Evolution of Benchmarking on SDSC Systems

Mahidhar Tatineni, San Diego Supercomputer Center

Benchmarking in the Data Center Workshop. PPoPP 2020

February 22, 2020

SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

SDSC is one of the original NSF-funded Supercomputer Centers

- Established 1985 as one of original NSF-funded supercomputer centers
- Transitioned from General Atomics to UCSD in 1997
- 225 staff with diverse computational, HPC, operational and applications backgrounds.
- 12,000 sq. ft. main data center, 5,000 sq. ft. secure data center (FISMA Moderate security level)
- Currently run 4 major supercomputers (Comet, GordonS, Popeye, Triton Cluster)
- ~15 petabytes capacity high performance storage systems

Benchmarking at SDSC

- Benchmarking is a significant part of system acquisition, acceptance, and production activities at SDSC.
- NSF requests for proposals typically have explicit benchmarking requirements. The most recent NSF solicitation* notes:

> ***"A successful proposal must clearly demonstrate how the proposed resource will support transformative discoveries in S&E. This may be done through a combination of analytical models projecting the anticipated performance of the proposed resource, appropriate benchmark results, and compelling empirical evidence validating that the resource will be a valuable scientific instrument for S&E discovery"***

- System designs are based on workload analysis from existing and past systems complemented by expected workload changes.
- System acceptance tests are designed based on a combination of microbenchmarks and expected application workload.
- Benchmarks are used in the production environment to monitor system health and to confirm performance after major upgrades.

***https://www.nsf.gov/pubs/2019/nsf19587/nsf19587.htm**

Evolution of Benchmark Suite

- The suite of low-level benchmarks and applications used has evolved with various generations of machines deployed at SDSC.

- Innovative hardware and software solutions lead to new benchmarking needs. Examples include
  - Large scale deployment of SSD hardware – started with the Trestles and Gordon supercomputers and continues till date
  - Filesystem solutions/features such as Lustre, IME, Weka, iSER
  - Software defined servers aggregating resources on Gordon (with vSMP)
  - Innovative HPC interconnect design and implementation matching HPC hardware to optimal application performance
  - Accelerator options such GPUs
  - Virtualization and containerization solutions on Comet

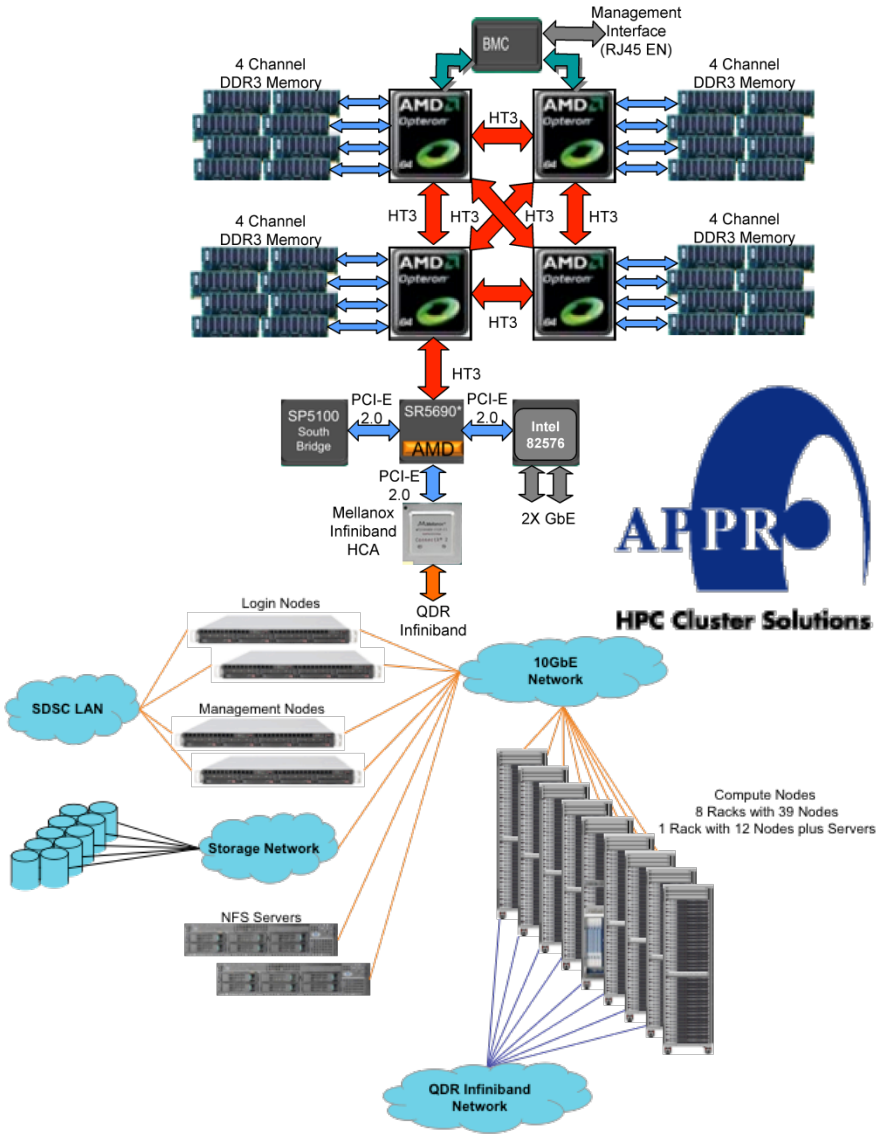
- Diversification of scientific domains using supercomputing resources leads to a rapid growth in software applications base. Examples:
  - Large growth in bioinformatics software stack
  - Increasing use of machine learning approaches in scientific research

NSF Funded HPC Systems at SDSC
Increasing diversity in features

- **Trestles** – A high-productivity HPC system targeted at modest-scale and gateway users *2011-2014*
  - Fat Tree QDR IB network
  - SSD based storage on compute nodes
  - 50 GB/s Lustre filesystem (ethernet based routed over bridge devices)
- **Gordon** – An innovative data-intensive supercomputer *2012-2017 (NSF)*
  - Dual rail 3-D Torus network with QDR IB
  - SSD storage in IO nodes, iSER based mounts on compute nodes. Total of 300TB of flash storage
  - Aggregation of physical nodes to form software defined servers (vSMP)
  - Lustre filesystems routed via IO nodes (100 GB/s performance)
- **Comet** – HPC for the long tail of science *2015 – 2021*
  - Fat Tree FDR IB network (with 4:1 oversubscription between racks)
  - SSD storage in each compute node
  - 200 GB/s Lustre filesystems
  - GPU nodes (K80, P100)
  - Virtual clusters
  - Containerization support (Singularity)

Trestles - System Description

System Component	Configuration
AMD MAGNY-COURS COMPUTE NODE	
Sockets	4
Cores	32
Clock Speed	2.4 GHz
Flop Speed	307 Gflop/s
Memory capacity	64 GB
Memory bandwidth	171 GB/s
STREAM Triad bandwidth	100 GB/s
Flash memory (SSD)	120 GB
FULL SYSTEM	
Total compute nodes	324
Total compute cores	10,368
Peak performance	100 Tflop/s
Total memory	20.7 TB
Total memory bandwidth	55.4 TB/s
Total flash memory	39 TB
QDR INFINIBAND INTERCONNECT	
Topology	Fat tree
Link bandwidth	8 GB/s (bidirectional)
Peak bisection bandwidth	5.2 TB/s (bidirectional)
MPI latency	1.3 us
DISK I/O SUBSYSTEM (SDSC's Data Oasis)	
File systems	NFS, Lustre
Storage capacity (usable)	150 TB: Dec 2010 2PB : June 2011 4PB: July 2012
I/O bandwidth	50 GB/s (June 2011)



Trestles Benchmarks

- Single node (32 cores) performance characterized by EP-DGEMM (~283 Gflops achieved), EP-Stream Triad benchmarks (108 GB/s) in HPCC benchmark.
- Interconnect performance tested using average ping pong latency (1.6μs) and bandwidth (~2.75 GB/s) from HPCC benchmark.
- Full system HPL run (~67 TF achieved)
- Application benchmarks with OpenMP, MPI, and hybrid cases.

Code & Version	Benchmark	Processes	Threads/Process	Runtime (m:s)
ABySS 1.2.5	Ecoli.k31	8	1	8:25
SOAPdenovo 1.05	Ecoli.k31	1	8	3:26
Velvet 1.012	Ecoli.k31	1	1	8:38
MAFFT 6.843	BB30003	1	32	0:17
MrBayes 3.1.2	RDPII_218	8	4	1:42
RAxML 7.2.7	RDPII_218	10	6	6:16
Amber	dhfr	64	1	6:42
NAMD	ApoA1	64	1	3:49

Gordon – An Innovative Data Intensive Supercomputer

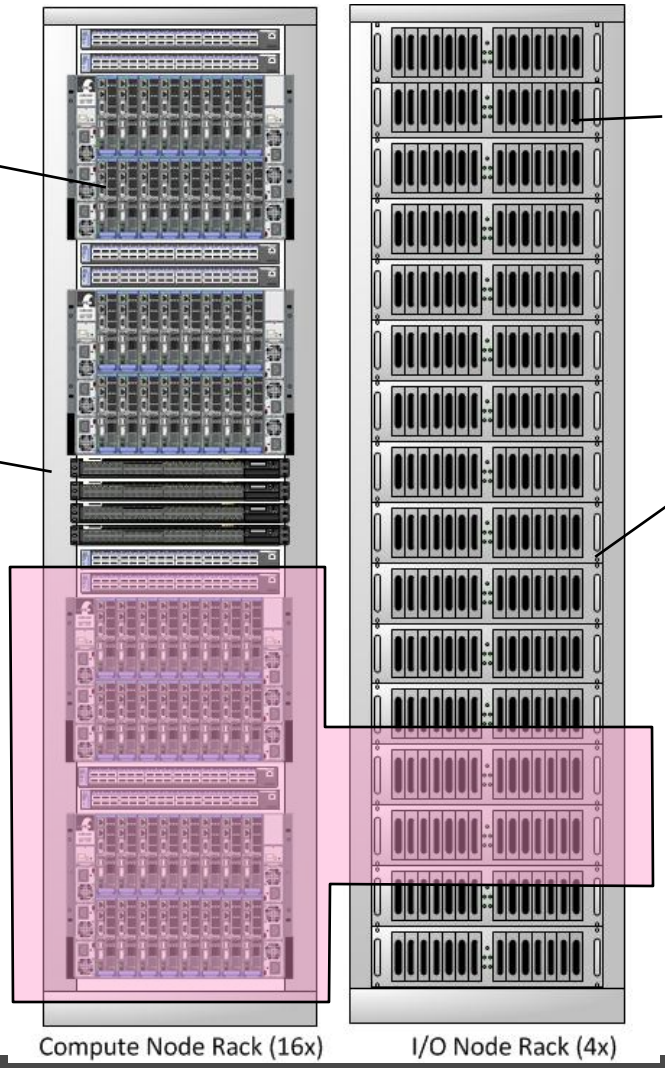
- Designed to accelerate access to massive amounts of data in areas of genomics, earth science, engineering, medicine, and others
- Emphasizes memory and IO over FLOPS.
- Appro integrated 1,024 node Sandy Bridge cluster
- 300 TB of high performance Intel flash
- Large memory supernodes via vSMP Foundation from ScaleMP
- 3D torus interconnect from Mellanox
- Funded by the NSF and available through the NSF Extreme Science and Engineering Discovery Environment program (XSEDE)

Gordon Design Highlights

- 1,024 2S Xeon E5 (Sandy Bridge) nodes
- 16 cores, 64 GB/node
- Intel Jefferson Pass mobo
- PCI Gen3

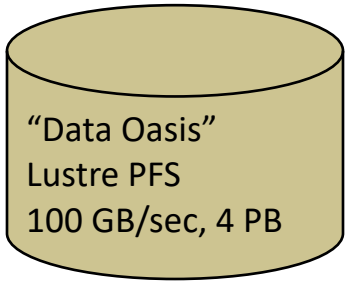
- 3D Torus
- Dual rail QDR

- Large Memory vSMP Supernodes
- 2TB DRAM
- 10 TB Flash

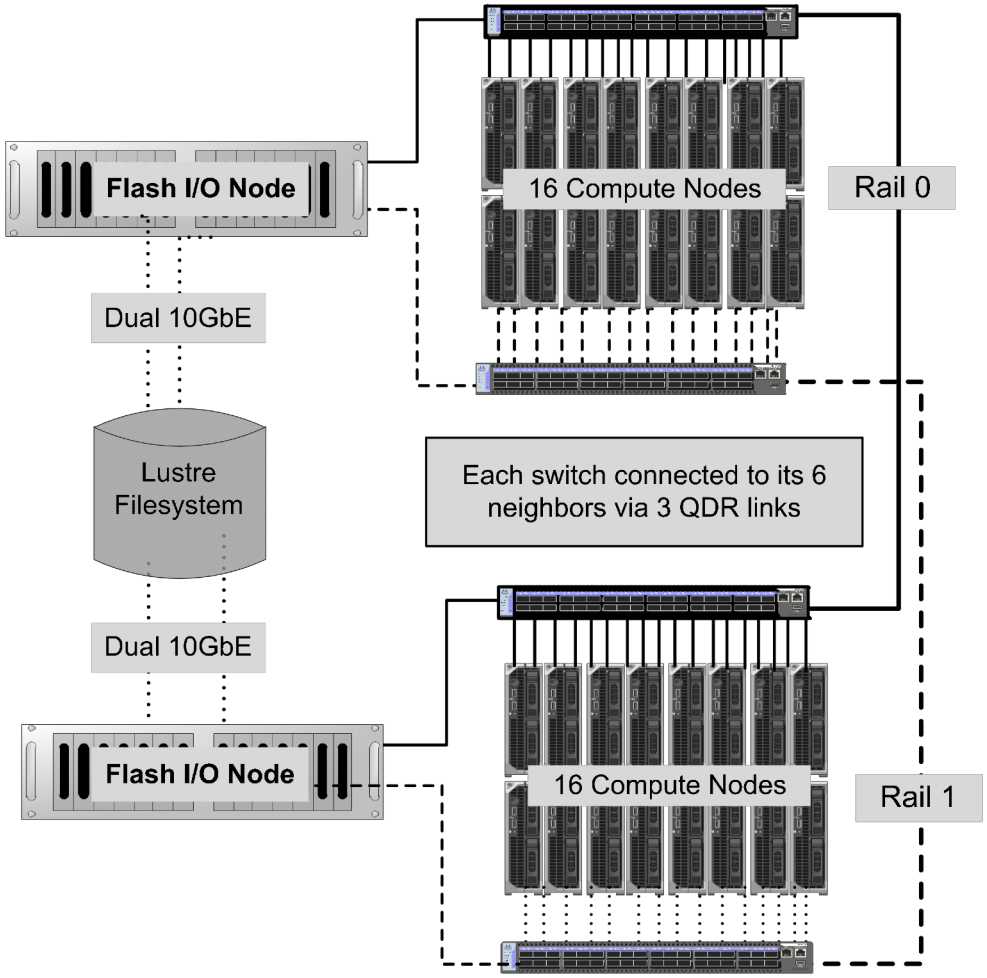


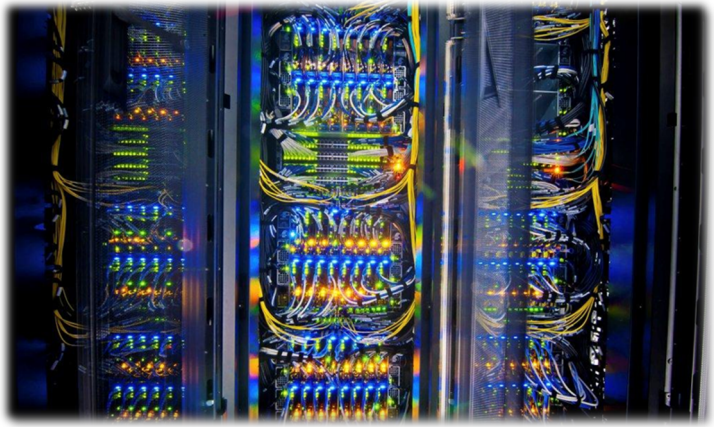
- 300 GB Intel 710 eMLC SSDs
- 300 TB aggregate

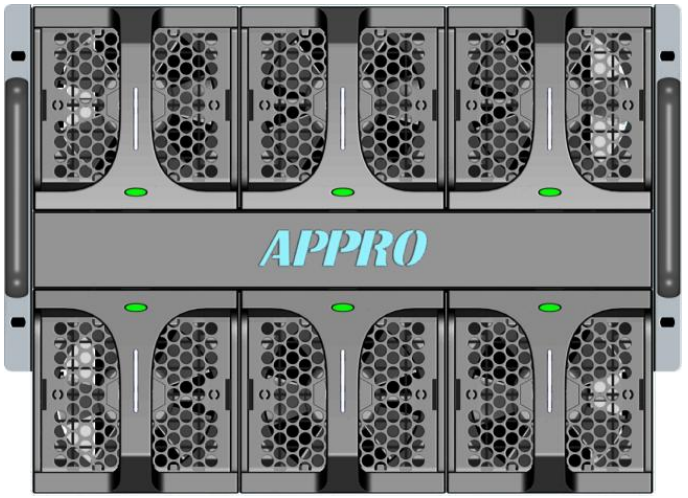
- 64, 2S Westmere I/O nodes
- 12 core, 48 GB/node
- 4 LSI controllers
- 16 SSDs
- Dual 10GbE
- SuperMicro mobo
- PCI Gen2



Subrack and Cabling Design Detail

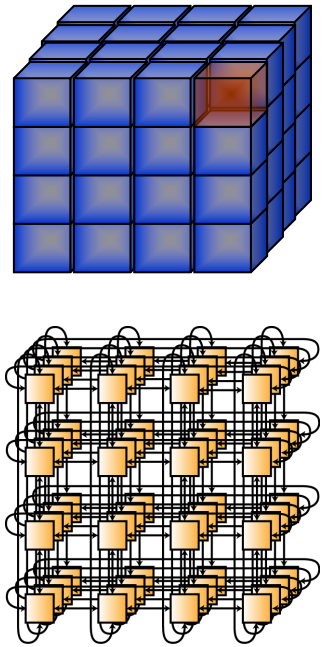




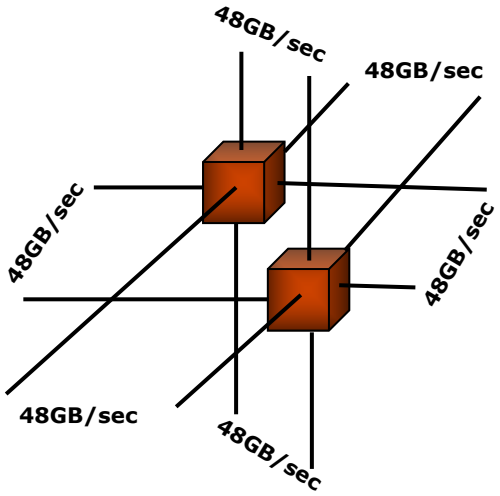


Gordon 3D Torus Interconnect Fabric
4x4x4 3D Torus Topology

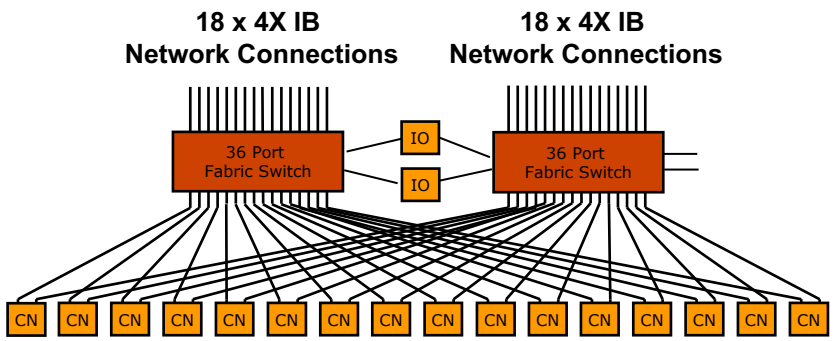
4X4X4 Mesh
Ends are folded on all three Dimensions to form a 3DTorus



Dual-Rail Network
increased Bandwidth & Redundancy



Single Connection to each Network
16 Compute Nodes, 2 IO Nodes



Gordon Benchmarks

- Low-level benchmarks included OSU Benchmarks, IOR, FIO, and HPCC
- HPL at node and full system level
- Application benchmarks (run both natively and in vSMP environment): WRF, GAMESS, PARATEC, MILC, and Enzo
- Interconnect tests have added complexity due to dual rail setup
  - Ping/pong latency, bandwidth for each rail
  - Dual rail performance verification (combined bandwidth)
  - Switch to switch network performance (3 links between neighboring switches)
- IO tests have added complexity due to various flash aggregration, export options, Lustre routing, IO Node connectivity
  - Lustre performance testing
  - Test the IO bandwidth and IO latency from vSMP supernode to aggregated SSD space from IO node
  - IO performance of iSER based filesystem on compute node – includes single drive case (one drive iSER mounted) and aggregated case (all 16 drives in RAID config, mounted via iSER on one node).

Key Performance Metrics (measured)

System	LINPACK	286 Tflop/s (84% of peak)
Compute node	LINPACK	290 Gflop/s (85% of peak)
	Memory bandwidth	67 GB/s (79% of peak)
I/O node	1 SSD sequential performance when mounted on a compute node using XFS	270/210 MB/s r/w (100% of spec)
	16 SSD sequential remote (r/w) with iSER and no file system (using 2 rails)	4.5/4.4 GB/s (exceeds spec)
	16 SSD IOPS local (r/w) 16 SSD IOPS remote (r/w)	600K/120K (exceed spec) 160K/32K
Interconnect	Link bandwidth	Rail 0: 3.9 GB/s Rail 1: 3.4 GB/s
	Latency (for traversing the maximum number of switch hops)	Rail 0: 1.44 μs Rail 1: 2.14 μs
Data Oasis	From a single LNET (I/O node) router	1.6 GB/s write 5 GB/s read
	From 32 LNET routers to 32 Lustre OSS's (half of total system)	50 GB/s

Verifying the 3D Torus
At 4000+ IB cables, not a trivial task

Scripts use output from ibhosts, ibnetdiscover, and ibhosts to verify the topology and generate easily readable files. Abbreviated output from this script is given below for two switches (ib109 and ib110). This verifies that 16 compute nodes and 1 I/O node are on each of the two rails of the torus (**mlx4_1**, and **mlx4_0**).

Switch	x	y	z	Rack	Pos	Name	HCA/xyz	Rack	Pos
ib109-1-3-2	1	3	2	18	6	ion-21-7	mlx4_1	21	7
ib109-1-3-2	**1**	**3**	**2**	**18**	**6**	**gcn-18-51**	**mlx4_1**	**18**	**51**
ib109-1-3-2	1	3	2	18	6	gcn-18-52	mlx4_1	18	52
ib109-1-3-2	1	3	2	18	6	gcn-18-53	mlx4_1	18	53
ib109-1-3-2	1	3	2	18	6	gcn-18-54	mlx4_1	18	54
ib110-1-3-2	1	3	2	18	7	ion-21-7	mlx4_0	21	7
ib110-1-3-2	**1**	**3**	**2**	**18**	**7**	**gcn-18-51**	**mlx4_0**	**18**	**51**
ib110-1-3-2	1	3	2	18	7	gcn-18-52	mlx4_0	18	52
ib110-1-3-2	1	3	2	18	7	gcn-18-53	mlx4_0	18	53
ib110-1-3-2	1	3	2	18	7	gcn-18-54	mlx4_0	18	54

Sampling of Interconnect Benchmarks

Link Latency and Bandwidth

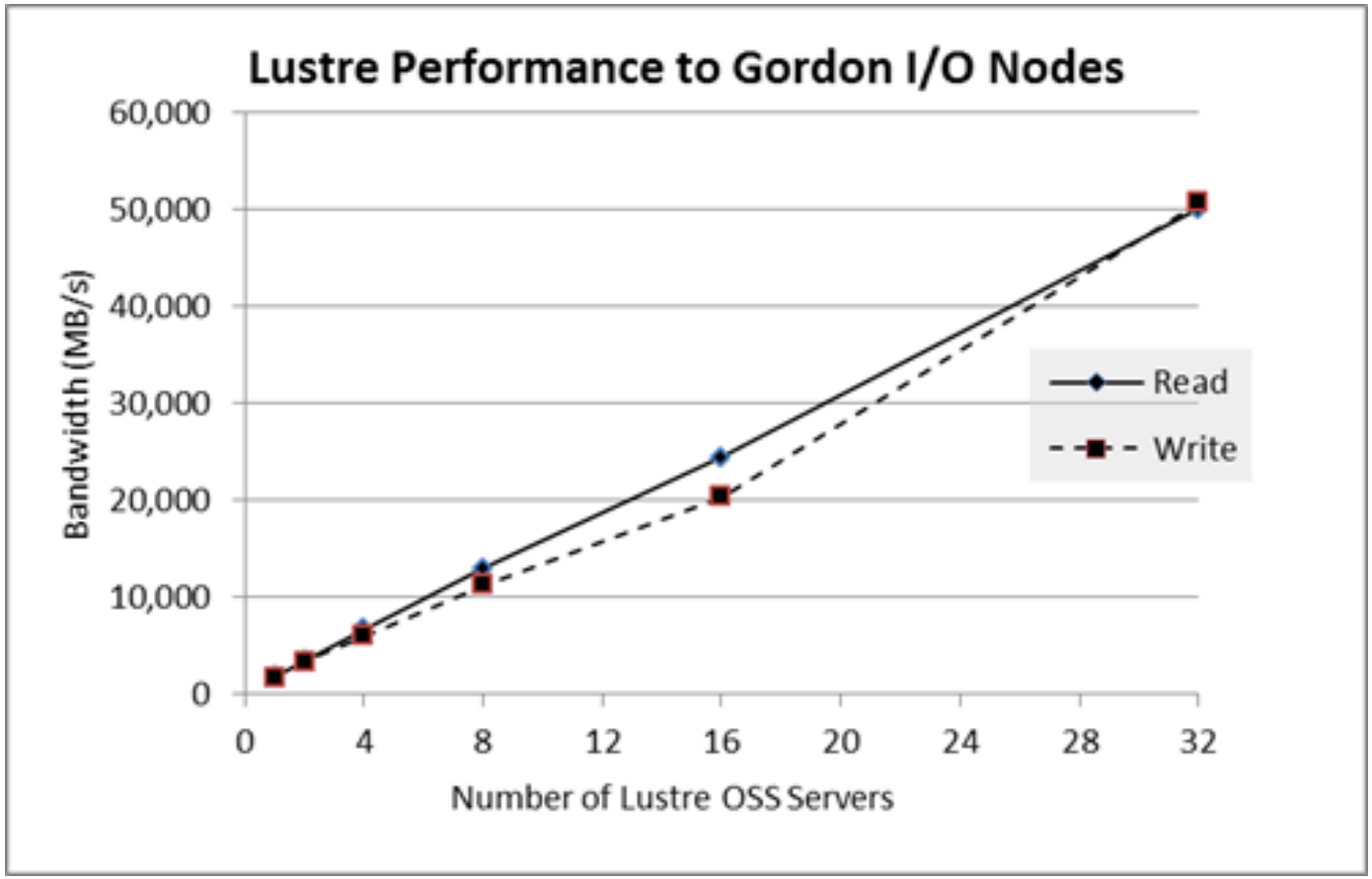
	Rail 0	Rail 1
Latency (μs)	1.44	2.16
Full Duplex Bandwidth (MB/s)	7,515	6,457

Bandwidth Across Max Hops

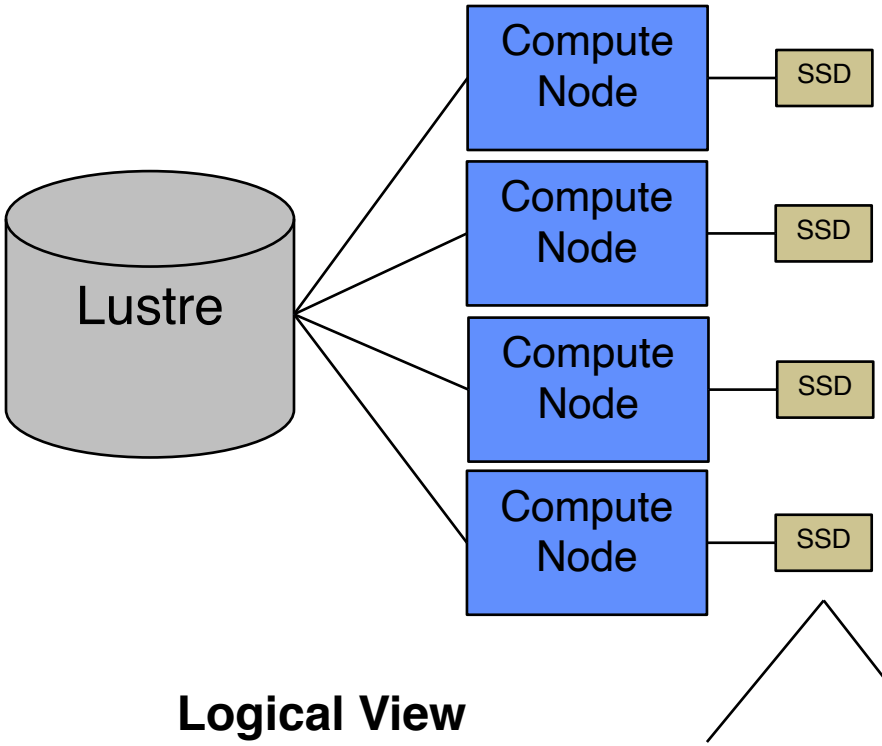
Node #1	Node #2	MB/s
gcn-13-1	gcn-9-44	6,894
gcn-19-22	gcn-3-47	6,896
gcn-7-31	gcn-15-81	6,893

Interswitch links (3x) : 21 GB/s

Data Oasis Performance



Exporting Flash
Model A: One SSD per Compute Node



Logical View

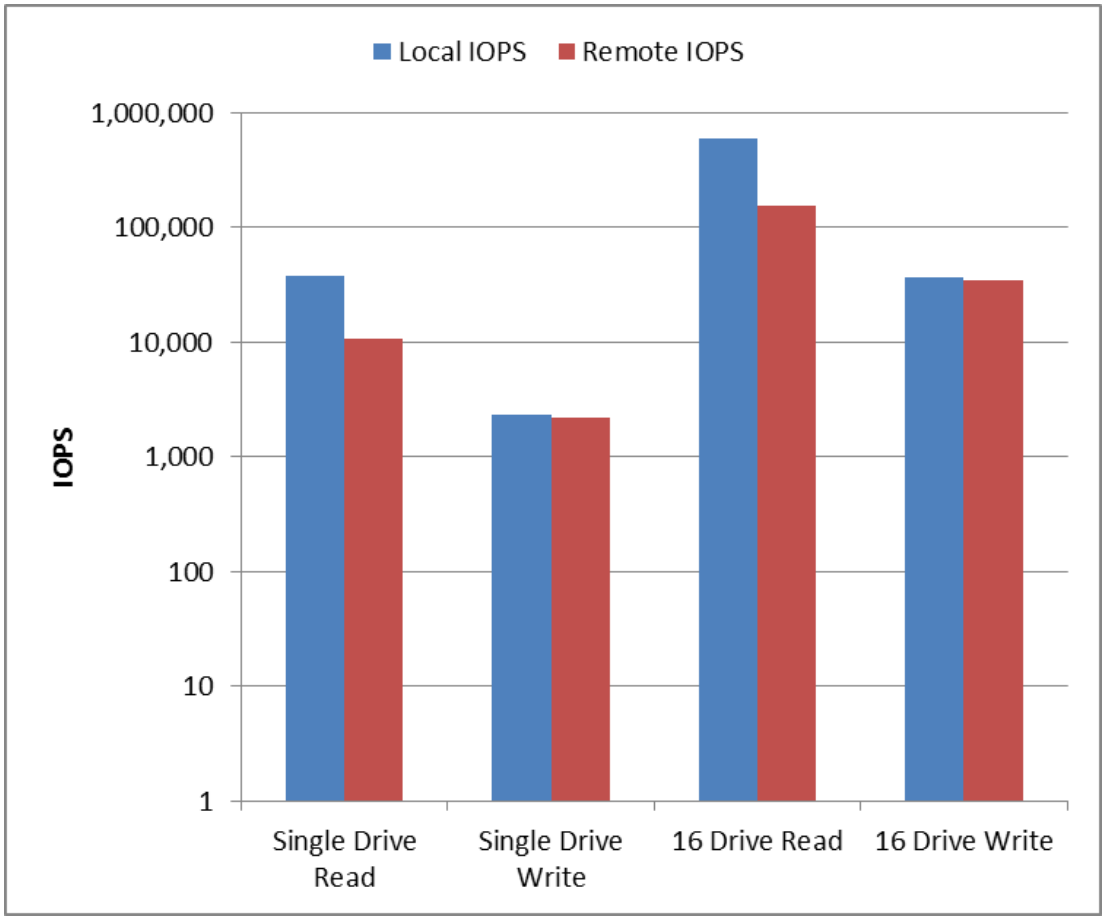
- One 300 GB flash drive exported to each compute node appears as a local file system
- Lustre parallel file system is mounted identically on all nodes.
- Data is purged at the end of the run

Use cases:

- Applications that need local, temporary scratch
- Gaussian
- Abaqus

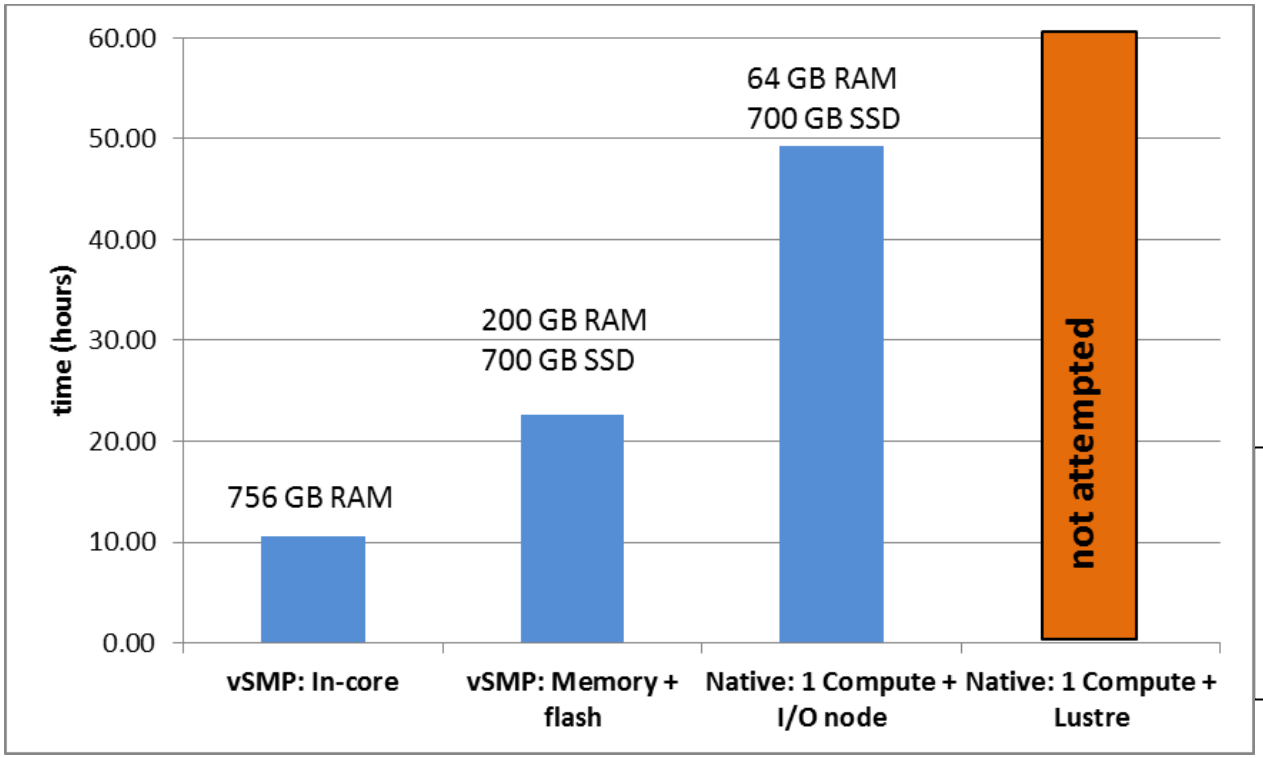
File system appears as:
/scratch/$USER/$PBS_JOBID

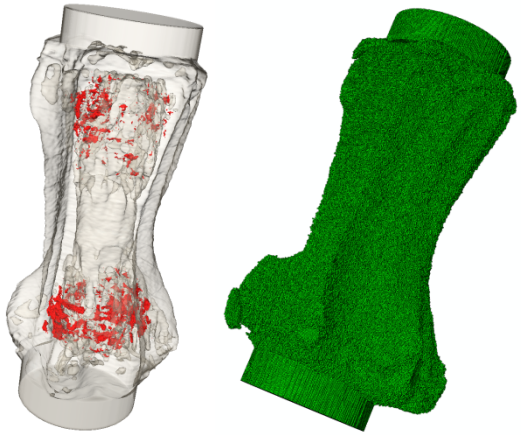
Mounting the I/O nodes with iSER Random Performance



Axial compression of caudal rat vertebra using Abaqus and vSMP

The goal of the simulations is to analyze how small variances in boundary conditions effect high strain regions in the model. The research goal is to understand the response of trabecular bone to mechanical stimuli. This has relevance for paleontologists to infer habitual locomotion of ancient people and animals, and in treatment strategies for populations with fragile bones such as the elderly.





- 5 million quadratic, 8 noded elements
- Model created with custom Matlab application that converts 25^3 micro CT images into voxel-based finite element models

Source: *Matthew Goff, Chris Hernandez. Cornell University. Used by permission. 2012*

Native vs vSMP Application Performance

Application	Problem size	Cores	native time (s)	vSMP time (s)	vSMP/native
MILC	medium	64	134	145	1.08
	large	256	983	1011	1.03
PARATEC	medium	64	278	246	0.88
	large	256	424	474	1.12
WRF	standard	128	409	445	1.09
ENZO	n/a	512	1700	1360	0.80

Comet: System Characteristics

- **Total peak flops ~2.76 PF**
- **Dell primary integrator**
  - *Intel Haswell processors w/ AVX2*
  - *Mellanox FDR InfiniBand*
- **1,944 standard compute nodes (46,656 cores)**
  - *Dual CPUs, each 12-core, 2.5 GHz*
  - *128 GB DDR4 2133 MHz DRAM*
  - *2*160GB GB SSDs (local disk)*
- **72 GPU nodes**
  - *36 nodes with two NVIDIA K80 cards, each with dual Kepler3 GPUs*
  - *36 nodes with 4 P100 GPUs each*
- **4 large-memory nodes**
  - *1.5 TB DDR4 1866 MHz DRAM*
  - *Four Haswell processors/node*
  - *64 cores/node*

- Hybrid fat-tree topology
  - FDR (56 Gbps) InfiniBand
  - Rack-level (72 nodes, 1,728 cores) full bisection bandwidth
  - 4:1 oversubscription cross-rack
- Performance Storage (Aeon)
  - 7.6 PB, 200 GB/s; Lustre
  - Scratch & Persistent Storage segments
- Durable Storage (Aeon)
  - 6 PB, 100 GB/s; Lustre
  - Automatic backups of critical data
- Home directory storage
- Gateway hosting nodes
- Virtual image repository
- 100 Gbps external connectivity to Internet2 & ESNet

Comet: Flexibility to Address Diverse Needs

- **Wide range of hardware options:**
  - Large number of regular compute nodes (**1944**) with **128GB** of memory and **210GB of local flash**.
  - Subset of compute nodes have **1.5TB of local flash**
  - 4 large memory (**1.5TB RAM**) nodes
  - **36 GPU nodes** with 4 GPUs (in 2 K80s) each
- **Flexible Software Environment**
  - **Rich set of applications** (>100) in regular compute environment
  - **Hadoop/Spark capability** can be enabled within regular scheduler environment.
  - Supports **Singularity based containerization** to enable other Linux based environments (for example Ubuntu). Users can upload their own images!
  - Virtual Clusters (VC) – see operational bullet below.
- **Flexible Operations**
  - **Flexible scheduler environment** – shared and exclusive queues, long running jobs, focus on quick turn around time
  - Research Groups/communities, who have people in their group with ***expert system administration skills***, can build their **own virtual clusters** with a custom OS and custom operational setup.

Comet Benchmarks

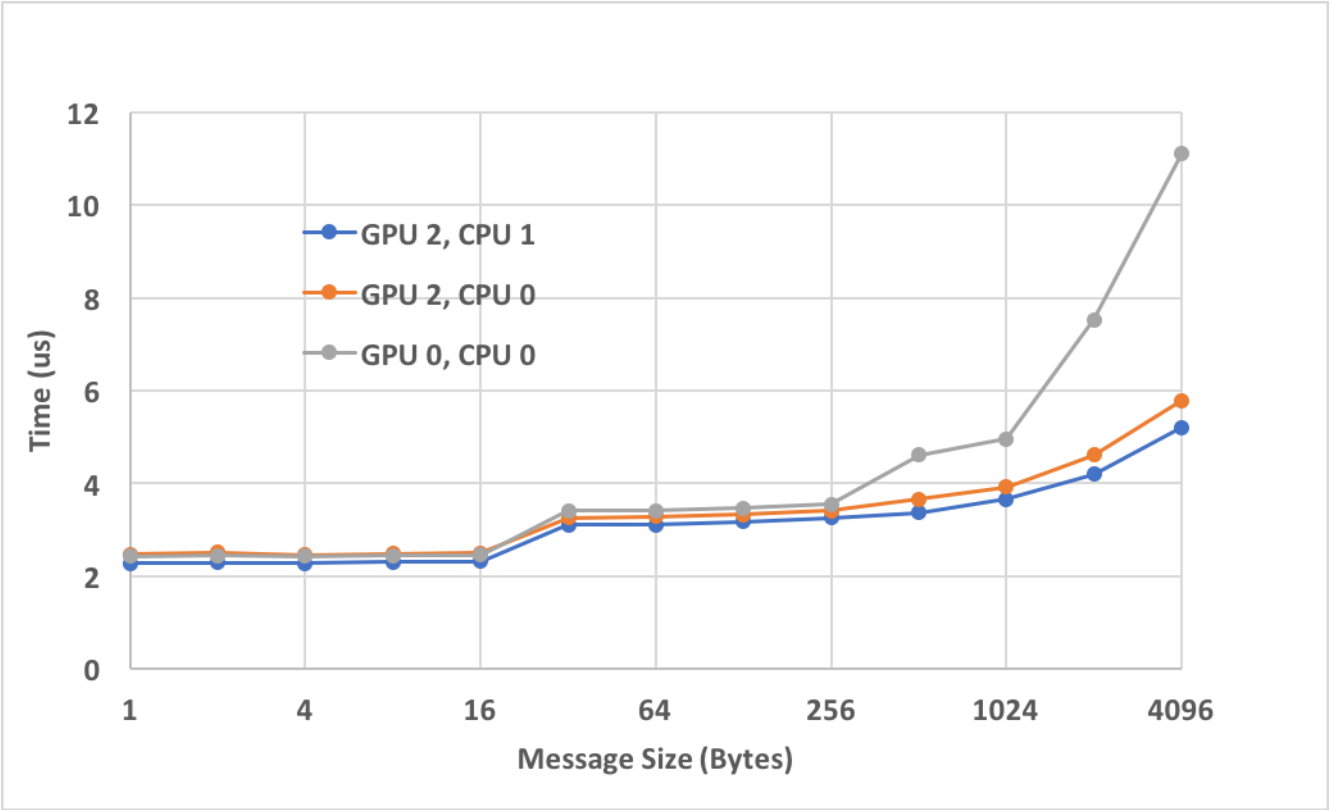
- Low-level benchmarks included STREAM, OSU Benchmarks, IOR, and FIO
- HPL at node, rack, and system level
- Application benchmarks: NEURON, OpenFOAM, Quantum Espresso, RAxML, VisIt, WRF, Abaqus, SOAPdenovo2, and AMBER (GPU)
- Lustre and flash filesystems I/O testing. Lustre tests at node, switch, rack, and maximum bandwidth levels
- Low-level and application benchmarks in virtual cluster (SRIOV enabled) environment
- Performance of applications in containerized (Singularity) environment
- Performance of applications with GPU Direct RDMA (GDR) usage – HOOMD-Blue, Caffe, TensorFlow
- Benchmarks testing RDMA Spark and Hadoop implementations – frameworks are spun up within the regular Comet scheduler environment (SLURM)

MVAPICH2-GDR via Singularity Containers

- **Installed in Singularity Container**
  - NVIDIA driver, CUDA 9.2 (this can alternately be pulled in via the --nv flag)
  - Mellanox OFED stack
  - gdrcopy library - *kernel module is on the host system*.
  - MVAPICH2-GDR (w/o slurm)
  - TensorFlow (conda install)
  - Horovod (pip installed)

- **Job Launch**
  - Use mpirun/mpirun_rsh on the host (external to the image) and wrap the executable/script in Singularity ”exec” command.
  - Launch using mpirun_rsh within the Singularity image.

OSU Latency (osu_latency) Benchmark Inter-node, K80 nodes



- Latency between GPU 2 , process bound to CPU 1 on both nodes: 2.27 μs
- Latency between GPU 2 , process bound to CPU 0 on both nodes: 2.47 μs
- Latency between GPU 0 , process bound to CPU 0 on both nodes: 2.43 μs

TensorFlow Benchmark (tf_cnn_benchmarks)

- **Interactive access to resources using ”srun”**
- **Get an interactive shell in Singularity image environment**

  singularity shell ./centos7mv2gdr.img
- **Run benchmark using hosts (get list from Slurm)**

  export MV2_PATH=/opt/mvapich2/gdr/2.3.2/mcast/no-openacc/cuda9.2/mofed4.5/mpirun/gnu4.8.5

  export MV2_USE_CUDA=1

  export MV2_USE_MCAST=0

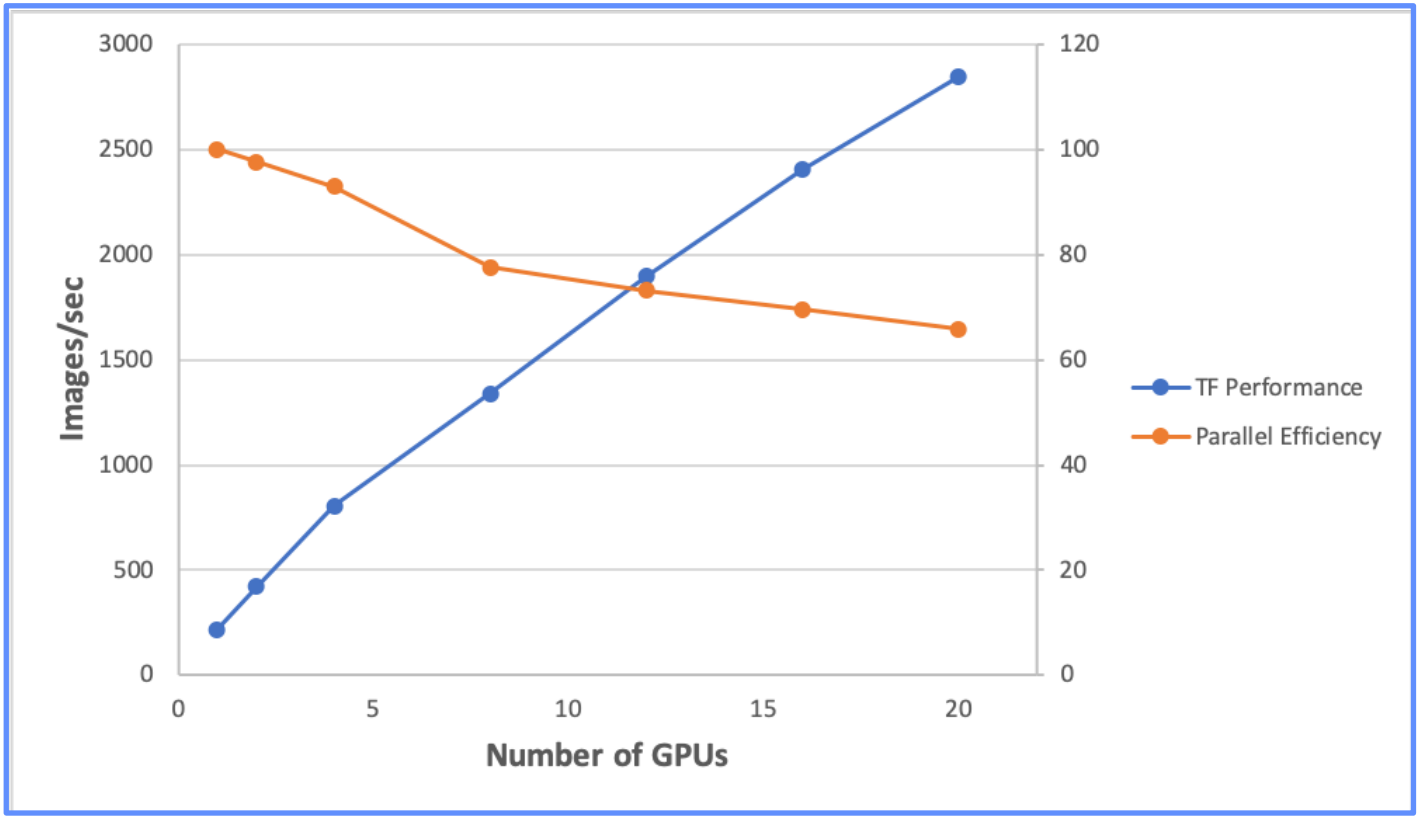
  export MV2_GPUDIRECT_GDRCOPY_LIB=/opt/gdrcopy/lib64/libgdrapi.so

  export CUDA_VISIBLE_DEVICES=0,1

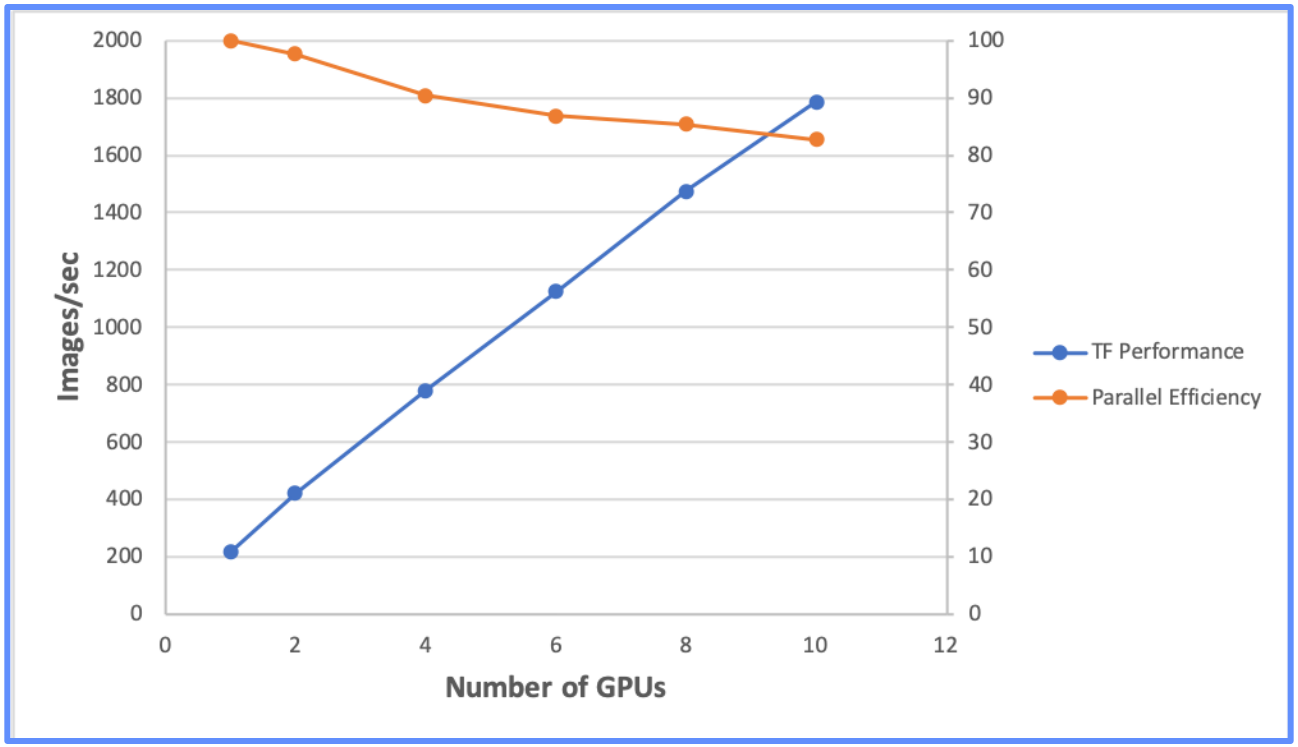
  export MV2_SUPPORT_TENSOR_FLOW=1

  $MV2_PATH/bin/mpirun_rsh -export -np 4 comet-34-16 comet-34-16 comet-34-17 comet-34-17 python tf_cnn_benchmarks.py --model=resnet50 --variable_update=horovod > TF_2NODE_4GPU.txt

TensorFlow Benchmark (GPU 0,1,2,3 on each node)



TensorFlow Benchmark (GPU 0,1 on each node)

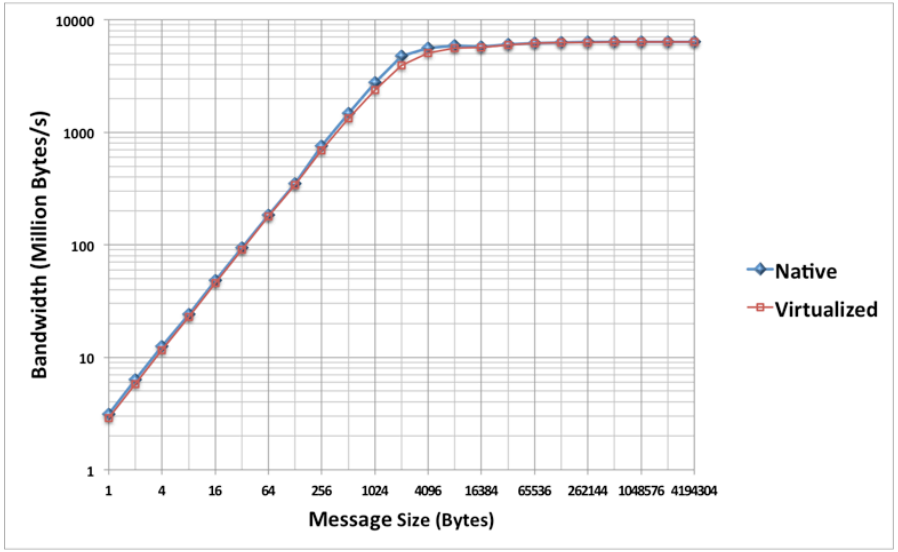


Virtualization on Comet

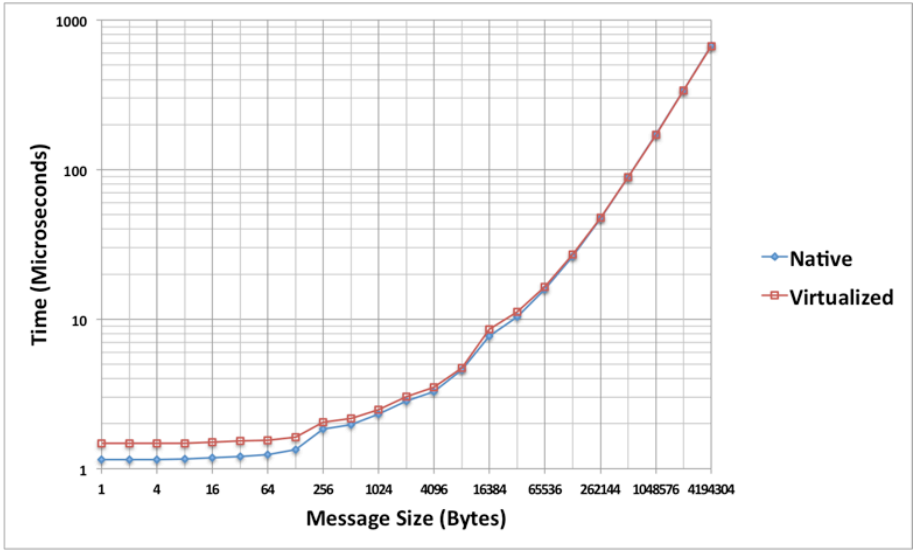
- **Containers using Singularity (http://singularity.lbl.gov)**
  - Migrate complex software stacks from their campus to Comet.
  - Singularity runs in user space and requires very little special support – in fact it actually reduces it in some cases.
  - Applications include: Tensorflow, Torch, Fenics, and custom user applications.
  - Docker images can be imported into Singularity
  - Currently used by ~20 research groups on Comet.
- **Comet Virtual Clusters**
  - KVM based full virtualization with SRIOV support.
  - Full root access, PXE install, persistent disk images, near native InfiniBand
  - Nucleus Rest API and Cloudmesh (Indiana University) management
  - Backends to scheduled jobs consuming XSEDE allocations.

IB Network Performance Comparison Virtual Cluster vs Native

Network Bandwidth

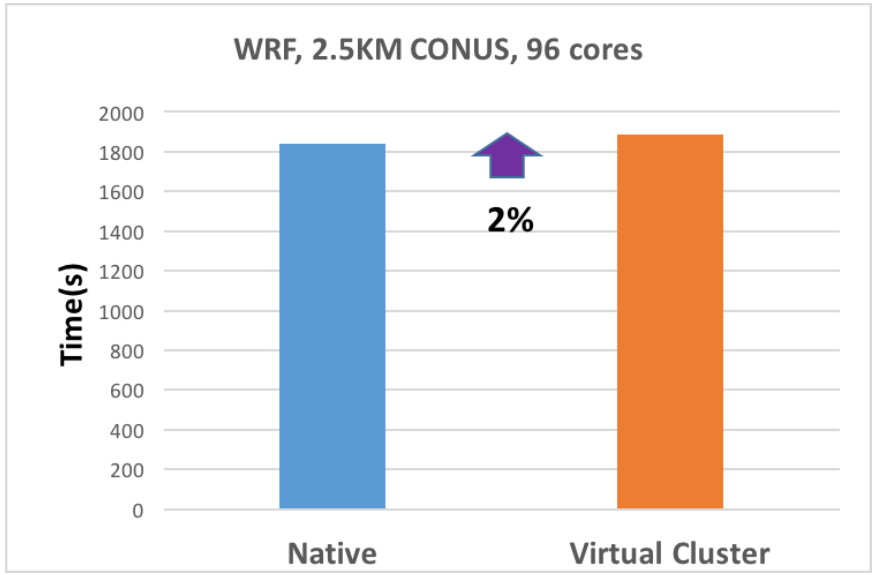


Network Latency



Application Benchmarks on Comet Virtual Clusters

WRF Benchmark



- 96-core (4-node) calculation
- Test Case: 3hr Forecast, 2.5km resolution of Continental US (CONUS).
- 2% slower w/ SR-IOV vs native IB.

PSDNS Benchmark

Cores (Nodes)	Time/Step (s)
32(2)	101.51
64(4)	67.03
128(8)	33.99

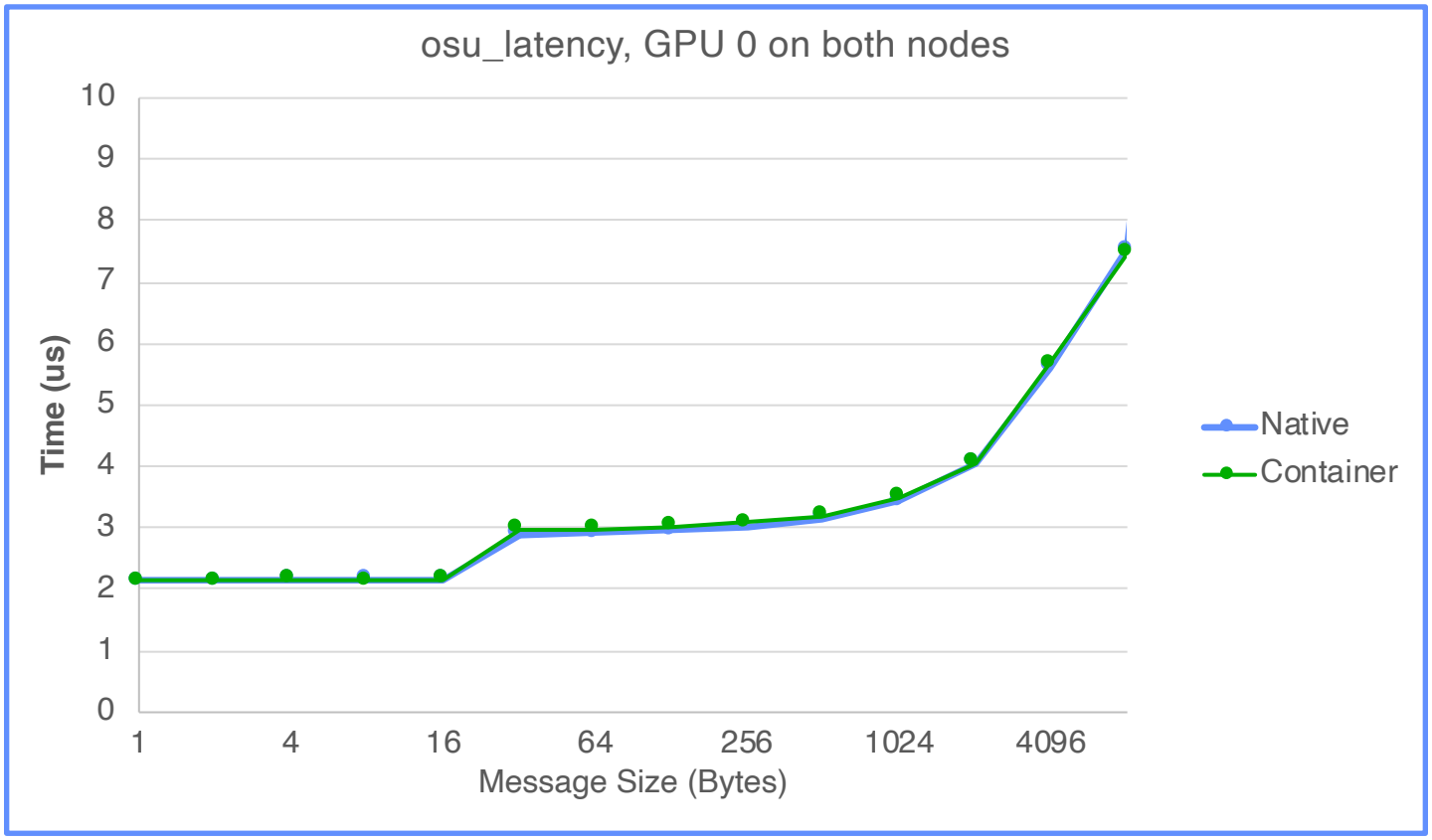
- FFT based application
- Communication intensive, mainly alltoallv
- Bisection bandwidth limited.

WRF 3.4.1 – 3hr forecast

Singularity: Provides Flexibility for OS Environment

- Singularity (http://singularity.lbl.gov)is a relatively new development that has become very popular on Comet.
- Singularity allows groups to easily migrate complex software stacks from their campus to Comet.
- Singularity runs in user space, and requires very little special support – in fact it actually reduces it in some cases.
- Applications include: Tensorflow, Torch, Fenics, and custom user applications.
- Docker images can be imported into Singularity.

MVAPICH2-GDR (v2.3.2) Results Using Containerized Approach



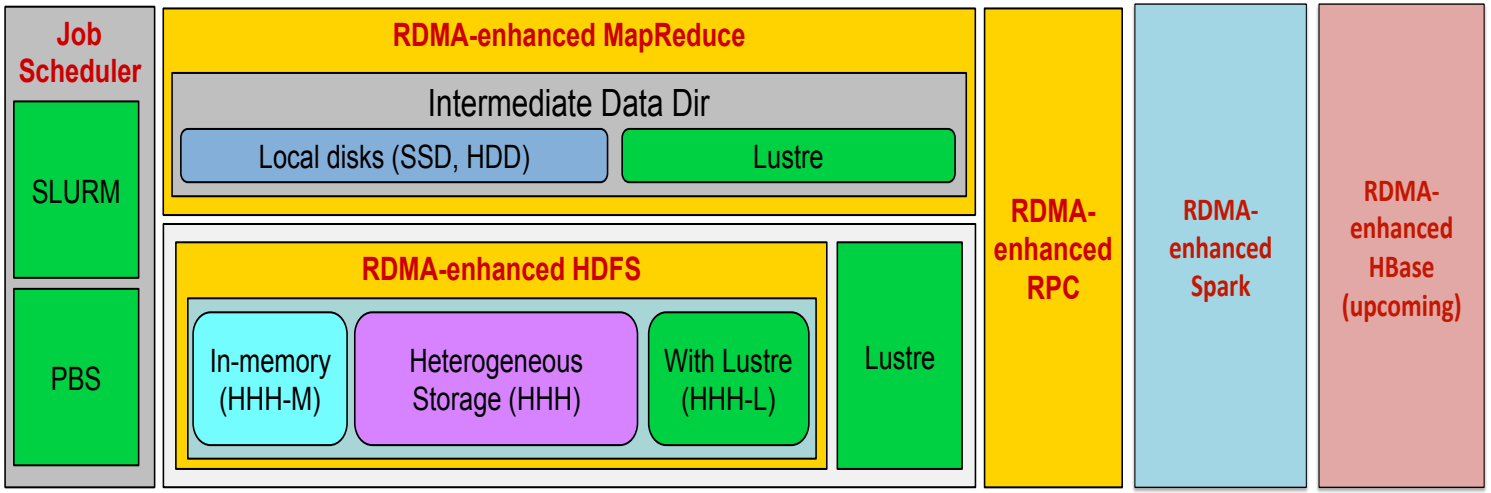
Application example using Singularity and MVAPICH2

- **Neuron YuEtAl2012 benchmark, compared the same build options using gnu+MVAPICH2 compilers via singularity.**

Cores	Time (seconds)
192	373
384	188
768	107

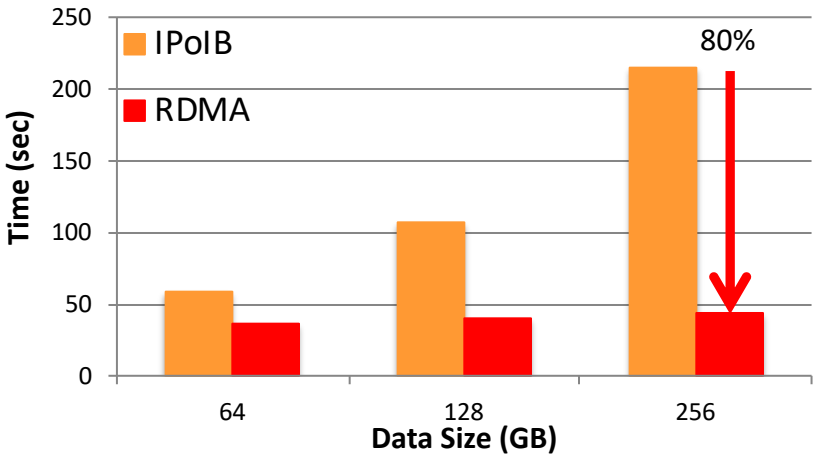
RDMA-Hadoop and Spark Design

- Exploit performance on modern clusters with RDMA-enabled interconnects for Big Data applications.
- Hybrid design with in-memory and heterogeneous storage (HDD, SSDs, Lustre).
- Keep compliance with standard distributions from Apache.

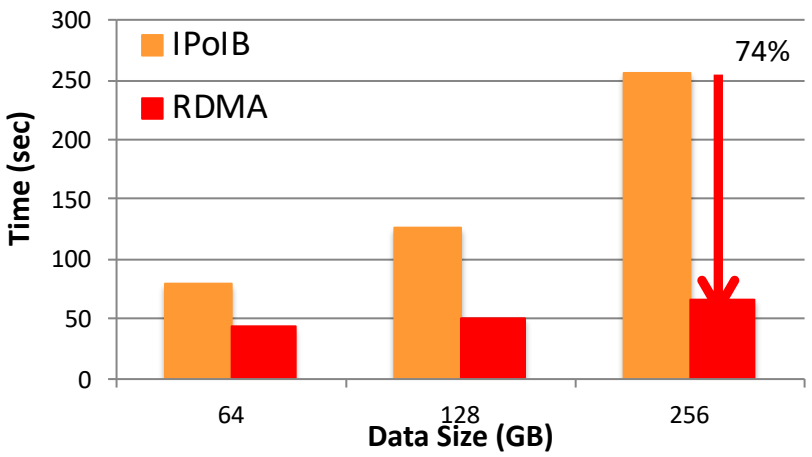


Reference: http://hibd.cse.ohio-state.edu/

Performance Evaluation on SDSC Comet – SortBy/GroupBy



64 Worker Nodes, 1536 cores, SortByTest Total Time

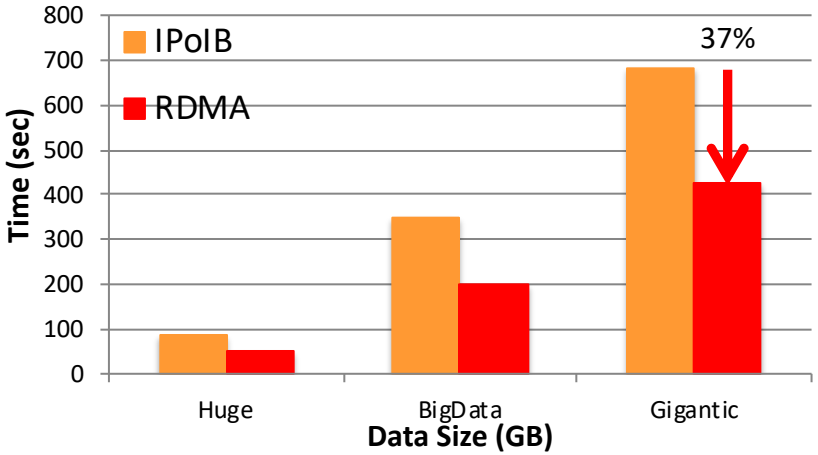


64 Worker Nodes, 1536 cores, GroupByTest Total Time

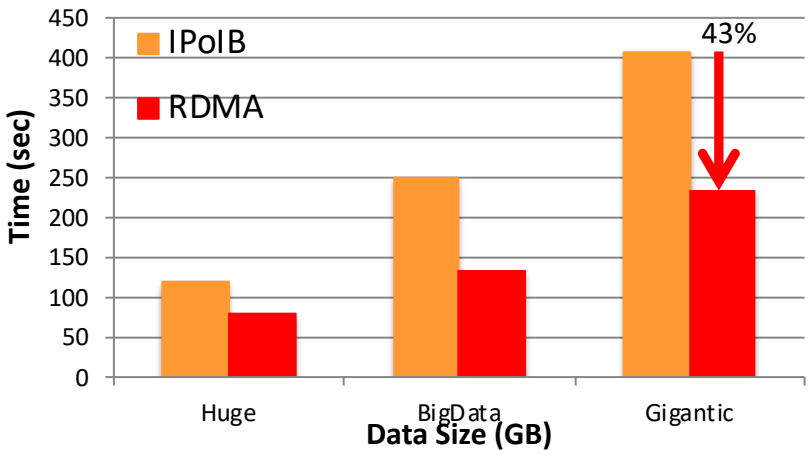
- **InfiniBand FDR, SSD, 64 Worker Nodes, 1536 Cores, (1536M 1536R)**
- **RDMA-based design for Spark 1.5.1**
- **RDMA vs. IPoIB with 1536 concurrent tasks, single SSD per node.**
  - SortBy: Total time reduced by up to 80% over IPoIB (56Gbps)
  - GroupBy: Total time reduced by up to 74% over IPoIB (56Gbps)

Reference: http://hibd.cse.ohio-state.edu/

Performance Evaluation on SDSC Comet – HiBench PageRank



32 Worker Nodes, 768 cores, PageRank Total Time



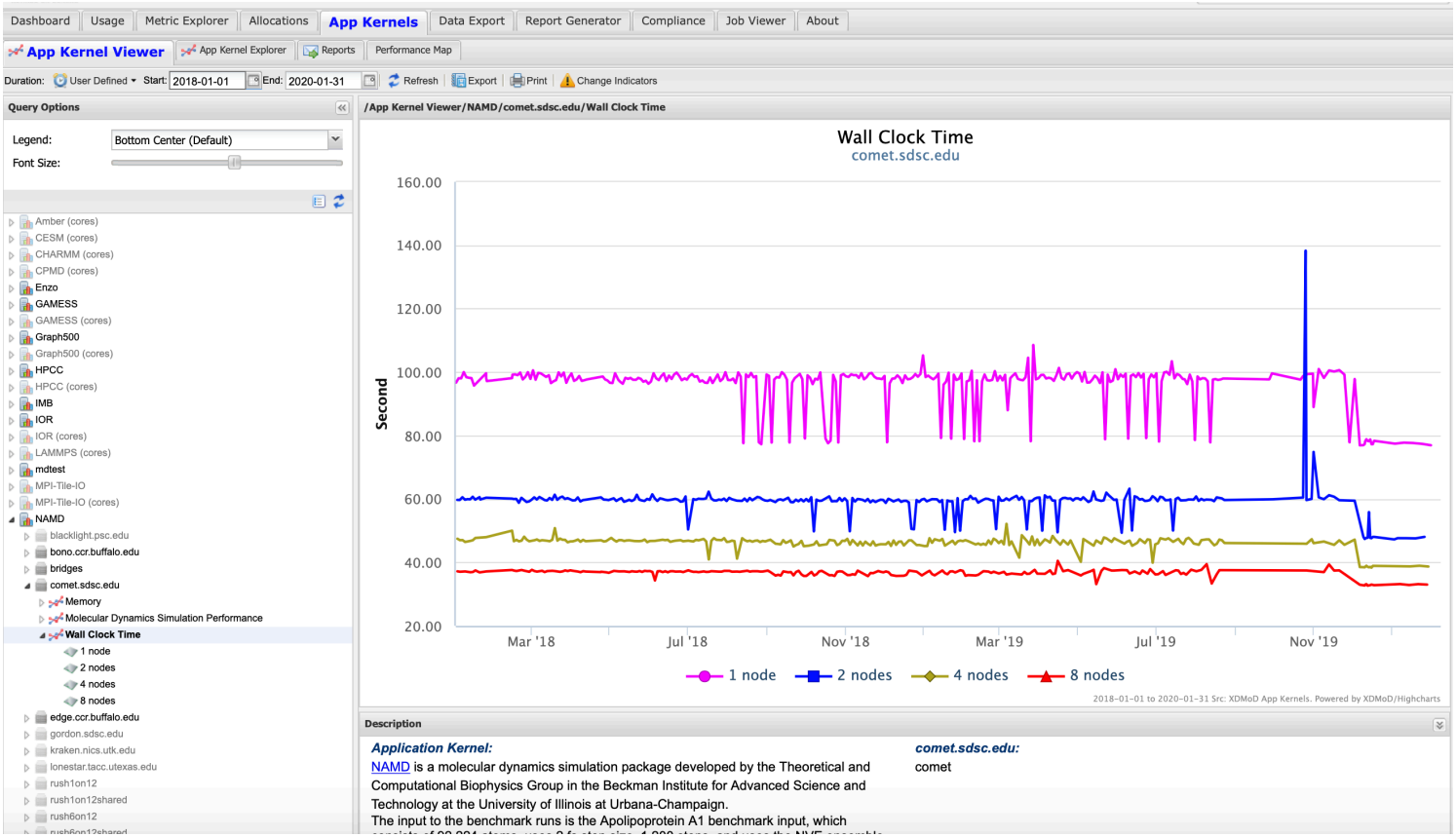
64 Worker Nodes, 1536 cores, PageRank Total Time

- **InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)**
- **RDMA-based design for Spark 1.5.1**
- **RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.**
  - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
  - 64 nodes/1536 cores: Total time reduced by 43% over IPoIB (56Gbps)

Reference: http://hibd.cse.ohio-state.edu/

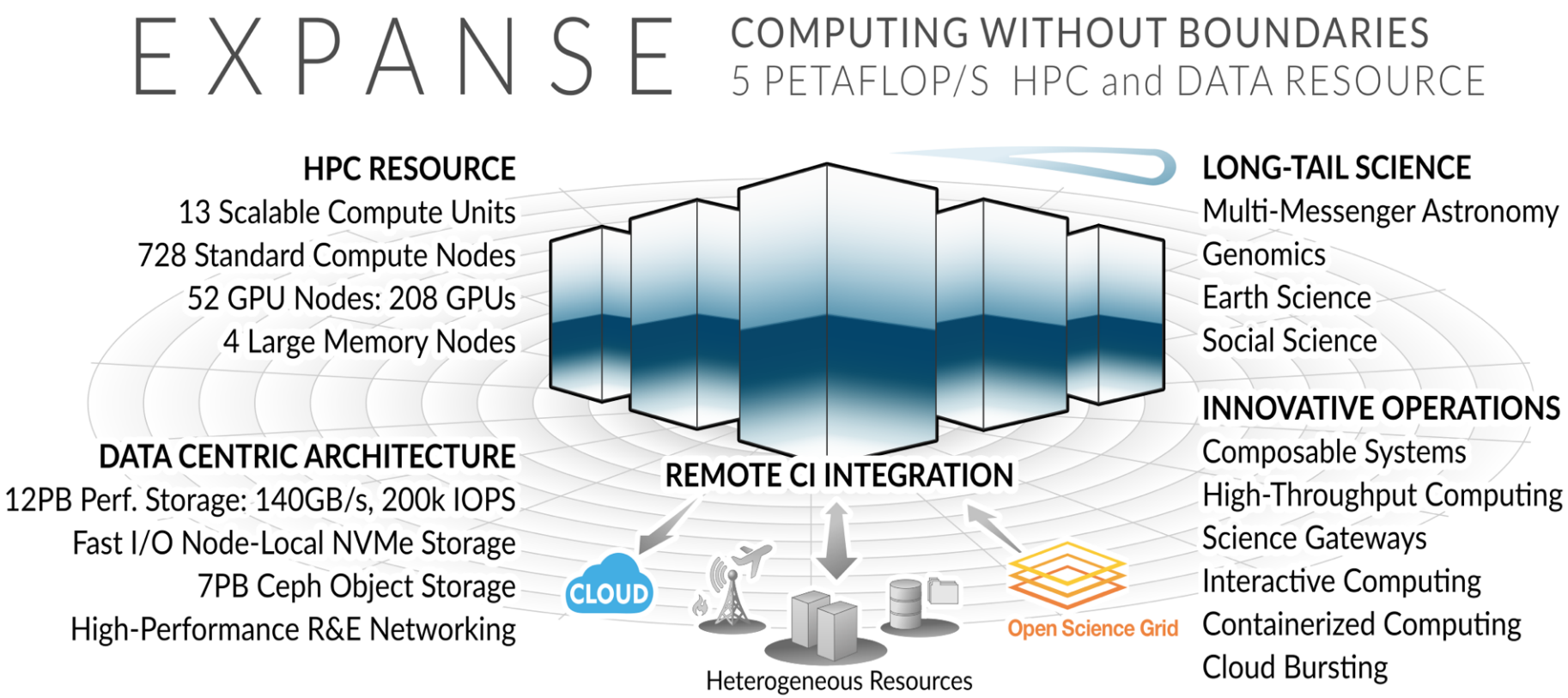
XDMoD Application Kernels

- XSEDE Metrics on Demand (XDMoD) is a comprehensive auditing framework (xdmod.ccr.buffalo.edu).
- Application kernels are run frequently (daily to several times per week) to continuously monitor HPC system performance.



Upcoming SDSC HPC System

Several innovative features => expansion of benchmarking efforts



Summary

- Benchmarking suite has evolved significantly as SDSC systems have expanded hardware and software features
- Early systems featured a traditional HPC workload and relatively simple network architecture
- Features such as dual rail networks, flash memory, GPUs, and vSMP require customization of benchmarking approach for effective performance testing
- Virtualization and containerization solutions need separate testing (with standard low-level benchmarks and applications)
- Rapidly expanding application base – significant rise in bioinformatics and machine learning, data analytics. Changed the mix of applications used for benchmarking
- XDMoD application kernels help monitor system performance over time