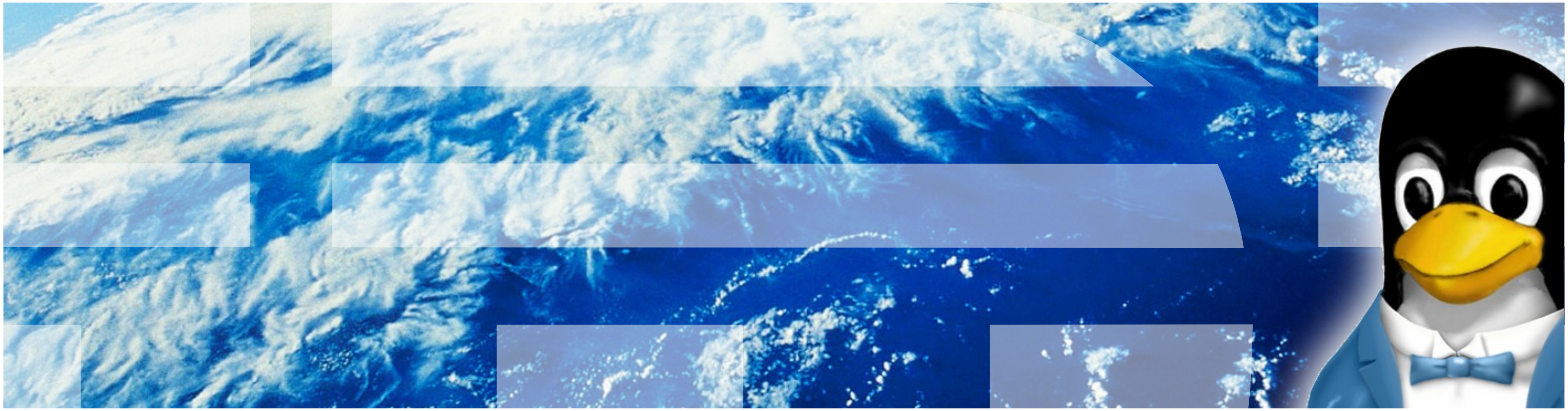
KVM on the IBM POWER7 Processor



Paul Mackerras, IBM LTC OzLabs

paulus@au1.ibm.com

Outline

- Introduction
- POWER® Architecture
- POWER7® Logical Partitioning facilities
- POWER7 Processor
- POWER Server Hardware
- KVM on POWER7

Introduction

- **Goal: provide KVM as a viable virtualization technology on POWER systems**
- **Currently PowerVM™ is the only option available to customers for virtualizing POWER systems**
  - In fact the hypervisor (pHyp) is present even if all the machine resources are dedicated to a single OS instance
  - pHyp does various platform management and error recovery functions
- **All operating system kernels for POWER machines assume paravirtualization**
  - Linux®, AIX®, i/OS
  - Platform/hypervisor interface defined by PAPR document (Power Architecture Platform Requirements)
- **Power ISA includes hardware support for virtualization**
  - Three privilege levels: hypervisor, supervisor, user
  - Logical partitioning facilities available in hypervisor state
- **We want to use these logical partitioning facilities for KVM**
  - Have to boot without pHyp
  - Alternative firmware, that gives us access to hypervisor state, is in development

POWER Architecture

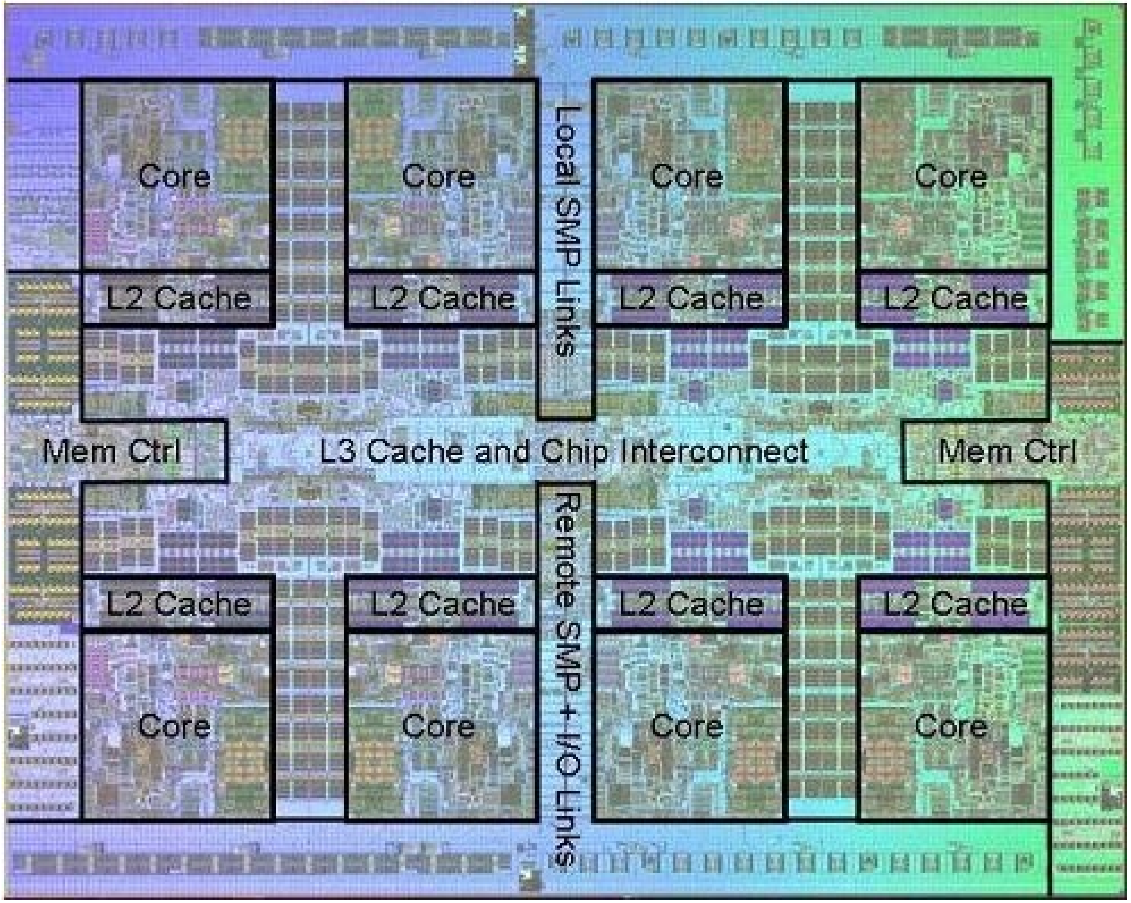
- **RISC architecture**
  - Many registers, fixed-length instructions
  - Only load and store instructions access memory; no hardware stack
  - All I/O is memory-mapped
- **Unusual memory management architecture**
  - Two stages of translation: Effective address → Virtual address → Real address
  - Effective → virtual translation under OS (supervisor) control, via SLB
    - Granularity is coarse: 256MB or 1TB
    - Used by Linux to give each process a separate virtual space
  - Virtual → real translation under hypervisor control, via MMU hash table
    - One hash table per partition
    - Multiple page sizes supported: 4kB, 64kB, 16MB, 16GB
- **Defined by Power ISA V2.06b specification**
  - available from www.power.org

POWER7 Logical Partitioning facilities

- **Hypervisor mode, with some instructions and special-purpose registers (SPRs) only accessible in hypervisor mode**
  - Anything to do with MMU hash table, so the hypervisor controls which pages of memory and which I/O devices each partition can access
  - Control of which interrupts go to the guest directly and which go to the hypervisor
  - Control of timeslices allocated to partitions via the HDECR
- **Designed for partitioning rather than for emulation, and paravirtualization rather than full virtualization**
  - Some aspects of the machine can't be spoofed or emulated by the hypervisor
    - Processor version register
    - Privileged instructions and registers
  - Restriction that all SMT threads in a core have to be in the same partition
    - POWER7 supports 4 threads per core
- **Partition Compatibility Register allows emulation of earlier POWER processors**
  - POWER5++, POWER6®
  - Disable new instructions and facilities (e.g. VSX)

POWER7 Processor

- **567 mm^2, 45nm lithography**
- **1.2B transistors**
- **Eight processor cores**
  - 12 execution units/core
  - 4-way SMT
  - 32 threads per chip
  - 256kB L2 cache/core
- **32MB on-chip eDRAM shared L3 cache**
- **Dual DDR3 memory controllers**
  - 100GB/s memory BW per chip
- **Scalability up to 32 sockets**
  - 360 GB/s SMP BW/chip
  - 20000 coherent ops in flight

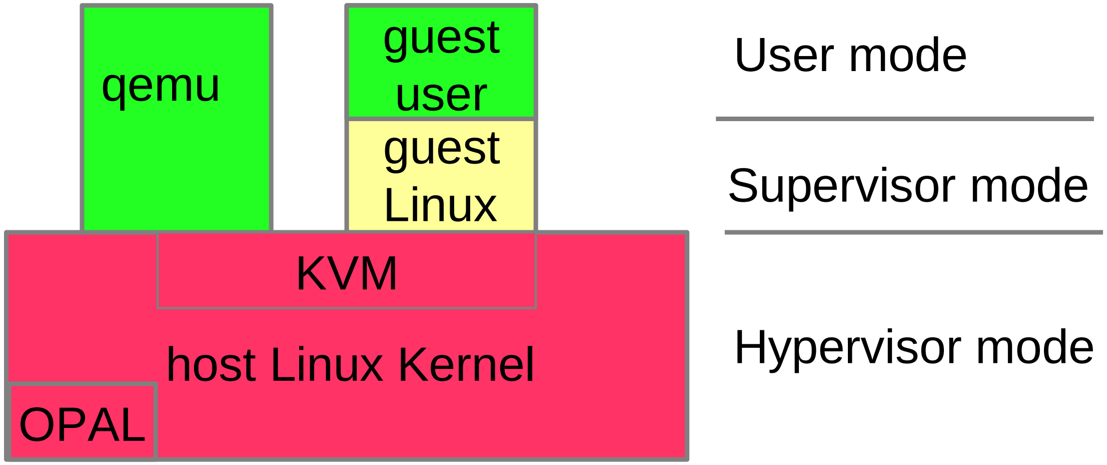


POWER7 Server Hardware

- **Scalable SMP**
  - Multiple point-to-point buses connecting processor chips and I/O hub chips
    - “Fabric” rather than a single bus
  - Memory controllers integrated on processor chips
  - Cache coherence maintained across the whole machine
- **I/O Hub chips integrate multiple PCI-Express host bridges with IOMMUs**
  - Advanced virtualization and isolation facilities
    - Ensure that device can be confined to a partition
  - “Enhanced Error Handling” (EEH)
    - Immediately isolate a device from the system on error
      - PCI errors
      - DMA access outside permitted bus address range
    - Designed to provide isolation and prevent propagation of bad data
  - Concept of “partitionable endpoint” (PE)
    - PE is the unit that can be isolated on error
    - Bus address space for DMA partitioned between PEs
    - Each PE has its own IOMMU table (TCE table)

KVM on POWER7

- **Host kernel runs in hypervisor mode, guest kernel runs in supervisor mode**
  - Host has access to all memory and all machine resources
  - No need to trap and emulate privileged instructions in guests
  - Allows us to run guests at full hardware speed
  - Require special firmware to gain access to hypervisor mode
- **Partition switch on entry to and exit from guests**
  - Each guest has its own MMU hash table
  - Host kernel runs in partition 0 and has a separate MMU hash table
- **Guests are paravirtualized using the PAPR interface**
  - Can run existing Power Linux distributions (RHEL6, SLES11SP1, etc.)



KVM on POWER7

- **New machine type “pseries” added to qemu**
  - Qemu can emulate a pseries partition on x86 machines
  - Guests run pSeries kernels
- **New KVM flavor “book3s_hv” added to kernel**
  - Alternative to “book3s_pr” flavor of KVM developed by Alexander Graf, which runs the guest in user mode and emulates privileged instructions and facilities
- **Virtual I/O supported with both PAPR and virtio interfaces via qemu**
  - PAPR VSCSI for virtual disks and CD, VETH for virtual networking
  - PAPR console interfaces
  - virtio-blk, virtio-net, etc. (require drivers in guest)
    - virtio “doorbell” mechanism: hcall or emulated MMIO?
- **Partition firmware is open-source SLOF (Slim-Line Open Firmware)**
- **New platform type “powernv” (Power non-virtualized) in host kernel**
  - Enabled by CONFIG_PPC_POWERNV
  - Platform code in arch/powerpc/platforms/powernv
  - “OPAL” platform firmware abstracts some platform details
    - Console, RTC, NVRAM, power/reboot control
    - Some arcane platform error recovery details

KVM on POWER7

- **Features currently working:**
  - Run guest in supervisor mode
  - Guest memory backed by 16MB large pages (not pageable/swappable)
  - Guests can use ST, SMT2, SMT4 processor core modes
    - Host kernel must run in 1 thread/core (ST) mode because of hardware restriction that all CPU threads must be in the same partition
  - Virtual I/O: both PAPR and virtio
  - Runs on POWER7 and PPC970 (YDL PowerStation)
- **Features currently being worked on/in prototype state:**
  - Guest memory backed by small (4k or 64k) pages, pageable/swappable
    - MMU notifier support
  - PCI pass-through
    - Current prototype works but requires modifications to VFIO
  - Emulated MMIO
    - Slight performance hit (0.5%) for routing page faults through host
    - Current guests don't expect MMIO emulation because pHyp doesn't support it
  - CPU/memory affinity (NUMA support)
  - Performance and robustness improvements
  - Porting libvirt and virt_manager

KVM on POWER7

- **Known issues/problems**
  - Performance of SMT4 guests
    - Every exit to the kernel requires pulling all 4 threads out of the guest
    - Need to handle as many exits as possible in hypervisor real mode
      - Can access all memory, but MMU context is still the guest
      - Doesn't require pulling the other threads out of the guest
  - Mismatch between VFIO expectations/requirements and Power hardware capabilities and limitations
- **Future work**
  - SR-IOV support
  - Hot-plug CPU, memory, I/O
  - Guest hibernation and migration
  - Guest power management
  - Data de-duplication
  - Better support in management tools

Legal Statement

This work represents the view of the author and does not necessarily represent the view of IBM.

IBM, IBM (logo), AIX, POWER, POWER6, POWER7 and PowerVM are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.