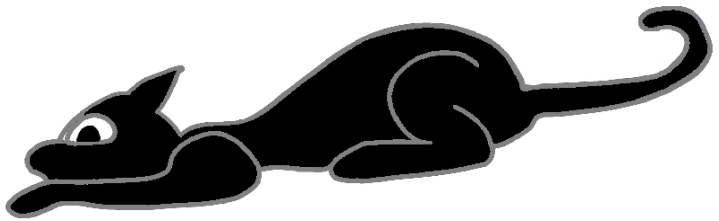
Dual Stack Virtualization: Consolidating HPC and commodity workloads in the cloud

Brian Kocoloski, Jiannan Ouyang,
Jack Lange
University of Pittsburgh



Summary

- Cloud computing holds great promise for HPC
  - Significant interest in scientific computing community
- Problem: HPC applications need HPC environments
  - Tightly coupled, massively parallel, and synchronized
  - Current services must provide dedicated HPC clouds
- Can we host HPC applications in a commodity cloud?
- Dual Stack Approach
  - Provision the underlying software stack along with HPC job
  - Commodity VMM should handle commodity applications
  - HPC VMM (Palacios) can provide HPC environment

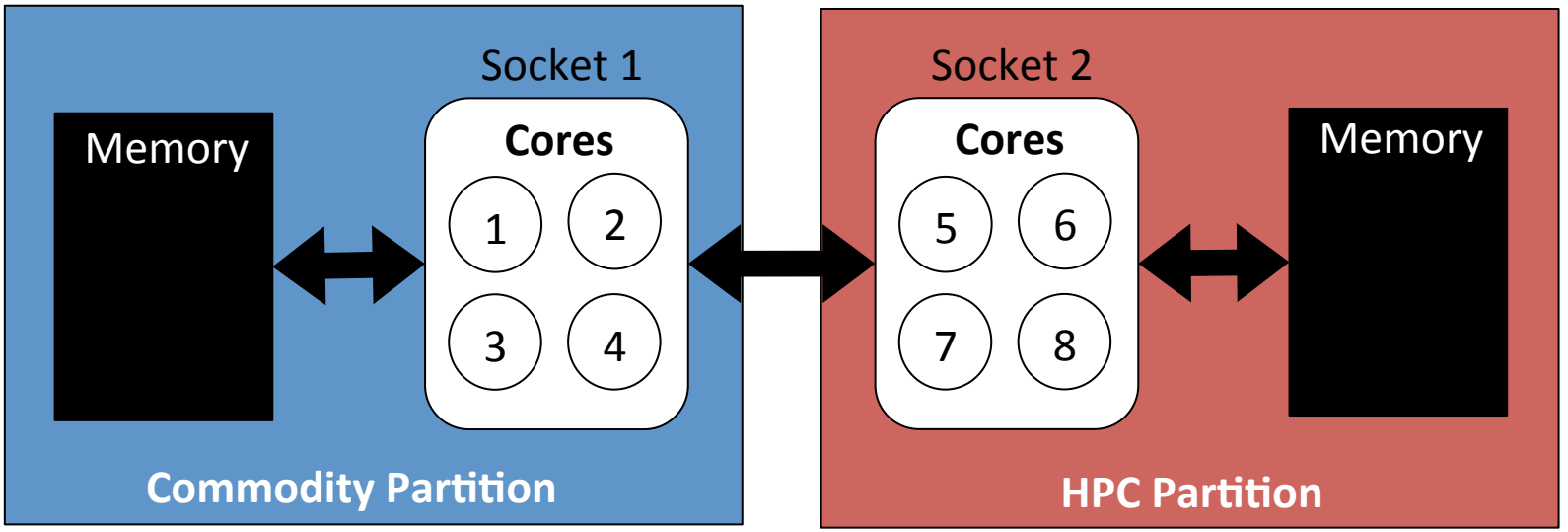
HPC in the cloud

- Clouds are starting to look like supercomputers…
  - Are we seeing a convergence?
- Not yet
  - Noise issues
  - Poor isolation
  - Resource contention
  - Lack of control over topology

- Very bad for tightly coupled parallel apps
  - Require specialized environments that solve these problems

- Approaching convergence
  - Vision: Dynamically partition cloud resources into HPC and commodity zones
  - This talk: partitioning compute nodes with performance isolation

User Space Partitioning



- Current cloud systems do support this, but…
- Interference still exists inside the OS
  - Inherent feature of commodity systems

HPC vs. Commodity Systems

- Commodity systems have fundamentally different focus than HPC systems
  - Amdahl's vs. Gustafson's laws
  - Commodity: Optimized for common case

- HPC: Common case is not good enough
  - At large (tightly coupled) scales, percentiles lose meaning
  - Collective operations must wait for slowest node
  - 1% of nodes can make 99% suffer
  - HPC systems must optimize outliers (worst case)

Commodity VMMs

- Virtualization is considered an "enterprise" technology
  - Designed for commodity environments
  - Fundamentally different, but not wrong!

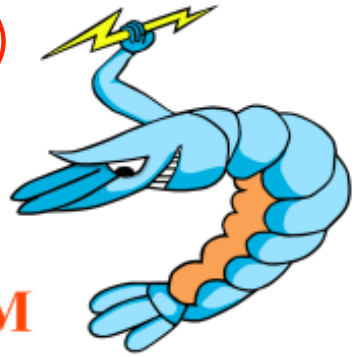
- Example: KVM architecture issues
  - Userspace handlers
  - Fairly complex memory management
  - Locking and periodic optimizations
  - Presence of system noise

Palacios VMM

- OS-independent embeddable virtual machine monitor
  - Established compatibility with Linux, Kitten, and Minix
- Specifically targets HPC applications and environments
  - Consistent performance with very low variance
- Deployable on supercomputers, clusters (Infiniband/Ethernet), and servers
  - 0-3% overhead at large scales (thousands of nodes)
    - VEE 2011, IPDPS 2010, ROSS 2011

Palacios

An OS Independent Embeddable VMM



Open source and freely available

http://www.v3vee.org/palacios

Palacios/Linux

- Palacios/Linux provides lightweight and high performance virtualized environments
  - Internally manages dedicated resources
    - Memory and CPU scheduling
  - Does not bother with "enterprise features"
    - Page sharing/merging, swapping, overcommitting resources

- Palacios enables scalable HPC performance on commodity platforms

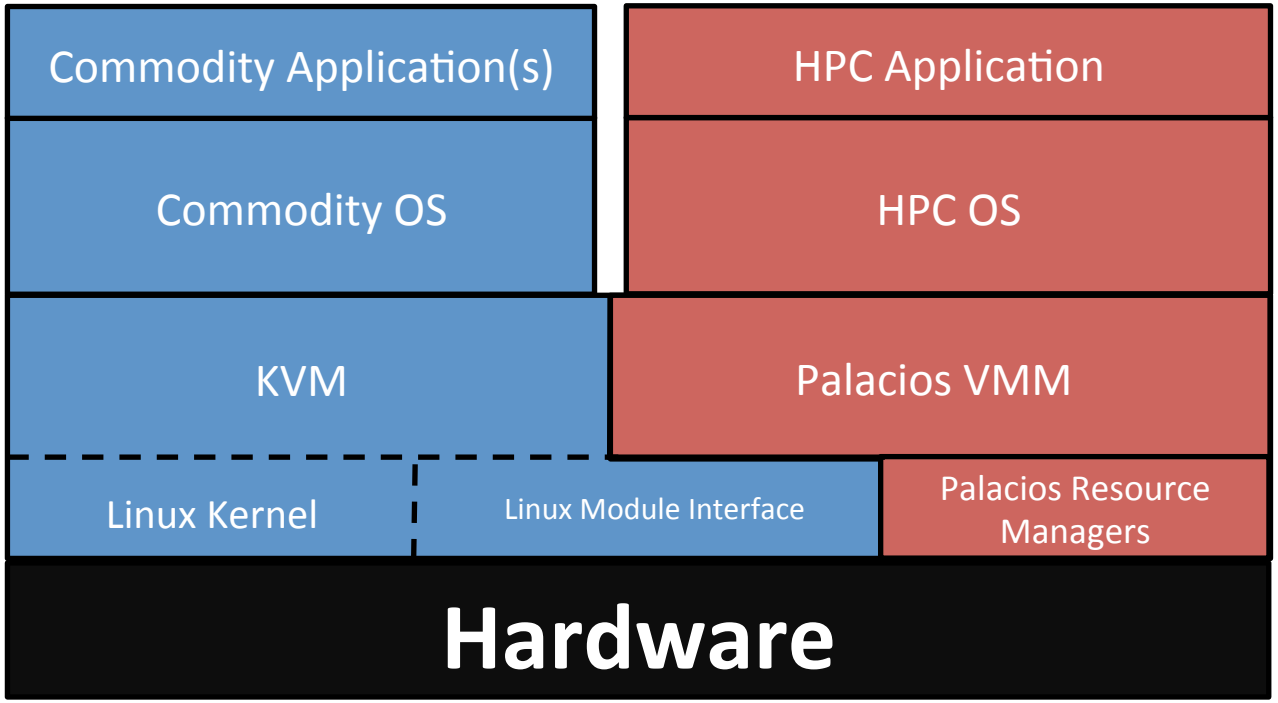
VMM Comparison

- Primary difference: Consistency
  - Requirement for tightly coupled performance at large scale
- Example: KVM nested paging architecture
  - Maintains free page caches to optimize performance
    - Requires cache management
  - Shares page tables to optimize memory usage
    - Requires synchronization

VMM	% of exits	Mean	Std Dev	# NPFS
KVM	52%	8804	5232	3,265,156
Palacios	50%	10876	2685	1,872,017

Dual Stack Architecture

- Partitioning at the OS level

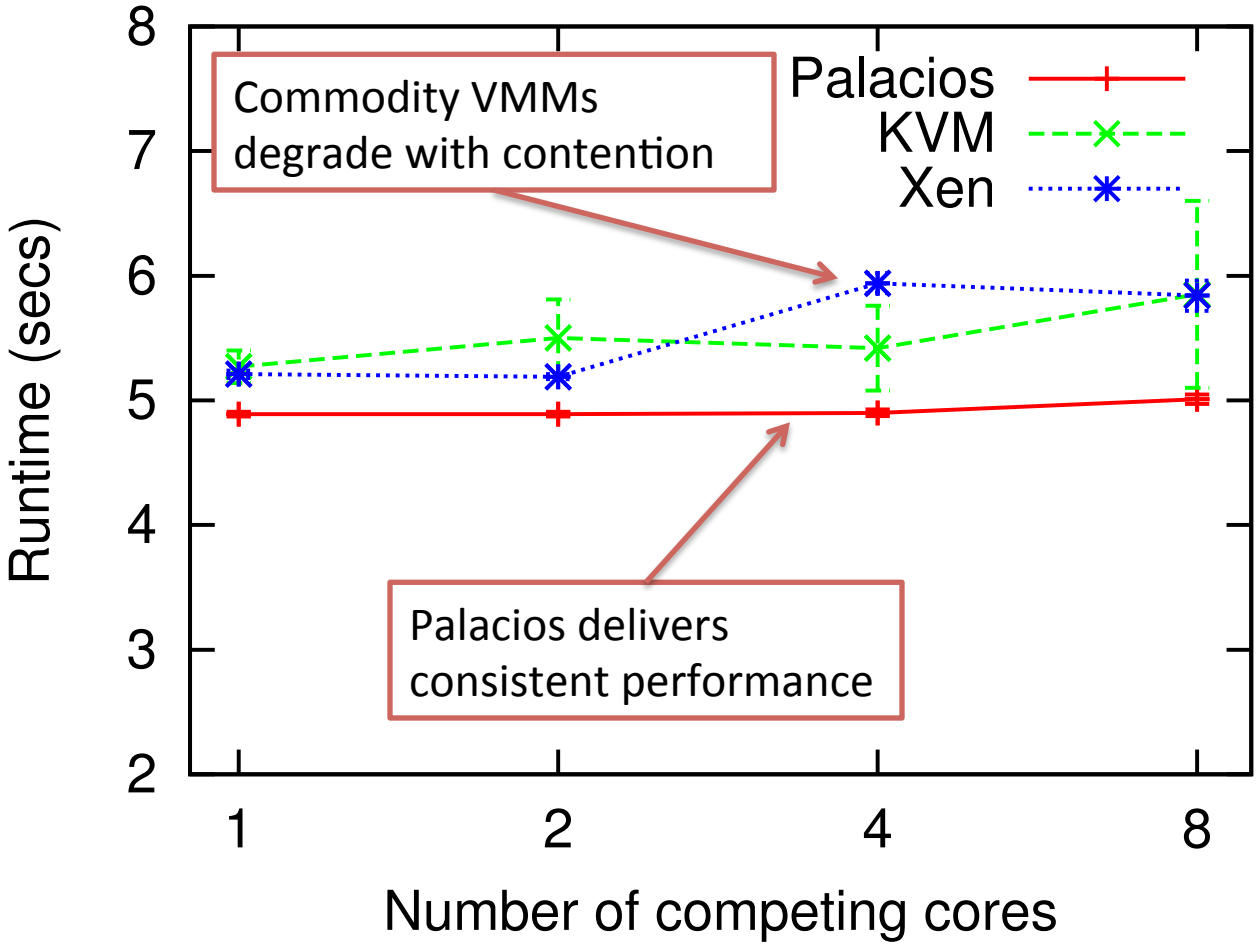


- Enable cloud to host both commodity and HPC apps
  - Each zone optimized for the target applications

Evaluation

- Goal: Measure VM isolation properties
- Partitioned a single node into HPC and commodity zones
  - Commodity Zone: Parallel Kernel compilation
  - HPC Zone: Set of standard HPC benchmarks
  - System:
    - Dual 6-core AMD Opteron with NUMA topology
    - Linux guest environments (HPC and commodity)
- Important: Local node only
  - Does not promise good performance at scale
  - But, poor performance will magnify at large scales

Results



MiniFE: Unstructured implicit finite element solver

Mantevo Project -- https://software.sandia.gov/mantevo/index.html

Discussion

- A dual stack approach can provide HPC environments in commodity systems
  - HPC and commodity workloads can dynamically share resources
  - HPC requirements can be met without fully dedicated resources

- Networking is still an open issue
  - Need mechanisms for isolation and partitioning
  - Need high performance networking architectures
    - 1Gbit is not good enough
    - 10Gbit is good, Infiniband is better
  - Need control over placement and topologies

Conclusion

- The cloud model is transformative for HPC workloads
  - But only if it can meet the demands of HPC users
- Cloud services need to explicitly support HPC workloads
  - Different requirements and behaviors than commodity applications
- A partitioned dual stack approach can get us there
  - Dynamically configured cloud infrastructures for multiple application classes

Thank you

- Jack Lange
  - jacklange@cs.pitt.edu
  - http://www.cs.pitt.edu/~jacklange
- Palacios
  - http://www.v3vee.org/palacios