

VEE'16 Panel: Sweet Spots and Limits for Virtualization

April 3, 2016

Kevin Pedretti
Center for Computing Research
Sandia National Laboratories



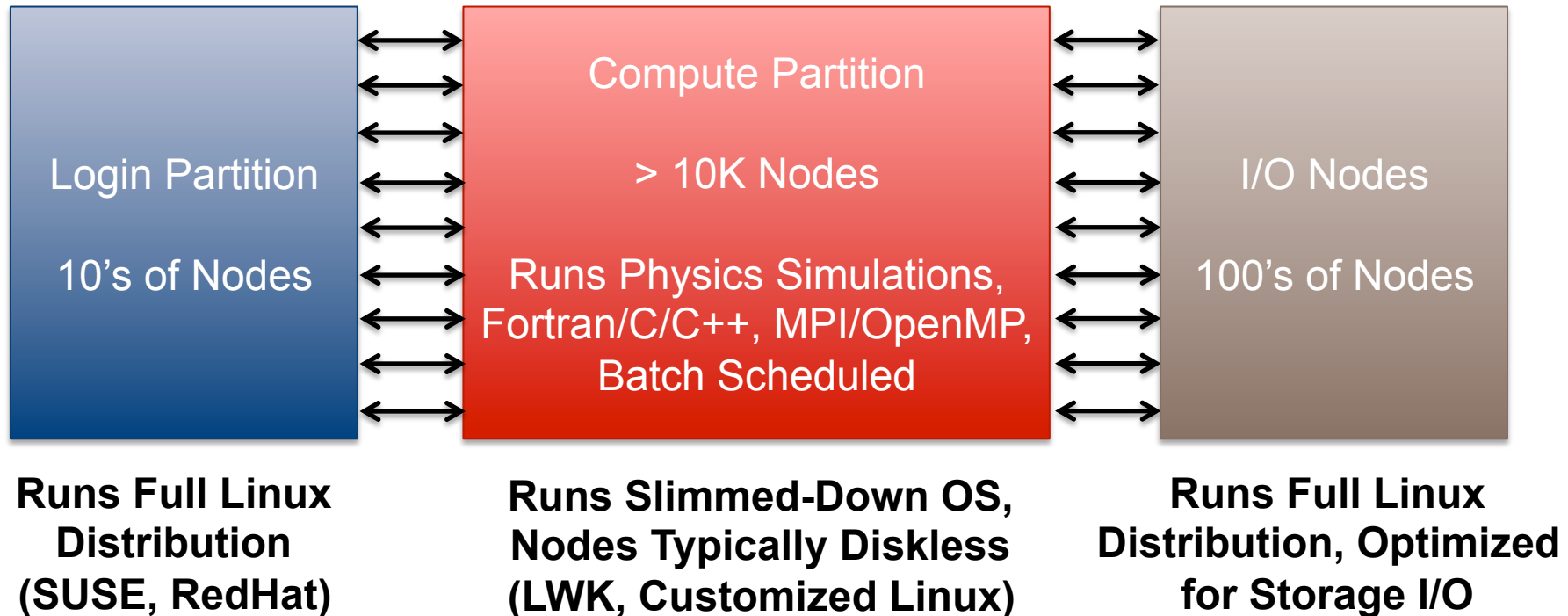
*Exceptional
service
in the
national
interest*




Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Large-Scale HPC Systems in General

- Distributed-memory MIMD
- Message passing used between nodes (MPI)
- Custom interconnect, leverage commodity processors and memory
 - Network point-to-point latency ~ 1 us, bandwidth ~ 10 GBytes/s and growing
- Several node types, system software for each type specialized to task
- Big systems typically \$100M - \$200M procurements



Why Virtualization in Large-Scale HPC?

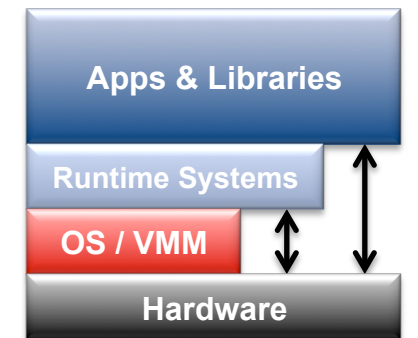
- Support multiple system software stacks in same platform
 - Vendor's stack good for physics simulations, bad for data analytics
 - Virtualization adds flexibility, deploy custom images on demand
 - Not just user-space containers, need ability to run different OS kernels
 - Special-purpose Lightweight Kernels: mOS, McKernel, Kitten 
 - Large-scale emulation experiments, networks + systems
 - Leverage industry momentum, student mindshare

- Virtualization overhead can be very low

- Don't oversubscribe, space share nodes, pin everything, use large pages, physically contiguous virtual memory
 - Demonstrated < 5% overhead in practice on 4K nodes (VEE'11)

- Challenges

- Deployment: getting virtualization into vendor's stack
 - Networking: pass-through OK, but want to be able to share NIC between VMs, getting RDMA drivers in guest, migration
 - Complex nodes: heterogeneous memory, many-core, SMT, NUMA, ...

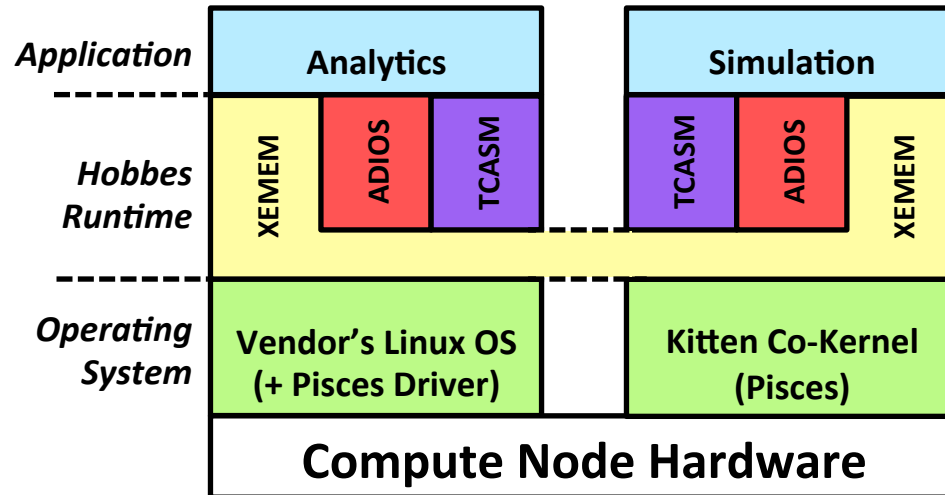


**Compute Node
System Software Stack,
OS Bypass**

Hobbes: Multi-Stack Approach for Application Composition



Node Virtualization Layer (NVL)



HPDC'15

Team:

- Kevin Pedretti, Jay Lofstead, Brian Gaines, Shyamali Mukherjee, Noah Evans (SNL)
- Jack Lange, Brian Kocoloski, Jiannan Ouyang (Pitt)
- Patrick Bridges, Oscar Mondragon (UNM)
- Peter Dinda, Kyle Hale (Northwestern)
- Mike Lang (LANL)
- David Bernholdt (Enclave lead), Hasan Abbasi (ORNL)
- Jai Dayal (GaTech)

■ Key Ideas

- No one-size-fits-all OS/R
- Partition node-level resources into “enclaves”
- Run (potentially) different OS/R stack in each enclave

■ Challenges

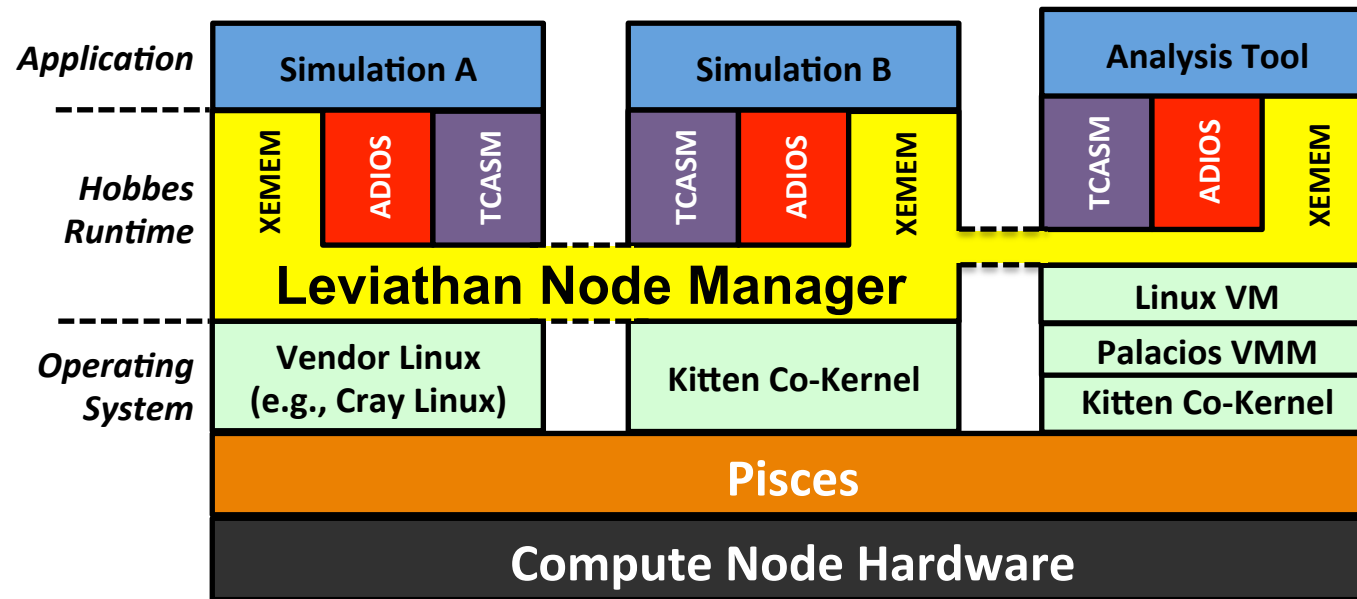
- Performance isolation
- Composition mechanisms

■ Approach

- Build a real, working system
- Leverage Kitten LWK and Palacios Hypervisor
- Use standard Linux host for bootstrap and enclave control
- Develop libhobbes for use by Apps/Tools/Services

Thank You

Hobbes Compute Node OS/R



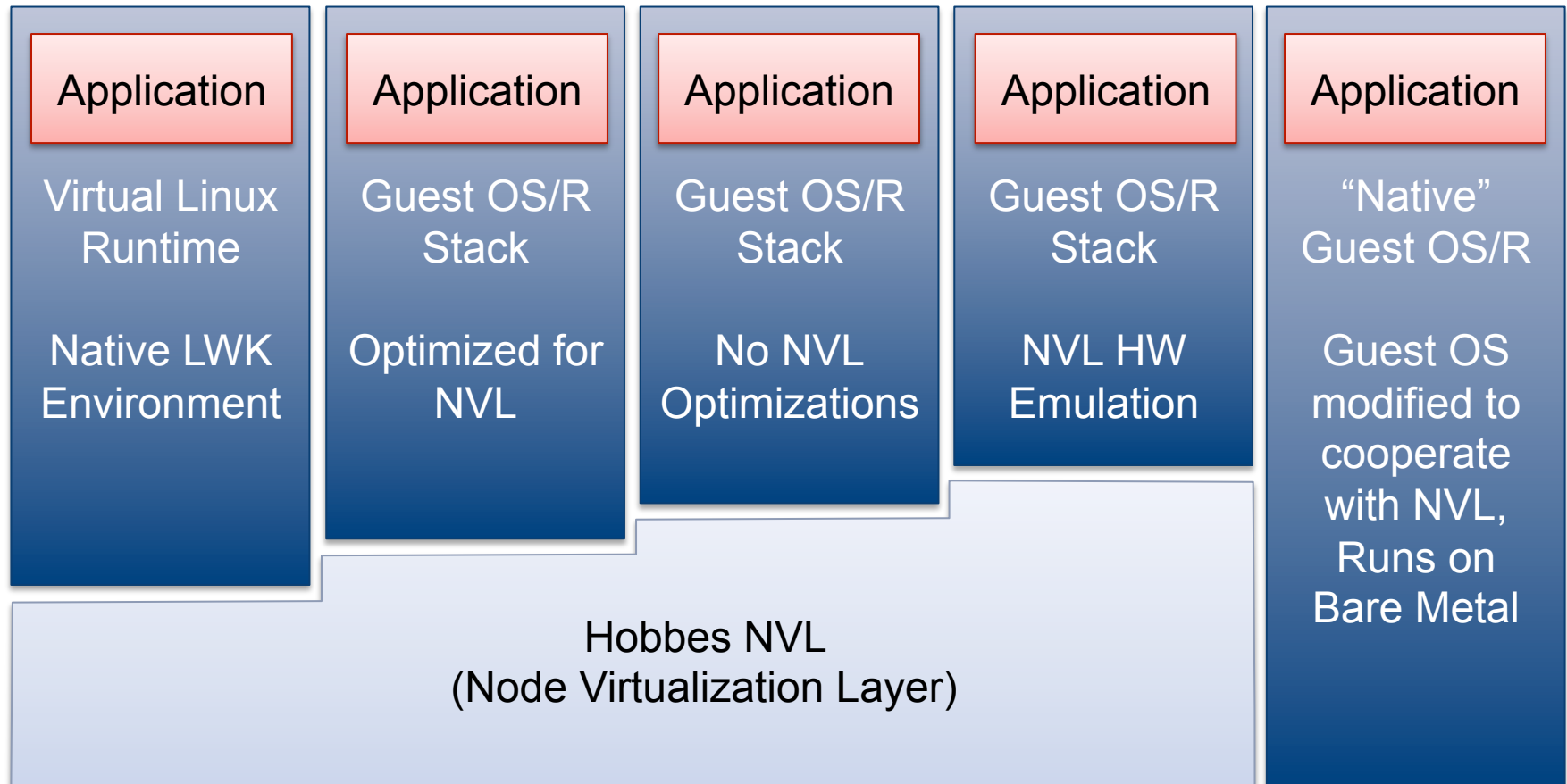
- Example above shows three enclaves, two native and one virtual machine
- Each application component runs in its own enclave, which is a partition of the compute node's resources (CPUs, memory, NICs)
- Approach leads to excellent performance isolation across enclaves
- XEMEM allows user level memory to be shared across enclaves, useful tool for application composition

HPDC'15: "Achieving Performance Isolation with Lightweight Co-Kernels"

HPDC'15: "XEMEM: Efficient Shared Memory for Composed Applications"

Hobbes NVL Has Multiple Levels of Virtualization

- Existing Hypervisors typically support one level, strict isolation
- NVL couples LWK “native” runtime with guest OS/R stacks



Hobbes NVL Provides Composition Mechanisms

