GPU Passthrough Performance: A Comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications

John Paul Walters[1], Andrew J. Younge[2], Dong-In Kang[1], Ke-Thia Yao[1], Mikyung Kang[1], Stephen P. Crago[1], and Geoffrey C. Fox[2]

[1] USC / Information Sciences Institute

{jwalters,dkang,kyao,mkkang,crago}@isi.edu

[2] Indiana University

{ajyounge,gcf}@indiana.edu

USC Viterbi School of Engineering

Information Sciences Institute

Outline

- Motivation
- Background and related work
- GPU performance across hypervisors
- Lessons learned
- Future work

Motivation

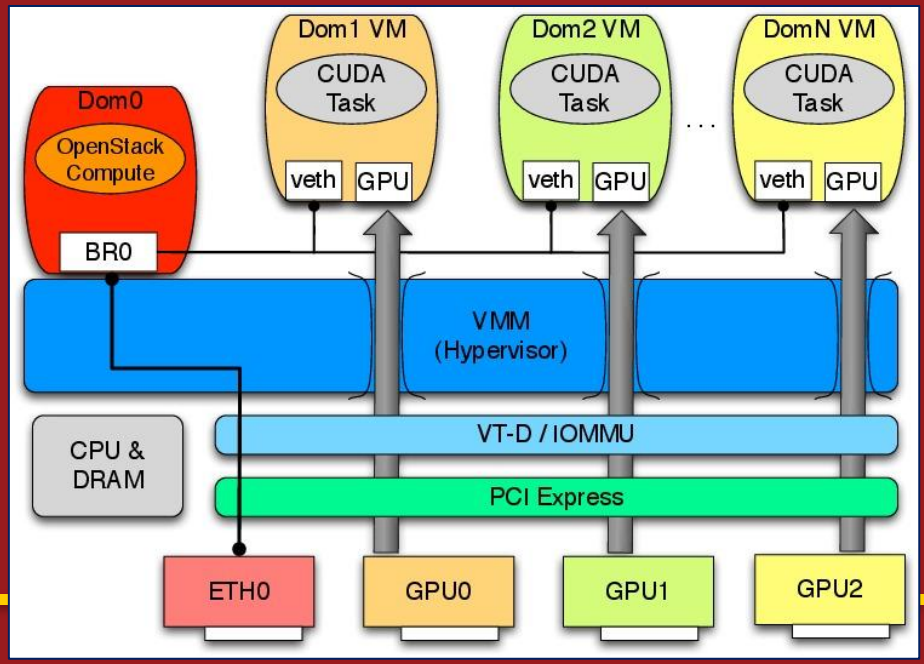
- Scientific workloads demand increasing performance with greater power efficiency
  - Architectures have been driven towards specialization, heterogeneity
- Infrastructure-as-a-Service (IaaS) clouds can democratize access to the latest, most powerful accelerators
  - Most of today's clouds homogeneous
  - Of the major providers, only Amazon offers virtual machine access to GPUs in the public cloud

Background and Related Work

- Options for GPU access in the cloud:
  - API remoting
    - Split driver into front-end/back-end
    - Interpose CUDA library and execute on behalf of the VM
      - Useful for sharing GPUs across networks
      - Performance implications for IO-sensitive applications
    - rCUDA, vCUDA, GViM, and gVirtus are examples
  - PCI Passthrough
    - Use host's IOMMU to pass disassociate physical device from host and attach to the guest
      - Advantage in performance
      - Disadvantage in flexibility (GPU bound to a single VM only)

Cloud Computing and GPUs

- GPU passthrough has historically been hard
  - Specific to particular GPUs, hypervisors, host OS
  - Legacy VGA BIOS support, etc.
- Today we can access GPUs through most of the major hypervisors
  - KVM, VMWare ESXi, Xen, LXC
- What are the performance implications?
- What are the lessons learned?



Experimental Methodology

- Benchmarks
  - Microbenchmarks: SHOC OpenCL
  - Application benchmarks:
    - LAMMPS: measures hybrid multicore CPU + GPU
    - GPU-LIBSVM: characteristic of big data applications
    - LULESH: hydrodynamics exascale proxy application
- Platforms
  - Westmere with Fermi C2075
  - Sandy Bridge with Kepler K20m
- Virtual machines: CentOS 6.4 with 2.6.32-358.23.2 kernel, 20 GB RAM, and 1 CPU socket
  - Control for NUMA effects

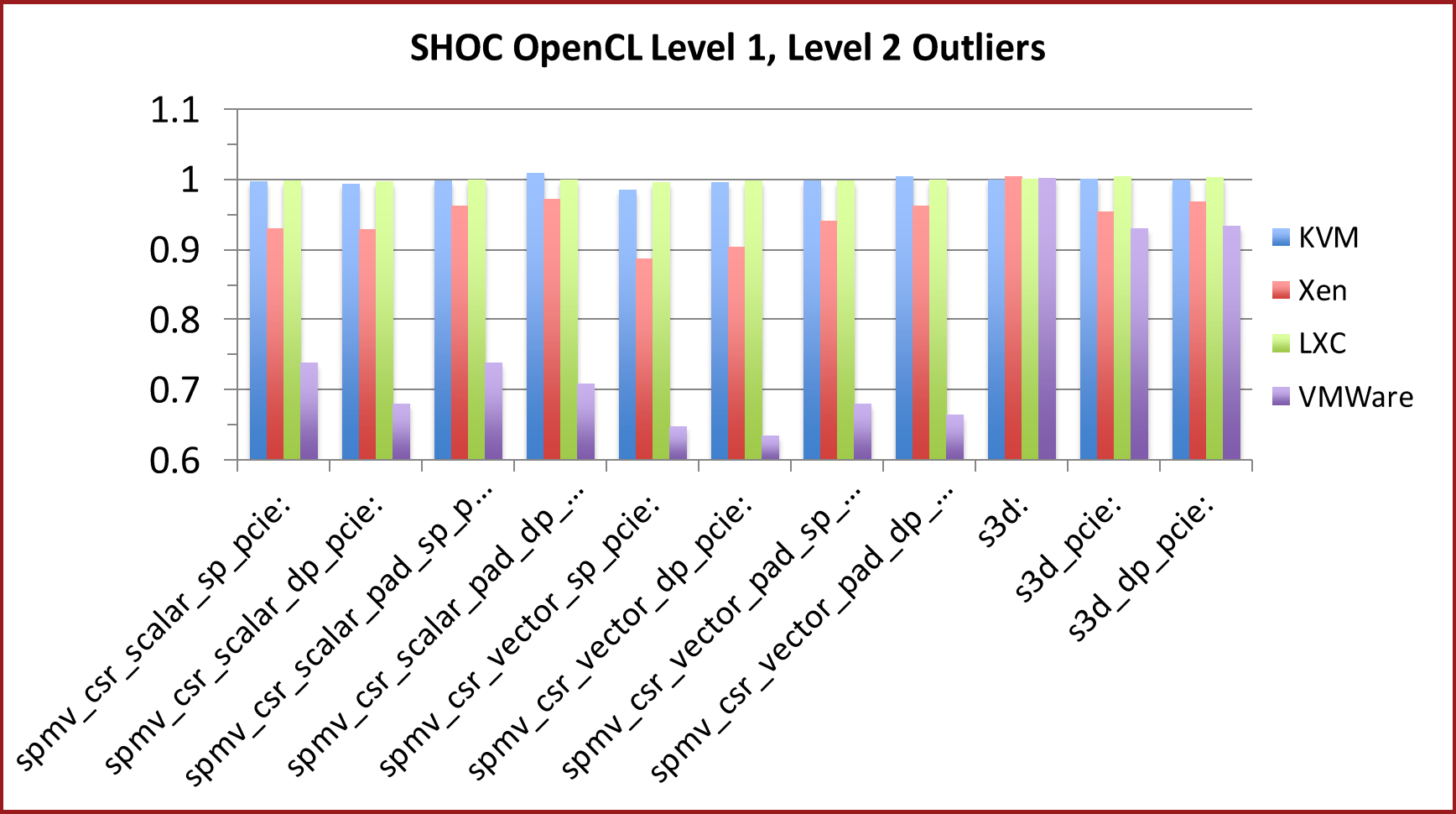
Hardware Setup

	Westmere + Fermi	Sandy Bridge + Kepler
CPU (cores)	2xX5660 (12)	2xE5-2670 (16)
Clock Speed	2.6 GHz	2.6 GHz
RAM	192 GB	48 GB
NUMA Nodes	2	2
GPU	2xC2075	1xK20m
PCI-Express	2.0	3.0
Release	~2011	~2013

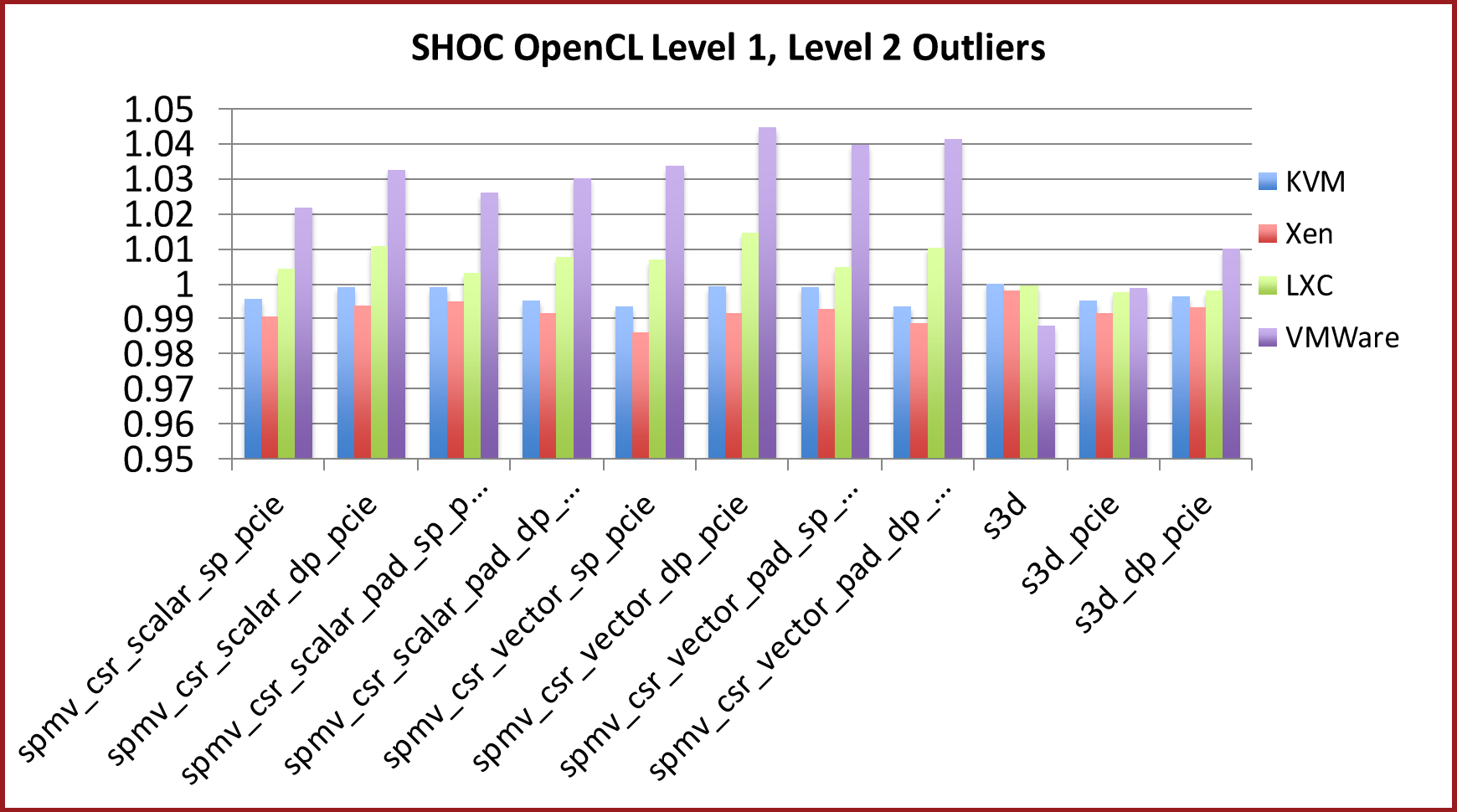
Hypervisor Configuration

Hypervisor	Linux Kernel	Linux Distro
KVM	3.12	Arch 2013.10.01
Xen 4.3.0-7	3.12 (dom0)	Arch 2013.10.01
VMWare ESXi 5.5.0	N/A	N/A
LXC	2.6.32-358.23.2	CentOS 6.4

C2075 Results – SHOC Outliers



K20 Results – SHOC Outliers

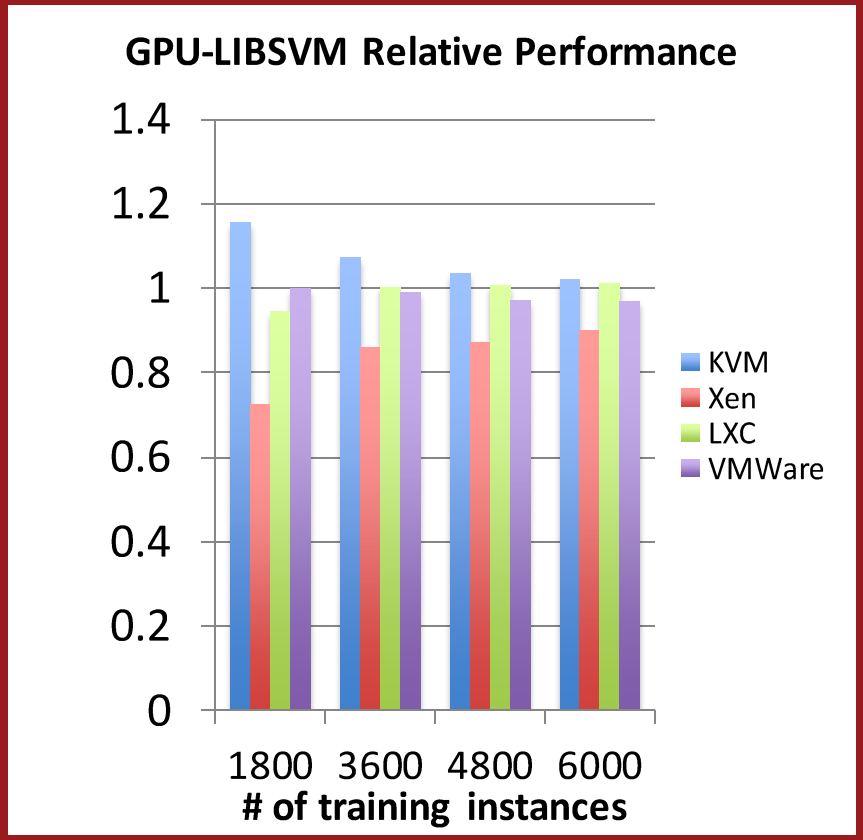


SHOC Observations

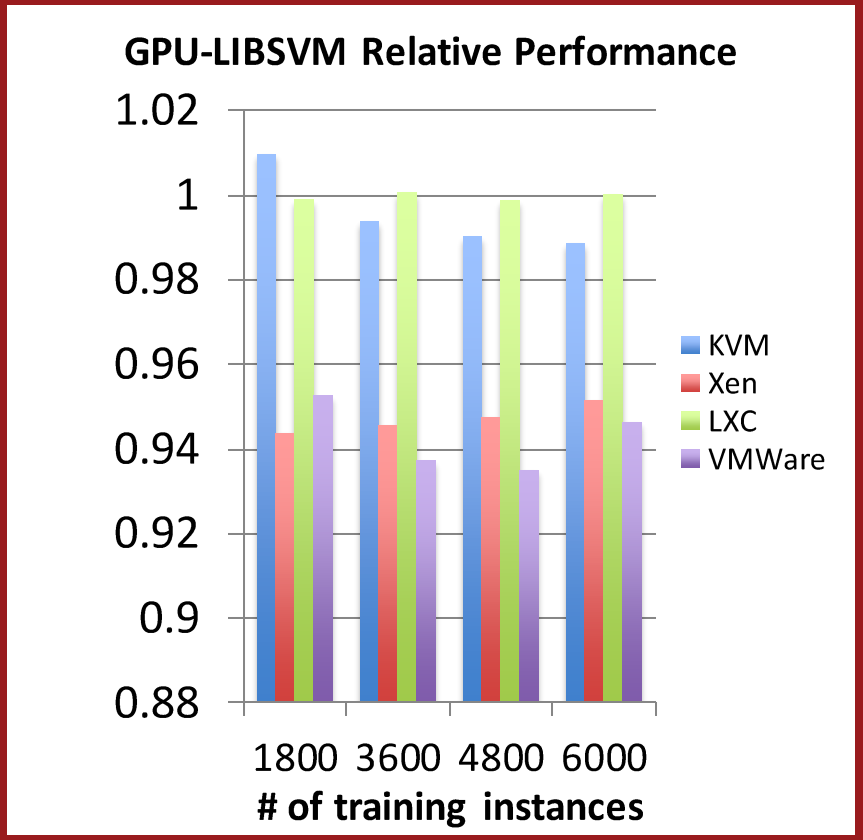
- Overall both Fermi and Kepler systems perform near-native
  - KVM and LXC showed no notable performance degredation
- Xen on the C2075 system shows some overhead
  - Likely because Xen couldn't activate large page tables
- VMWare results vary significantly between architecture
  - Kepler shows greater than base performance
  - Fermi shows largest overhead
- Some unexpected performance improvement for Kepler Spmv

GPU-LIBSVM Results

C2075 Results



K20m Results

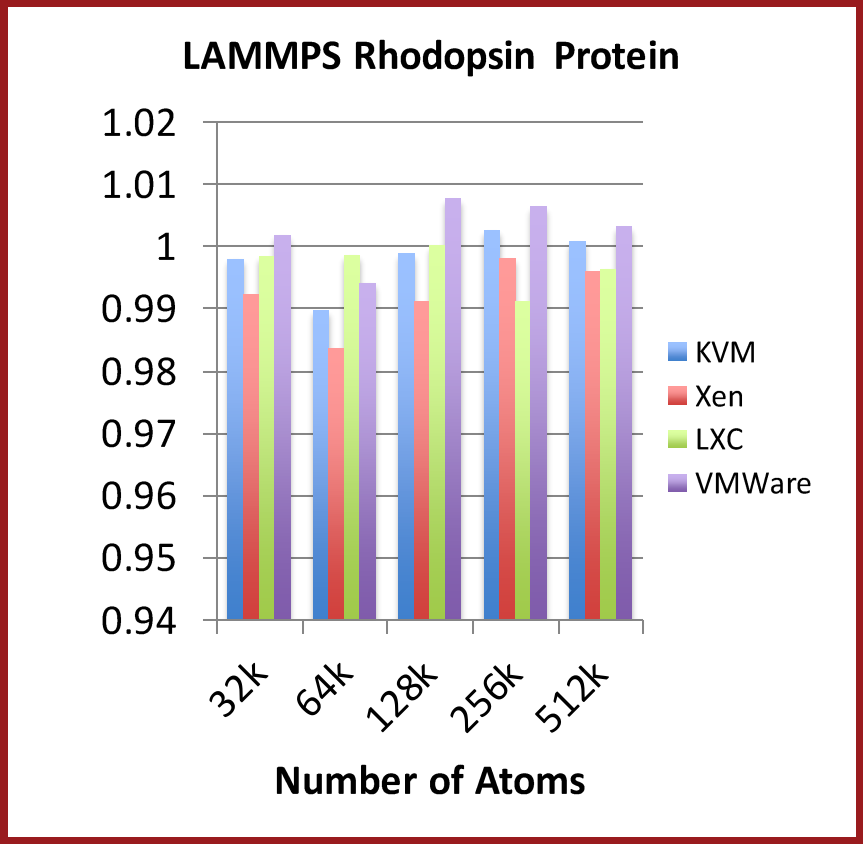


GPU-LIBSVM Observations

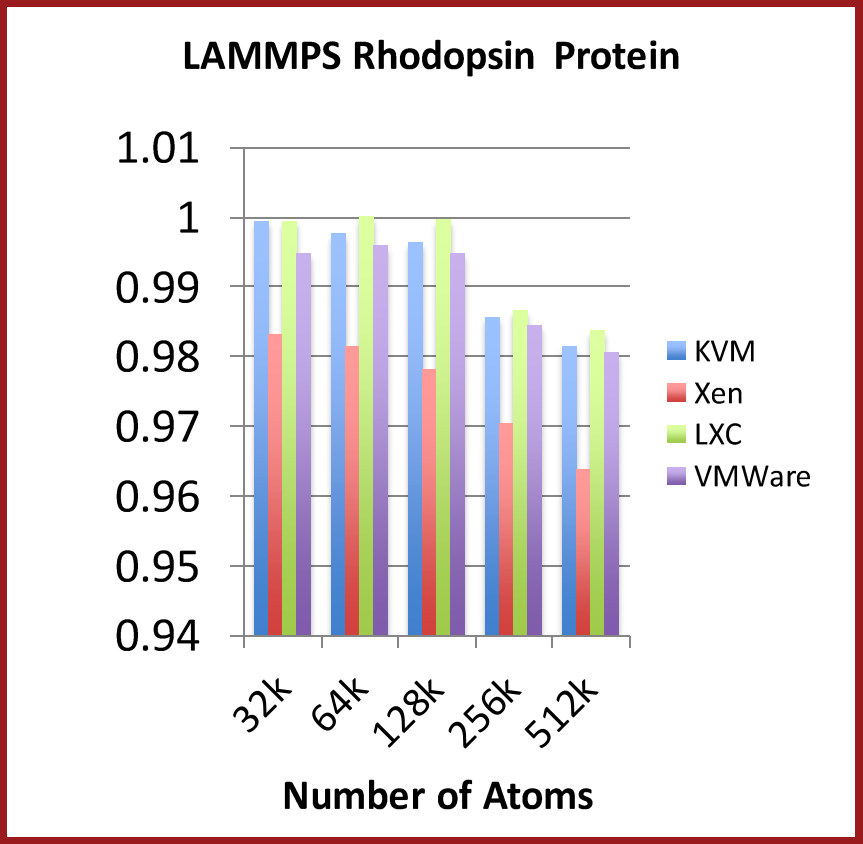
- Unexpected performance improvement for KVM on both systems
  - Most pronounced on Westmere/Fermi platform
- This is likely due to the use of transparent hugepages (THP)
  - Back the entire guest memory with hugepages
  - Improves TLB performance
  - More investigation needed to confirm

LAMMPS Rhodopsin Protein Results

C2075 Results



K20m Results

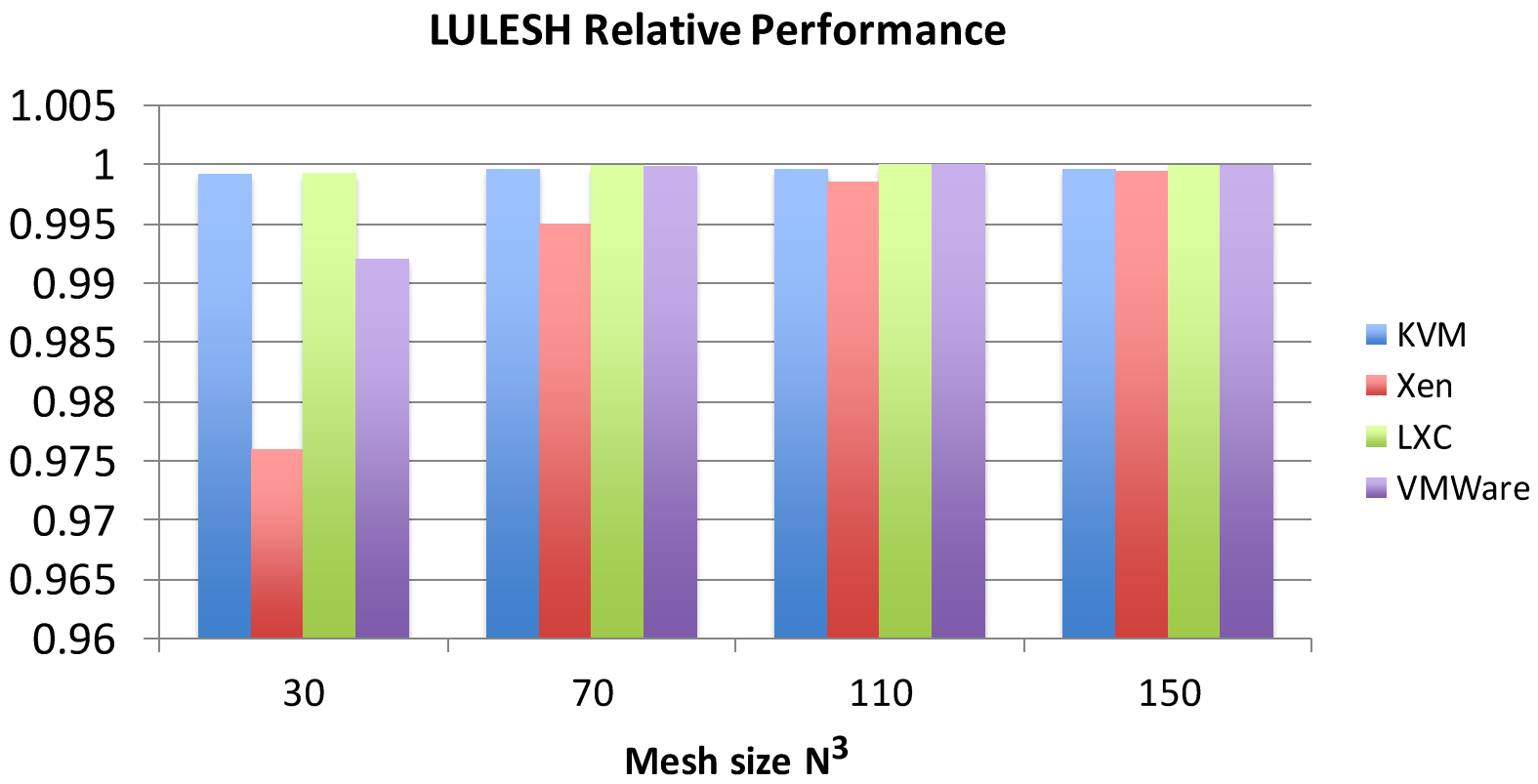


LAMMPS Observations

- Unique among the benchmarks
  - Exercises multiple CPU cores in addition to GPU
- Demonstrates high efficiency across both platforms
  - Unexpectedly higher efficiency for Westmere architecture
- Implications for heterogeneous workloads:
  - SMP CPU + GPU efficiency remains high
  - More research needed to generalize results

LULESH Hydrodynamics Performance

K20m Results



LULESH Observations

- LULESH (K20m only)
- Highly compute-intensive, little data movement
  - Expect little virtualization overhead
- Initially slight overhead from Xen
  - Decreases as mesh resolution (N^3) increases

Lessons Learned – Virtualized HPC

- Virtualization of high performance workloads historically controversial
  - Westmere results suggest this *was* sometimes legitimate
  - More than 10% overhead common
- Recent architectures (e.g. Sandy Bridge) have nearly erased those overheads
  - Lowest performing hypervisor (Xen) within 95% of native
  - Improved CPU integration with PCI-Express bus

Time to reconsider old arguments against virtualized HPC

Lessons Learned – Hypervisor Performance

- KVM consistently yields near-native performance across architectures
- VMWare's performance inconsistent
  - Near-native on Sandy Bridge, high overhead on Westmere
- Xen performed consistently average across both architectures
- LXC performed closest to native
  - Unsurprising, given LXC's design
  - Trades performance for flexibility
- Given these results we see KVM as holding a slight edge for GPU passthrough

Future Work

- Evaluate new hardware when available
- Combine with SR-IOV Interconnect work
  - Test multi-node GPU + InfiniBand deployment
  - Initial results promising
- Introduce mechanisms into Cloud Infrastructure
- NUMA-friendly cloud scheduling