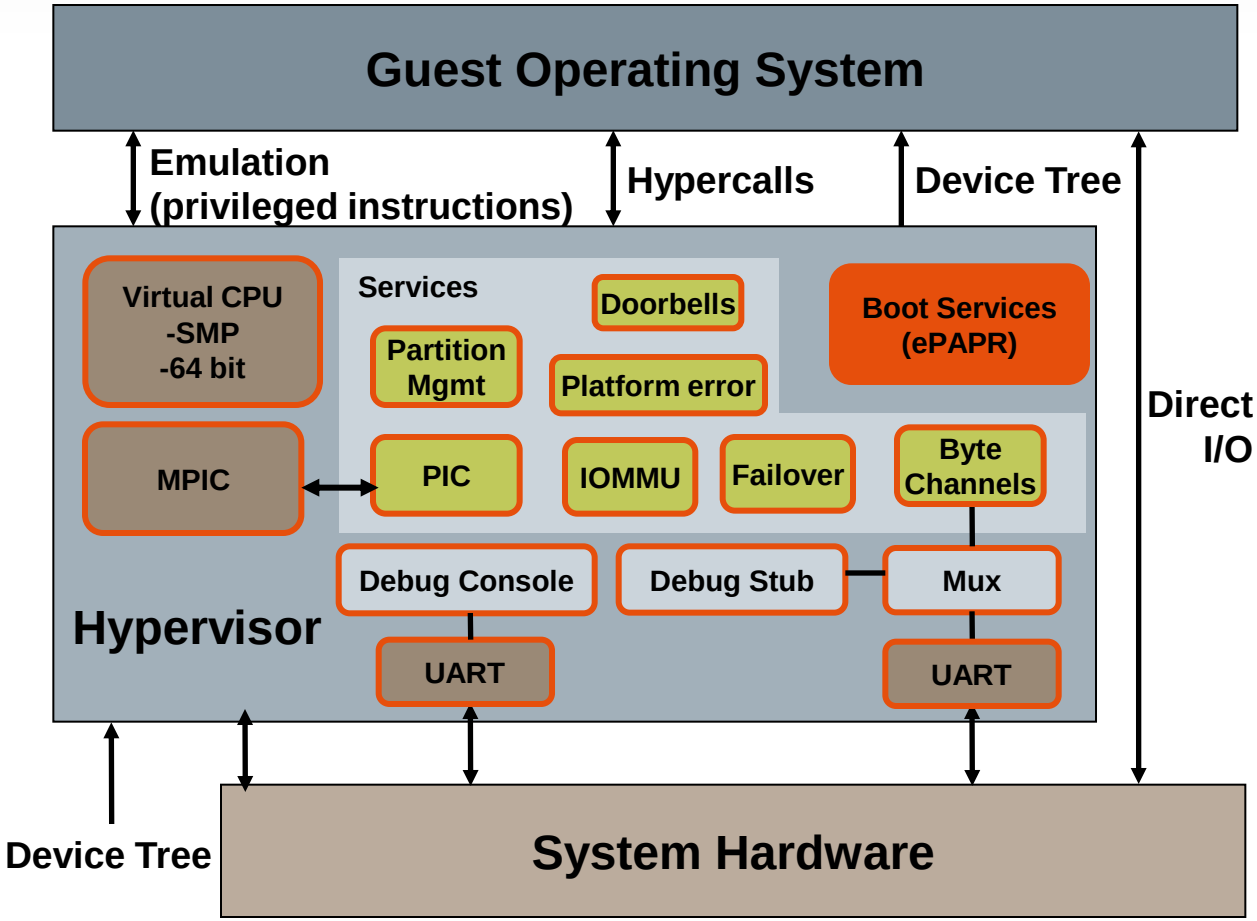
Guest External Interrupt Processing with Topaz

- No latency added by hypervisor for external interrupts
- PIC allows routing of interrupts to specified cores
- External interrupt configured to go directly to guest
- Interrupt acknowledgement automatically done by core and PIC vector is in EPR (external proxy register)



Topaz Features

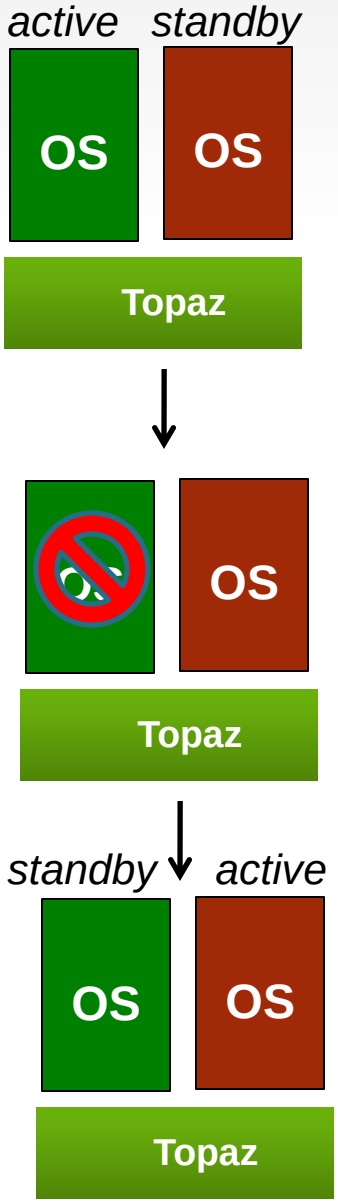
- Operating System sees a virtual core plus hypervisor services
  - Virtual CPU
  - Services via hypercall
  - Debug stub interface for debugging guest operating systems



Use Case - High Availability

- Topaz features mechanisms for configuring partitions in an active/stand-by arrangement

- Features
  - Notifications on partition state changes (e.g. watchdog timeout)
  - Mechanisms for active and standby partitions to share I/O devices– a standby partition that becomes active can claim active ownership
    - Interrupt & DMA reconfiguration
  - Mechanisms to claim error manager
  - If all partitions stop, system will reset

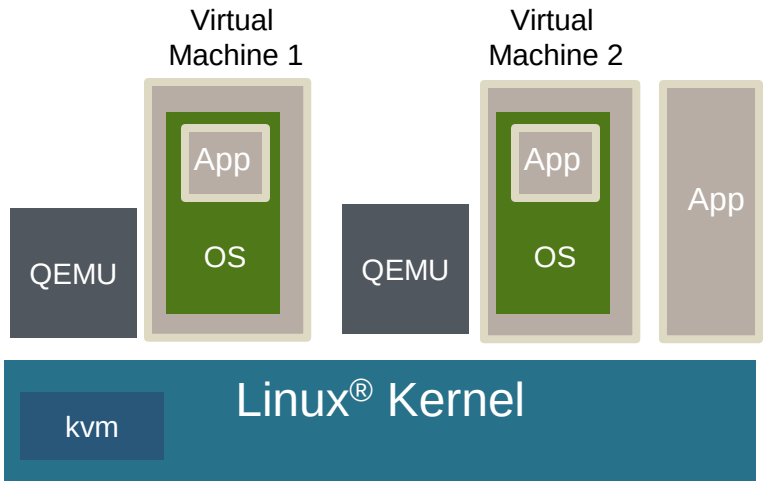


FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

KVM (Kernel-based Virtual Machine) Overview

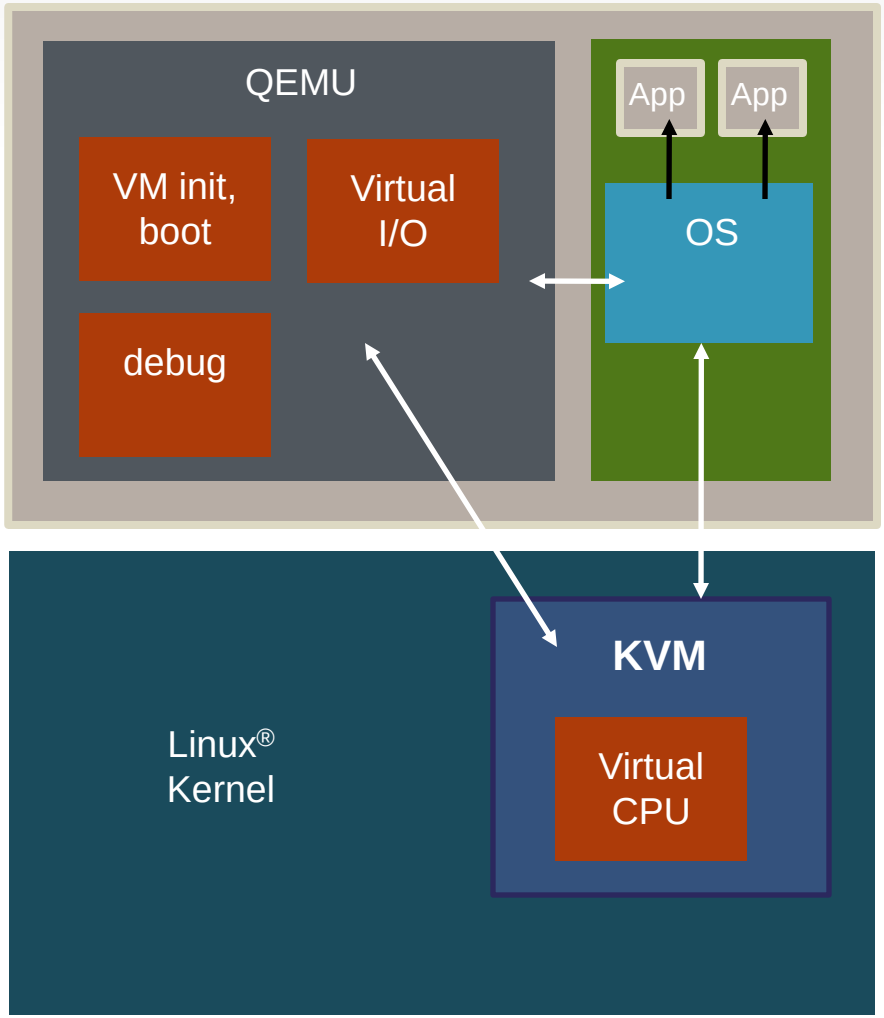
KVM - Overview

- KVM/QEMU– open source virtualization technology based on the Linux kernel
- Supports e500v2, e500mc, e5500 CPUs
- No or minimal OS changes required
- Virtual I/O – virtual disk, network interfaces, serial
- Direct/pass thru I/O – assign SoC devices to partitions (some limitations)
- ePAPR compliant
- e500v2 uses paravirtualization (OS modifications) for improved performance



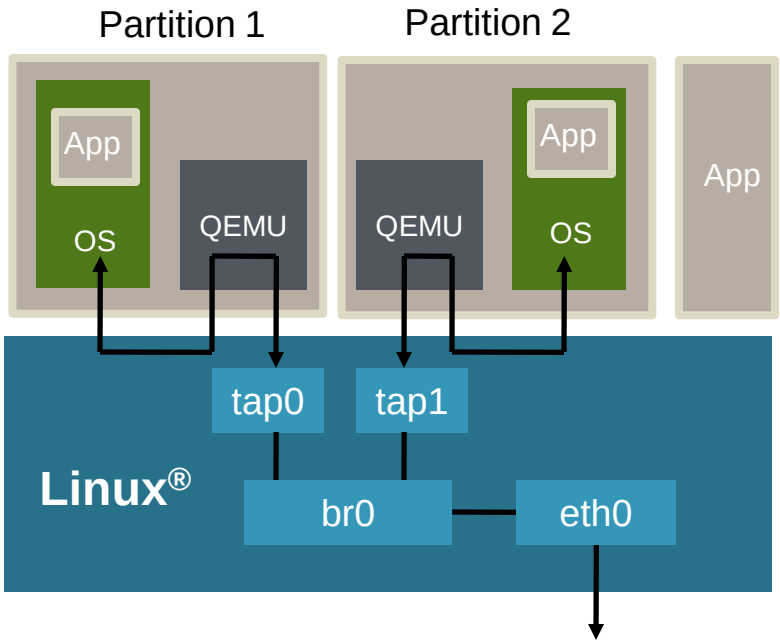
KVM/QEMU – Overview

- QEMU provides
  - Virtual machine setup
  - Initialization
  - Memory allocation
  - Virtual I/O services
  - Debug stub
- KVM provides
  - Virtual CPU services
  - API used by QEMU (see Documentation/kvm/api.txt)
- Kernel schedules VMs



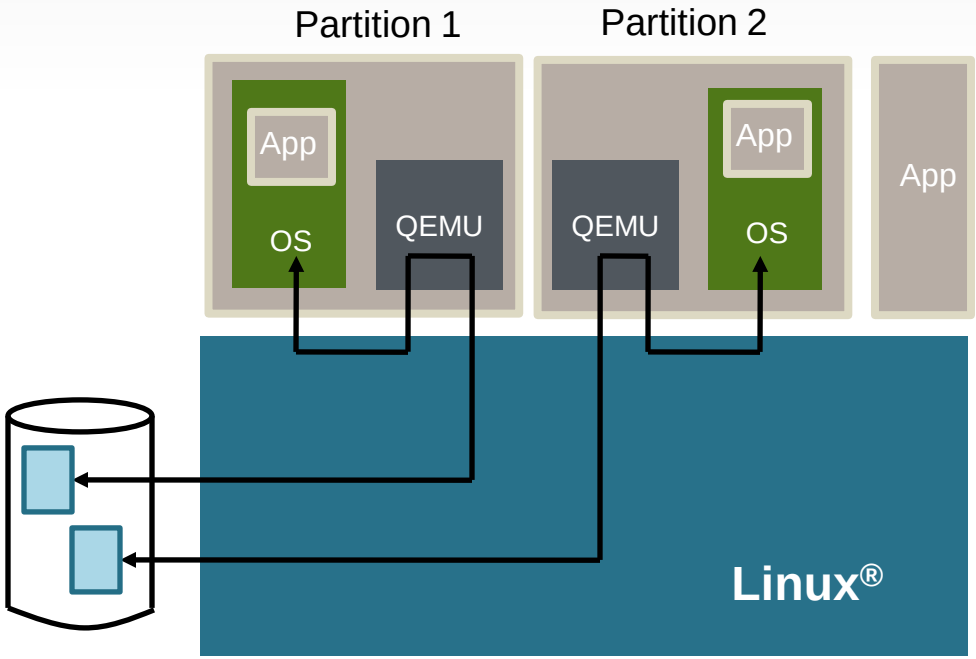
Virtio Networking

- Enables sharing of host network interfaces
- Host
  - Bridge (virtual switch) is connected to physical host interface
  - QEMU uses tun/tap device connected to the bridge
- Guest
  - Each guest sees a private “virtio” network device on PCI bus
  - Virtio network driver is needed in guest



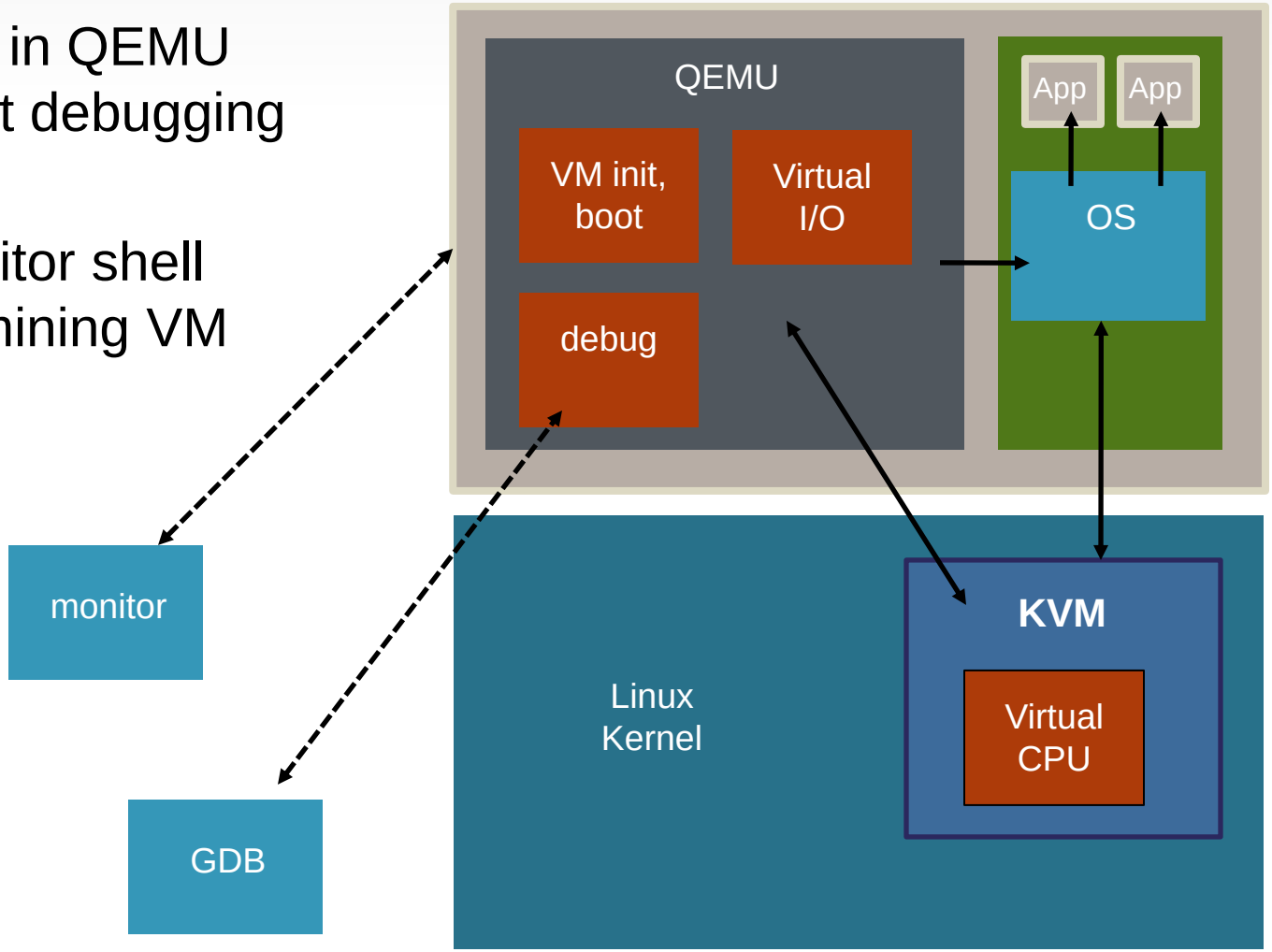
Virtio Block

- Give each virtual machine a private storage device
- Virtual disk could be single binary image on host file system or logical volume on the host's disk
- Guest sees a private "virtio" device on PCI bus
- Virtio block driver is needed in guest

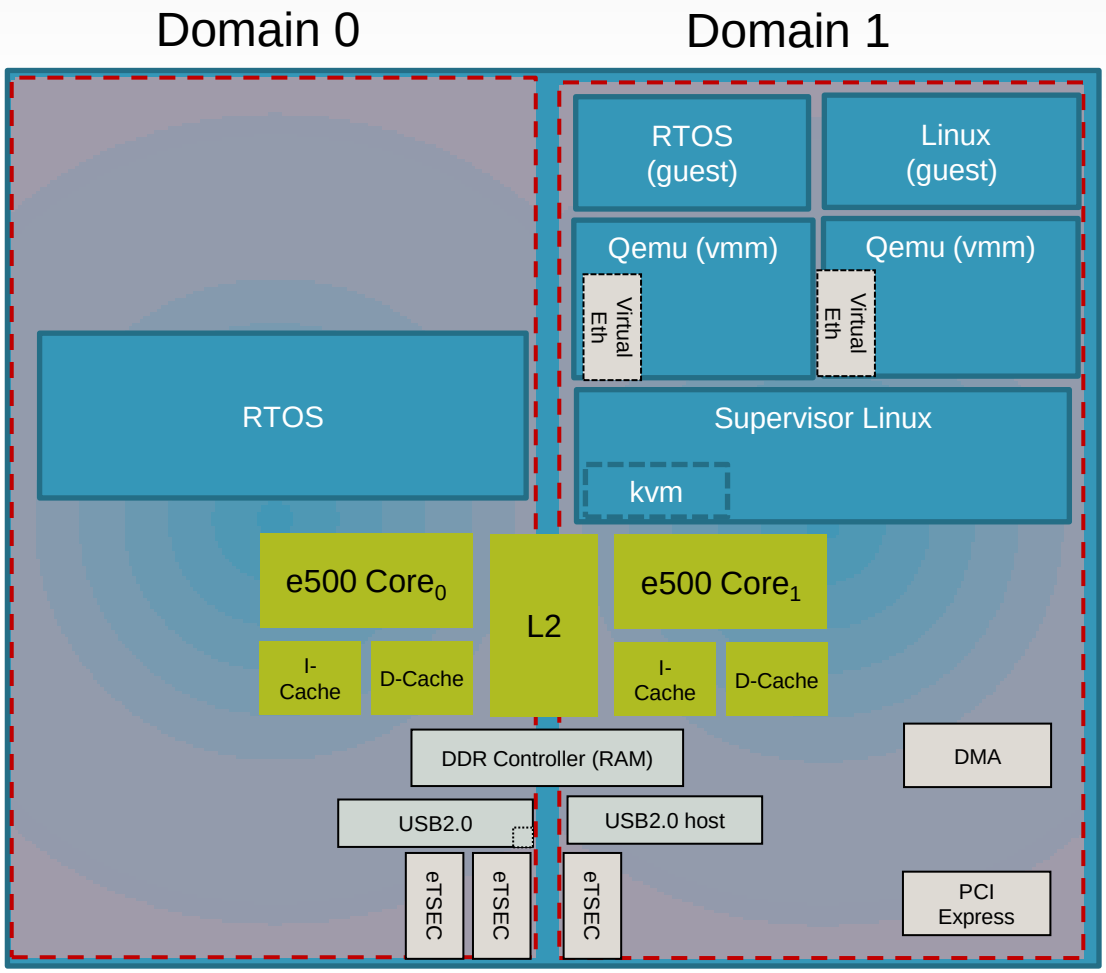


Debugging

- Debug stub in QEMU allows guest debugging using GDB
- QEMU monitor shell allows examining VM state



KVM consolidation Use Case



FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

Performance Considerations

CPU Performance Considerations

- The performance overhead when running on a hypervisor is workload dependent.

- What are the sources of CPU overhead when running under a hypervisor?
  - Privileged operations
    - Instructions– e.g. TLB operations (tlbwe, tlbilx, tlbsx)
    - Privileged SPRs– e.g. DEC, timer control registers
  - Exceptions – Decremeter, TLB misses, DSI/ISI, external interrupts, etc.
  - Scheduling / Context switches
    - May lead to excessive MMU invalidations
  - Hypercalls

freescale QorIQ

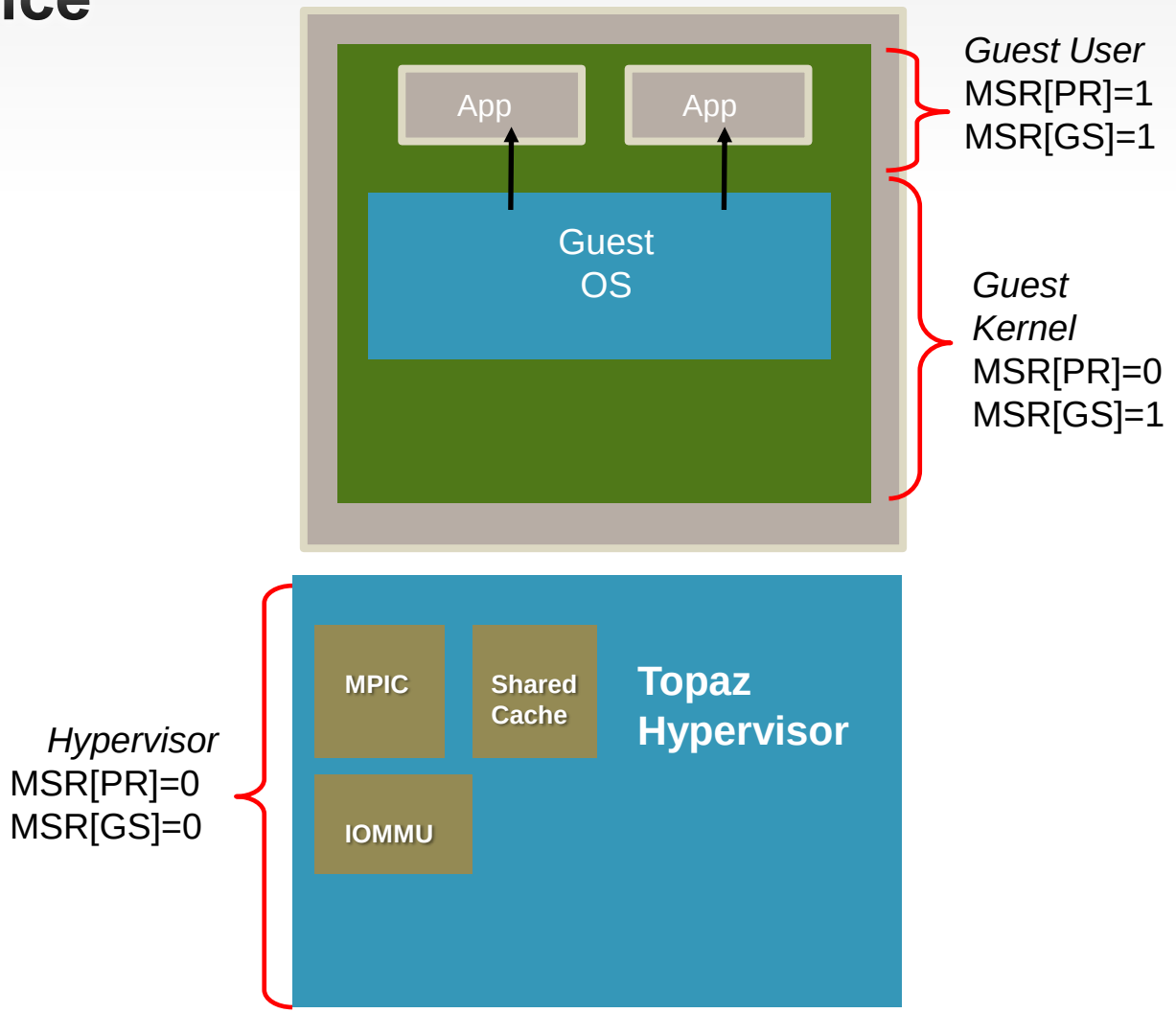
Core Support for Virtualization on QorIQ Silicon

- Virtualization extensions on e500mc / e5500 /e6500 cores
  - HV privilege level
    - Only E.HV privilege instructions trap, reduces the trap overhead
  - Partition ID / extended virtual address space
    - Possible to maintain multiple guest mappings on a single core
  - Shadow registers
    - Private copy of registers each for the HV and the guest state
  - Direct system calls
    - No trap during guest system calls
  - Direct external hardware interrupts to guest
    - Reduced interrupt latency with direct assigned devices
- LRAT support on e6500
  - Tlbwe instruction can be executed without trap

freescale QorIQ

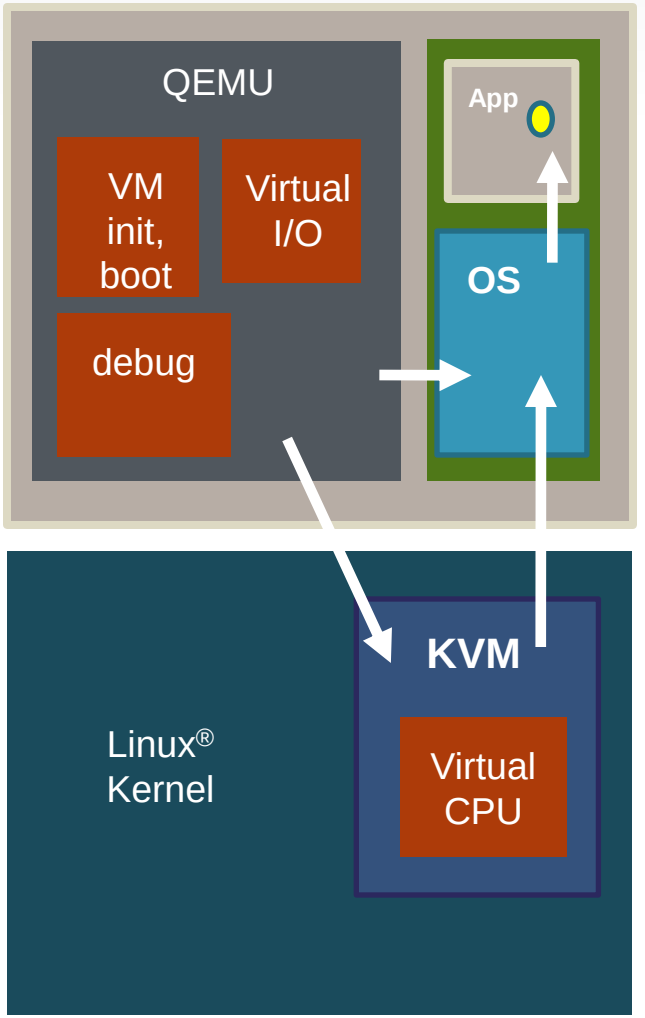
Topaz Performance

- Designed to leverage the virtualization extensions available on QorIQ platforms
- Minimize privilege instruction trap overhead by utilizing additional privilege level
- Supports Direct exception/interrupt delivery to guest

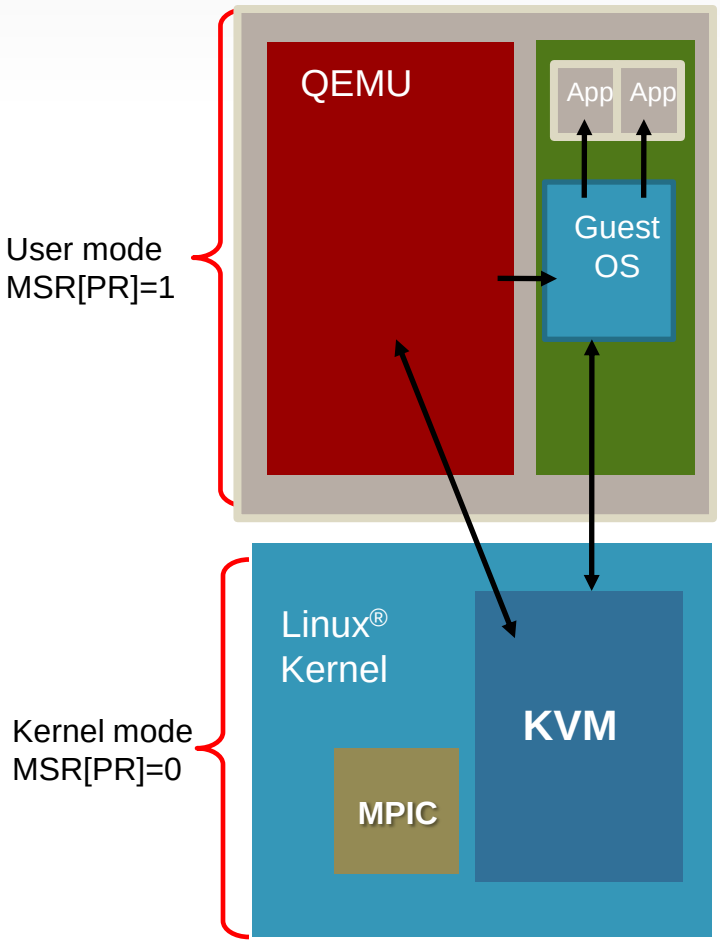


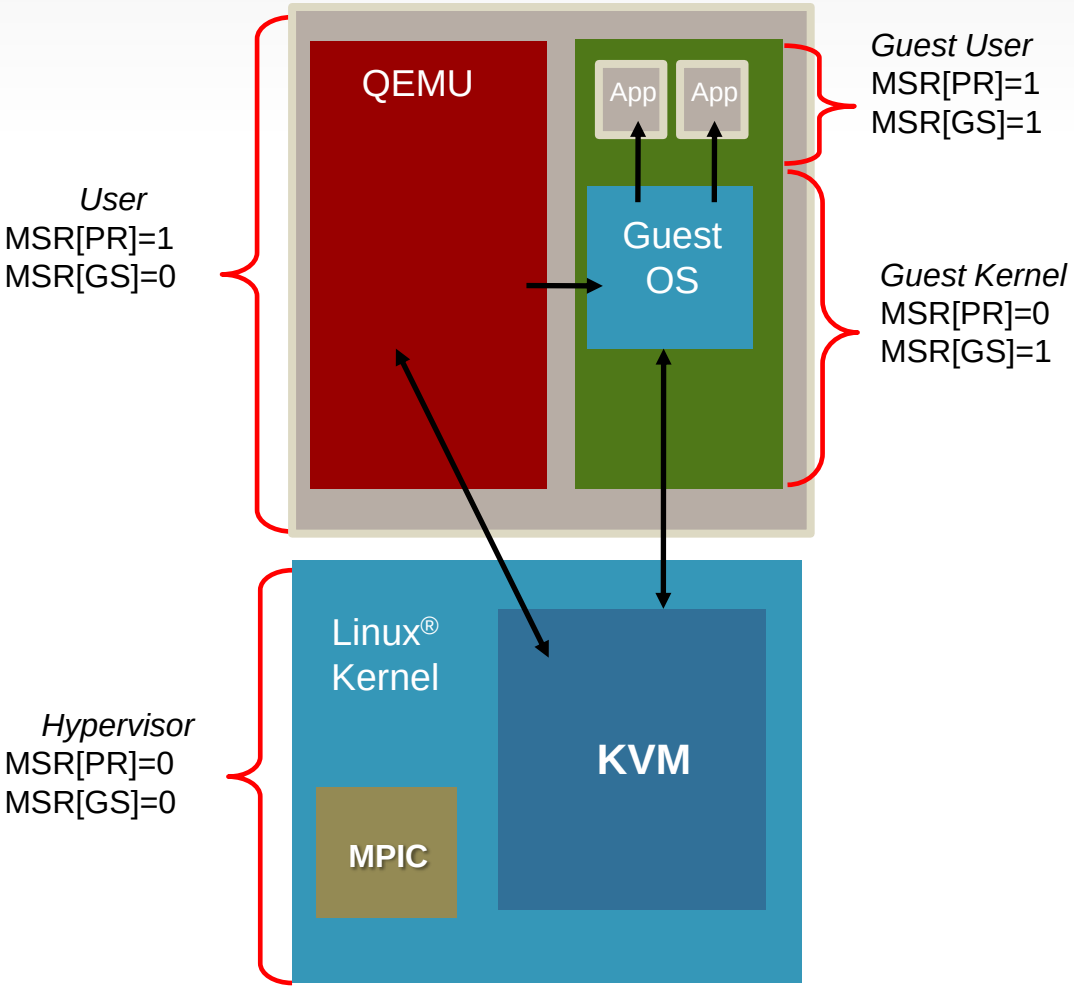
KVM Performance

- CPU utilization – VM competes with Linux applications
- Interrupt latency – Not able to support direct interrupt delivery
- I/O emulation overhead
- Privileged instruction overhead on e500v2 based SoCs
  - Has 2 privilege levels, OS runs in user mode – additional complexity and overhead due to this. OS modifications needed.



KVM – e500v2 vs e500mc





Summary

- Topaz is a good solution if:
  - Simply trying to statically partition hardware
  - Real time constraints
  - Topaz type failover

- KVM is a good solution if:
  - Your system is based on Linux, and you want to run an additional OS in a virtual machine
  - Need the additional features that come with KVM/QEMU
    - Need a scheduler to run multiple Oses
    - Need virtual I/O-- disk, network

For Further Information

- Topaz Users Guide in Freescale QorIQ SDK
- *KVM Users Guide* in Freescale QorIQ SDK
- *Freescale Power Architecture Book E Virtual CPU Specification*
- KVM
  - KVM website: http://www.linux-kvm.org
- QEMU
  - QEMU website: http://www.qemu.org
- Device Trees
  - ePAPR (Embedded Power Architecture Platform Requirements) version 1.1. https://www.power.org/resources/downloads/Power_ePAPR_APPROVED_v1.1.pdf

Facebook.com/Freescale
Tag yourself in photos
and upload your own!

Tweeting?
Please use hashtag
#FTF2012

Session materials will be posted @ www.freescale.com/FTF
Look for announcements in the FTF Group on LinkedIn or follow Freescale on Twitter

freescale QorIQ


freescale™