OAK RIDGE National Laboratory

OS/R support for Multi-Tenancy in Supercomputing Systems

Jack Lange

Oak Ridge National Lab: Senior Systems Scientist

University of Pittsburgh: Associate Professor

ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Overview

- **Multi-tenancy requires resource partitioning**

- OS/R features for node level resource partitioning
  - Containers
  - Virtual Machines
  - Co-kernels
- On demand portioning
  - VM Lifting
- Security isolation

Multi Tenancy in Cloud environments

- Cloud has been doing multi-tenancy for a while
  - Have developed a significant amount of system software to support it
    - Bare metal hypervisors, secure partition managers, etc…
  - HPC centers are significantly behind
- Why not just reuse cloud approaches?
  - Different use case:
    - Cloud approaches focus on commodity workloads
  - OLCF evaluation of cloud environments has identified issues
    - HPC offerings in the cloud are abstractions over commodity infrastructure

- **Supporting Multi-Tenancy effectively on HPC systems requires specialized system software support**

HPC System Software Support for Multi-Tenancy

- **Hardware features are available to support HPC multi-tenancy**
  - People often conflate HW features with commodity software stacks that use them
- **Example:** Virtualization extensions
  - HW virtualization extensions provide resource partitioning capabilities
  - Do not require full system emulation via a VMM
  - Specialized
- Relying on commodity system software is a **choice**
  - Driven by business and other practical considerations (not technical ones)
    - Not all system software needs to be as complex as Linux

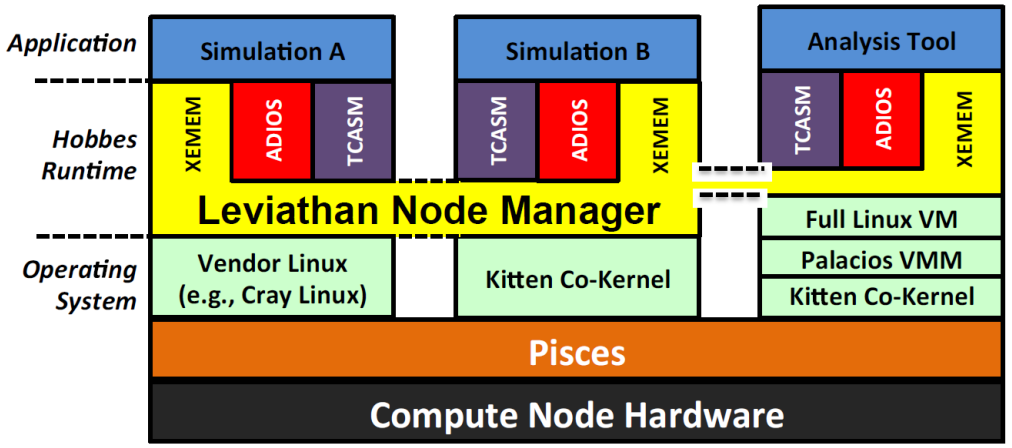
Multi-Tenancy vs Composed Applications

- **In-situ post processing and composed applications were expected to be Exascale workloads**
  - Multiple DOE/NSF research projects worked on this

- **Hobbes: Scalable application composition across multiple local OS/R stacks**
  - Partition local node hardware into separate system software domains
  - Hardware partitions (**enclaves**) run specialized OS/R stacks with specific App components
  - Flexible and specialized OS/R compositions
    - Native Linux Apps (using standard environment)
    - Native LWKs
    - Virtualized OS/Rs
      - Specialized Linux environments

Several other contemporary projects had similar approach

- Riken: McKernel (Fugaku)
- Intel: mOS (Aurora?)



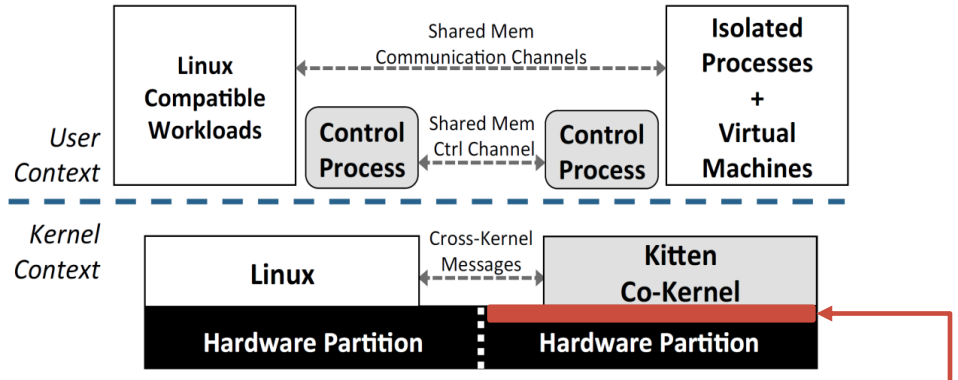
- B. Kocoloski, et al, **System-Level Support for Composition of Applications,** ROSS 2015
- J. Ouyang, et al, **Achieving Performance Isolation with Lightweight Co-Kernels**, HPDC 2015
- B. Kocoloski, et al, **XEMEM: Efficient Shared Memory for Composed Applications on Multi-OS/R Exascale Systems**, HPDC 2015

Covirt: Co-kernel fault isolation

- **Co-kernel approaches generally assume 1 job / node**
  - Each job contains multiple app workloads

- Native co-kernels run will full hardware access capabilities
  - Explicitly configured to ignore other resource partitions
  - Resource partitions are dynamic and can become inconsistent across OS/R instances
- Consistency bug in one co-kernel can take down entire node
  - **Not suitable for true multi-tenancy**

- Solution: Insert thin hypervisor layer for resource protection
  - Hypervisor **only** provides resource protection
  - VM Configuration managed by Host OS
    - Kept consistent with Host OS state
    - Isolates co-kernel bugs to inside the enclave resources

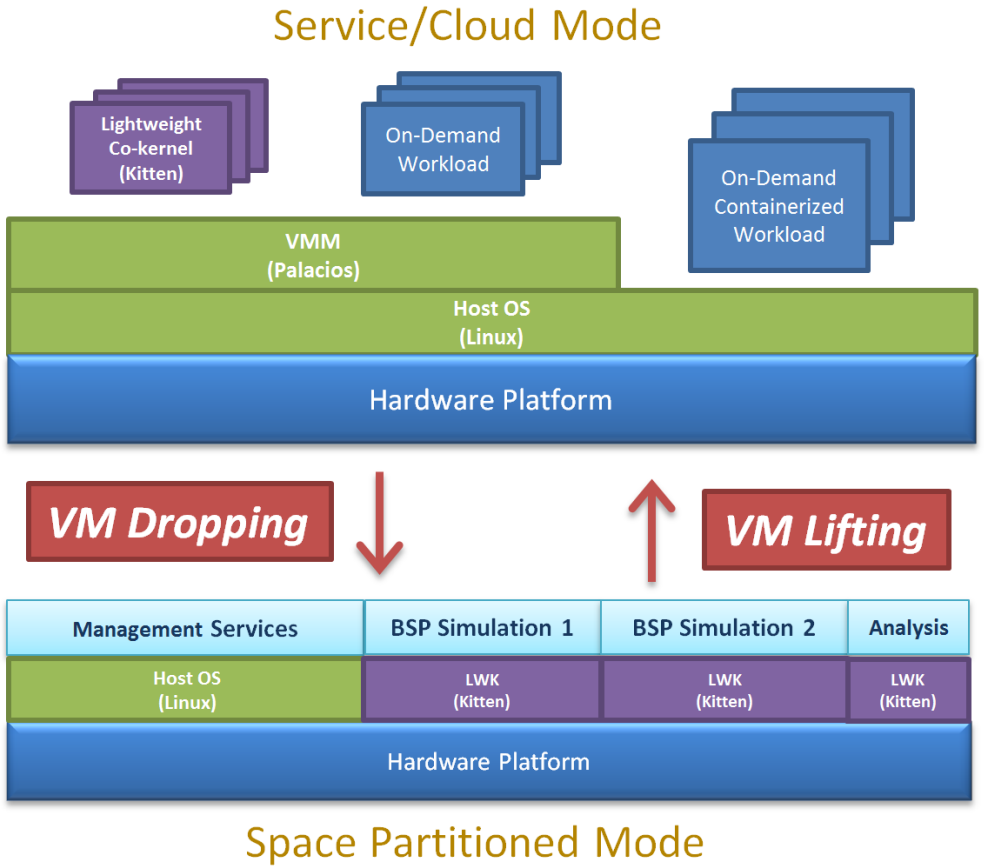


N. Gordon and J. Lange, **Covirt: Lightweight Fault Isolation and Resource Protection for Co-Kernels**, IPDPS 2021

OAK RIDGE National Laboratory

Preemption for On-Demand Workloads

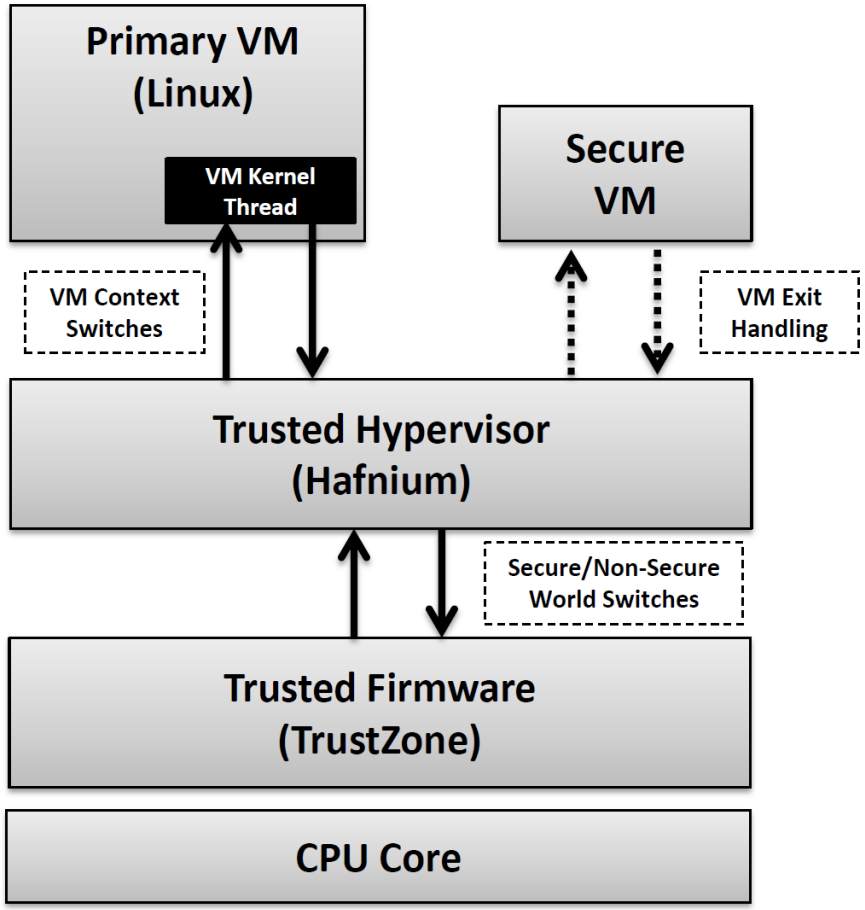
- Multi-Tenancy use cases imply comingling on-demand and batch scheduled workloads
  - **To ensure high utilization we need to be able to preempt jobs**
  - Can we preempt without killing a job?
- Dynamically switch between space-shared and time-shared partitioning
  - Leverage HW virtualization extensions
    - Switch between bare-metal hypervisor and type-2 VMM
  - VM-Lifting
    - Migrate native OS/R into a VM
    - Schedule node using an IaaS model
  - VM-Dropping
    - Migrate virtual OS/R to bare metal
    - Schedule using space-shared model



N. Gordon and J. Lange, **Lifting and Dropping VMs to Dynamically Transition Between Time- and Space-sharing for Large-Scale HPC Systems**, HPDC 2022

Security partitioning

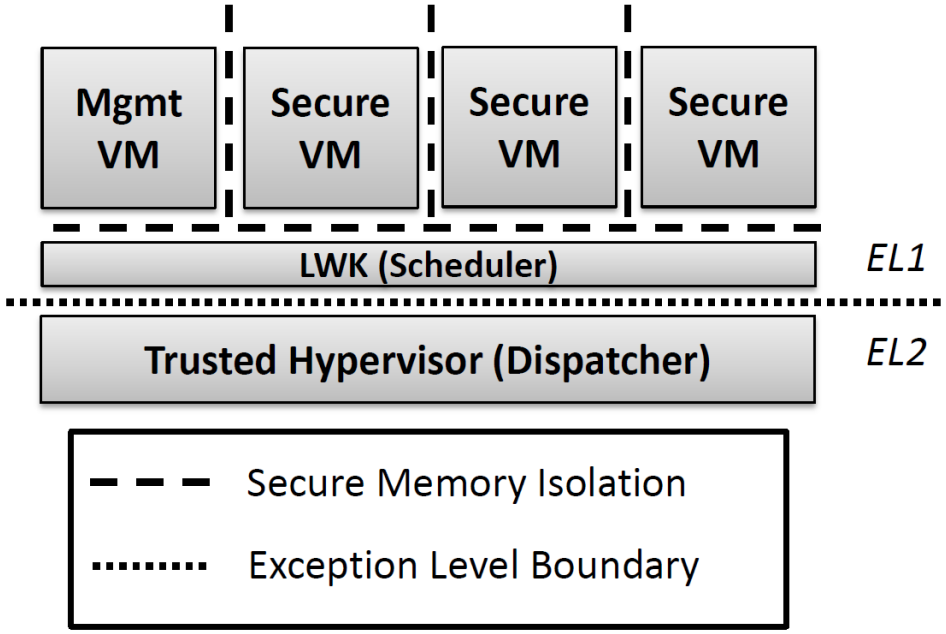
- **Multi-tenancy requires security isolation**
  - OS/VMs/Containers provide software enforced isolation
  - HW security isolation features are becoming prevalent
- HW security w/ virtualization can provide secure compartmentalization between multi-tenant environments
  - Clouds are already deploying this capability
    - E.g. Amazon's AWS Nitro
- At Pitt, we have developed a proof-of-concept HPC secure partitioning hypervisor for ARM64/TrustZone
  - Partitioning hypervisor + HPC specialized resource manager



J. Lange, et al, **Low Overhead Security Isolation using Lightweight Kernels and TEEs**, ROSS 2021

Security partitioning

- **Multi-tenancy requires security isolation**
  - OS/VMs/Containers provide software enforced isolation
  - HW security isolation features are becoming prevalent
- HW security w/ virtualization can provide secure compartmentalization between multi-tenant environments
  - Clouds are already deploying this capability
    - E.g. Amazon's AWS Nitro
- At Pitt, we have developed a proof-of-concept HPC secure partitioning hypervisor for ARM64/TrustZone
  - Partitioning hypervisor + HPC specialized resource manager



J. Lange, et al, **Low Overhead Security Isolation using Lightweight Kernels and TEEs**, ROSS 2021

Conclusion

- **HW capabilities exist to support node level multi-tenancy**
  - Can be employed efficiently for HPC
- **Commodity system software is not the only solution**
  - Specialized HPC system software stacks can be designed and deployed
- **Multi-tenancy allows system software compartmentalization**
  - We don't need to replace all of Linux
  - Small and lightweight hypervisors can do much of the heavy lifting

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov