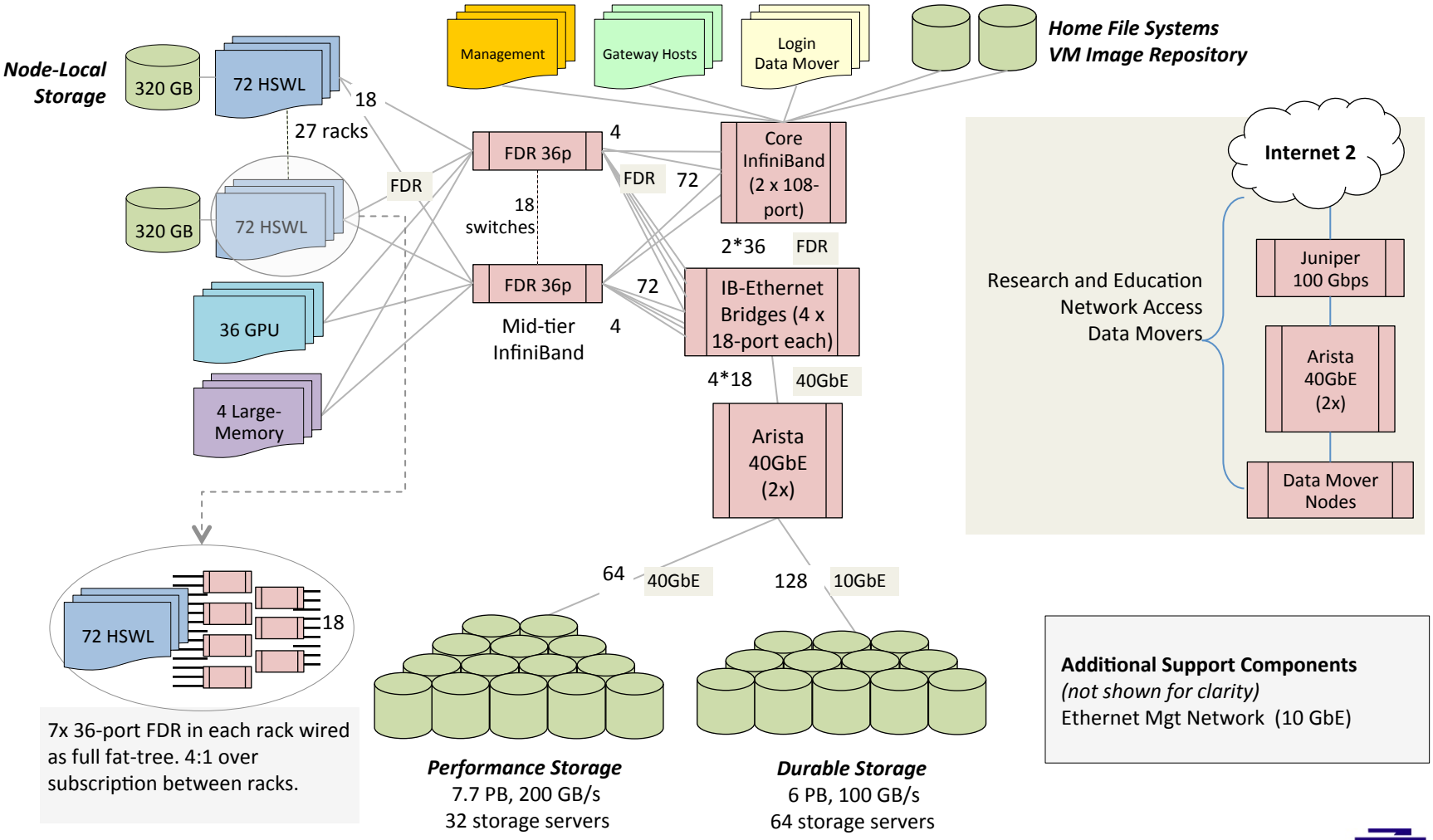
ECSS Staff Training
Comet Virtual Clusters

XSEDE 15
Saint Louis, MO
July 26-30

Rick Wagner
San Diego Supercomputer Center

Comet Network Architecture

InfiniBand compute, Ethernet Storage



Virtualized Clusters on Comet

Goal:

Provide a near bare metal HPC performance and management experience

Target Use

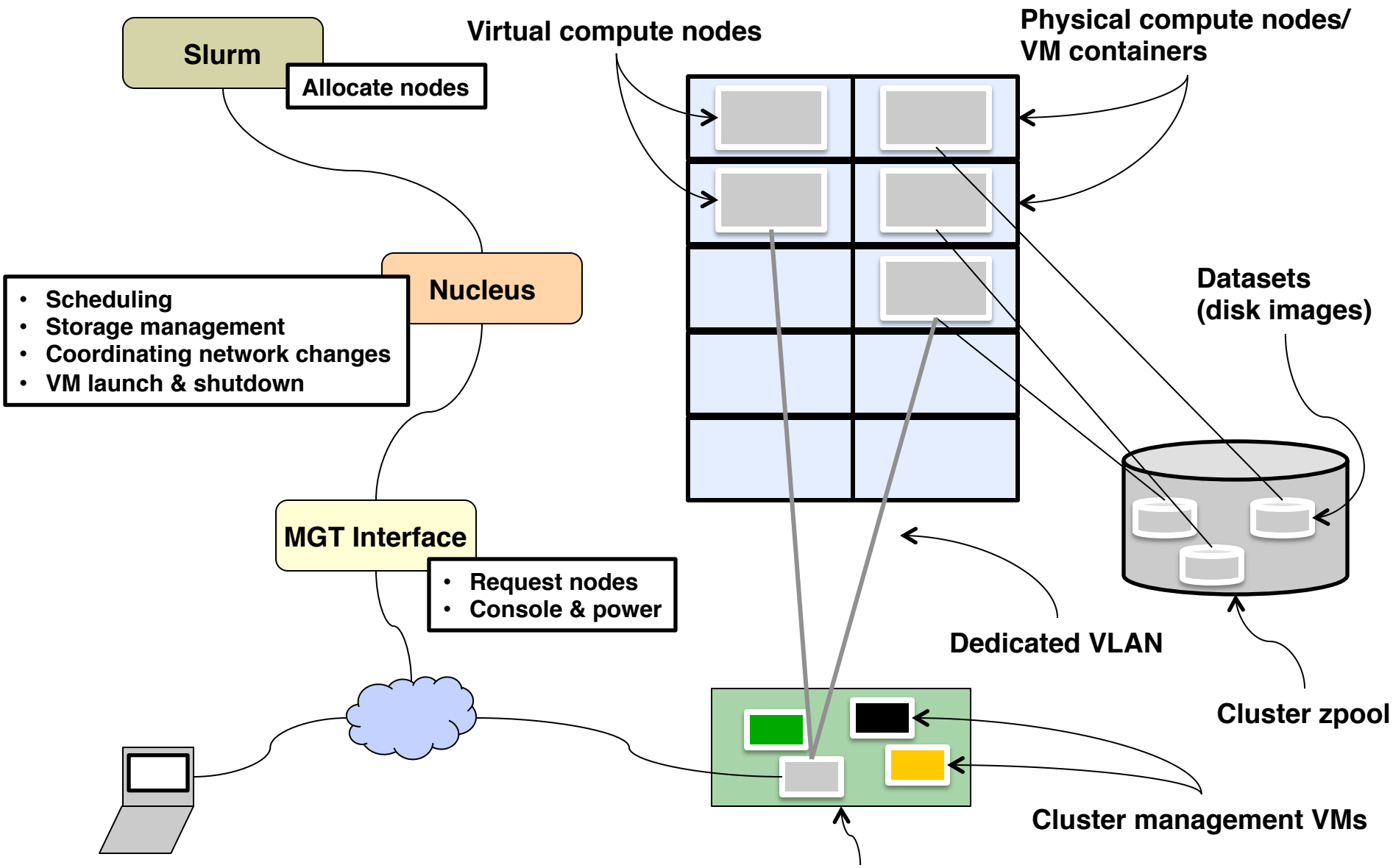
Projects that could manage their own cluster, and:

- can't fit our batch environment, and
- don't want to buy hardware or
- have bursty or intermittent need

Concepts, Ideas, Design

- Projects have persistent VM for cluster management
  - Modest: single core, 1-2 GB of RAM
- Standard compute nodes will be scheduled as containers via batch system
  - One virtual compute node per container
- Virtual disk images stored as ZFS datasets
  - Migrated to and from containers at job start and end
- VM use allocated and tracked like regular computing

~~OpenStack~~ ~~Amazon Clone~~


Slurm
Allocate nodes
Virtual compute nodes
Physical compute nodes/
VM containers
Nucleus
• Scheduling
• Storage management
• Coordinating network changes
• VM launch & shutdown
Datasets
(disk images)
MGT Interface
• Request nodes
• Console & power
Dedicated VLAN
Cluster zpool
Cluster management VMs
Persistent VM container

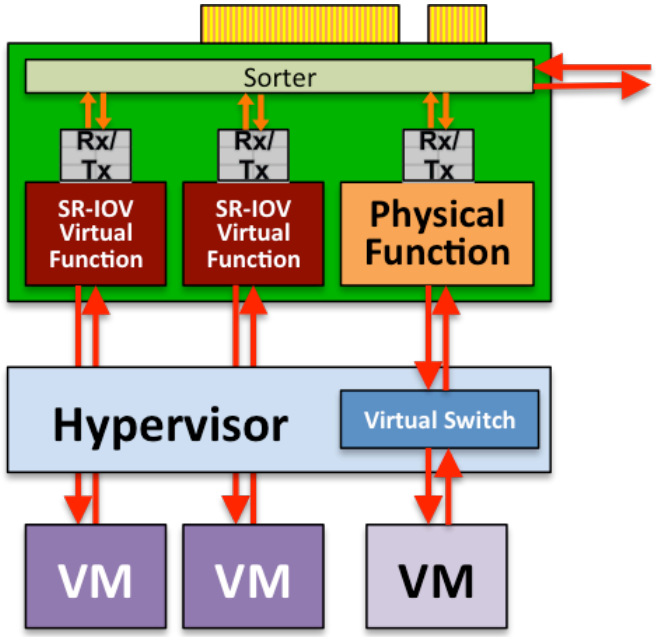
Enabling Technologies

- KVM—Let's us run virtual machines (all processor features)
- SR-IOV—Makes MPI go fast on VMs
- Rocks—Systems management
- ZFS—Disk image management
- VLANs—Isolate virtual cluster management network
- pkeys—Isolate virtual cluster IB network
- Nucleus—Coordination engine (scheduling, provisioning, status, etc.)

Yes, we have a lot of work to do.

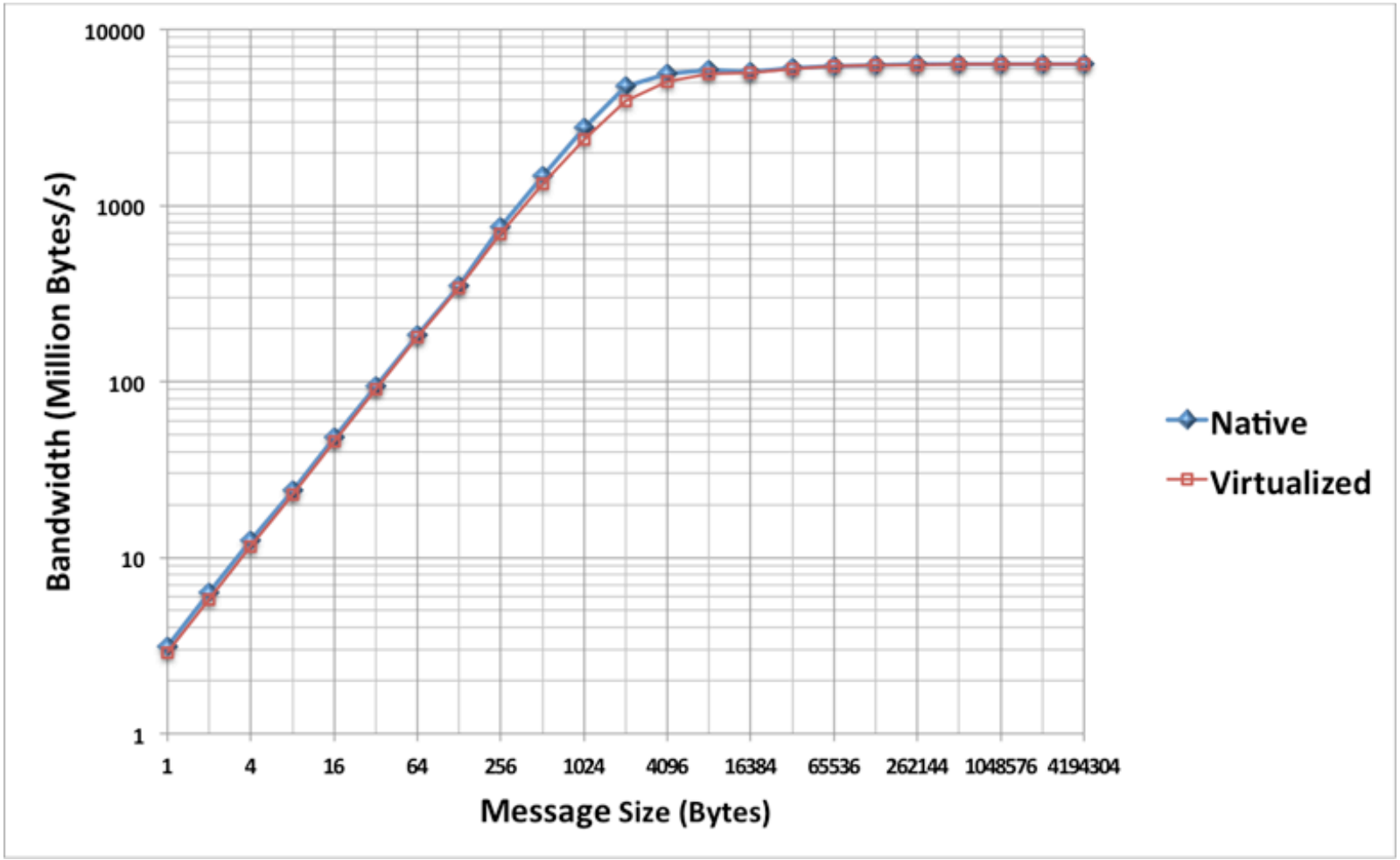
Single Root I/O Virtualization in HPC

- **Problem**: Virtualization generally has resulted in significant I/O performance degradation (e.g., excessive DMA interrupts)
- **Solution**: SR-IOV and Mellanox ConnectX-3 InfiniBand host channel adapters
  - One physical function → multiple virtual functions, each light weight but with its own DMA streams, memory space, interrupts
  - Allows DMA to bypass hypervisor to VMs
- ***SRIOV enables virtual HPC cluster w/ near-native InfiniBand latency/bandwidth and minimal overhead***

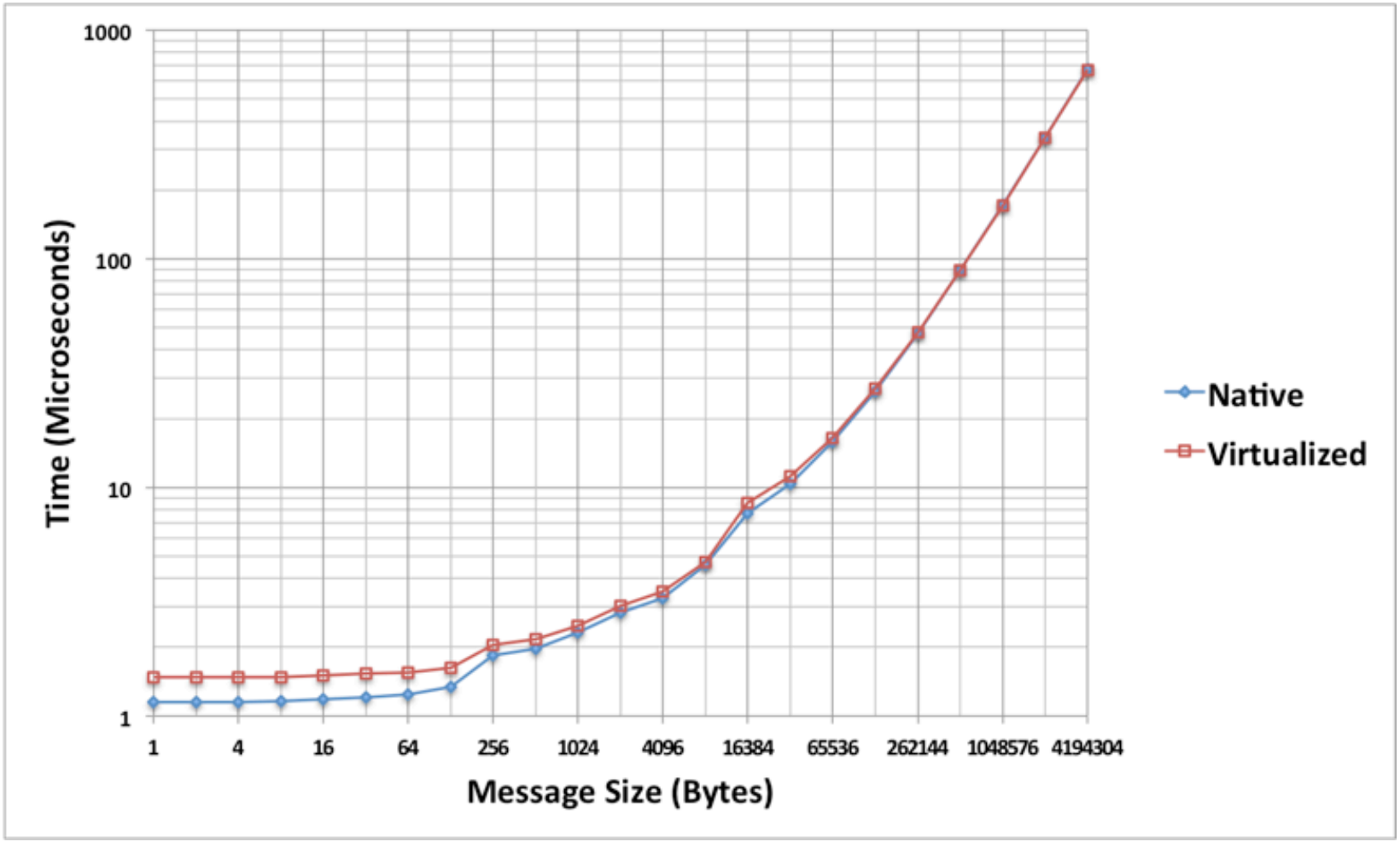


Mellanox TECHNOLOGIES

MPI bandwidth slowdown from SR-IOV is at most 1.21 for medium-sized messages & negligible for small & large ones

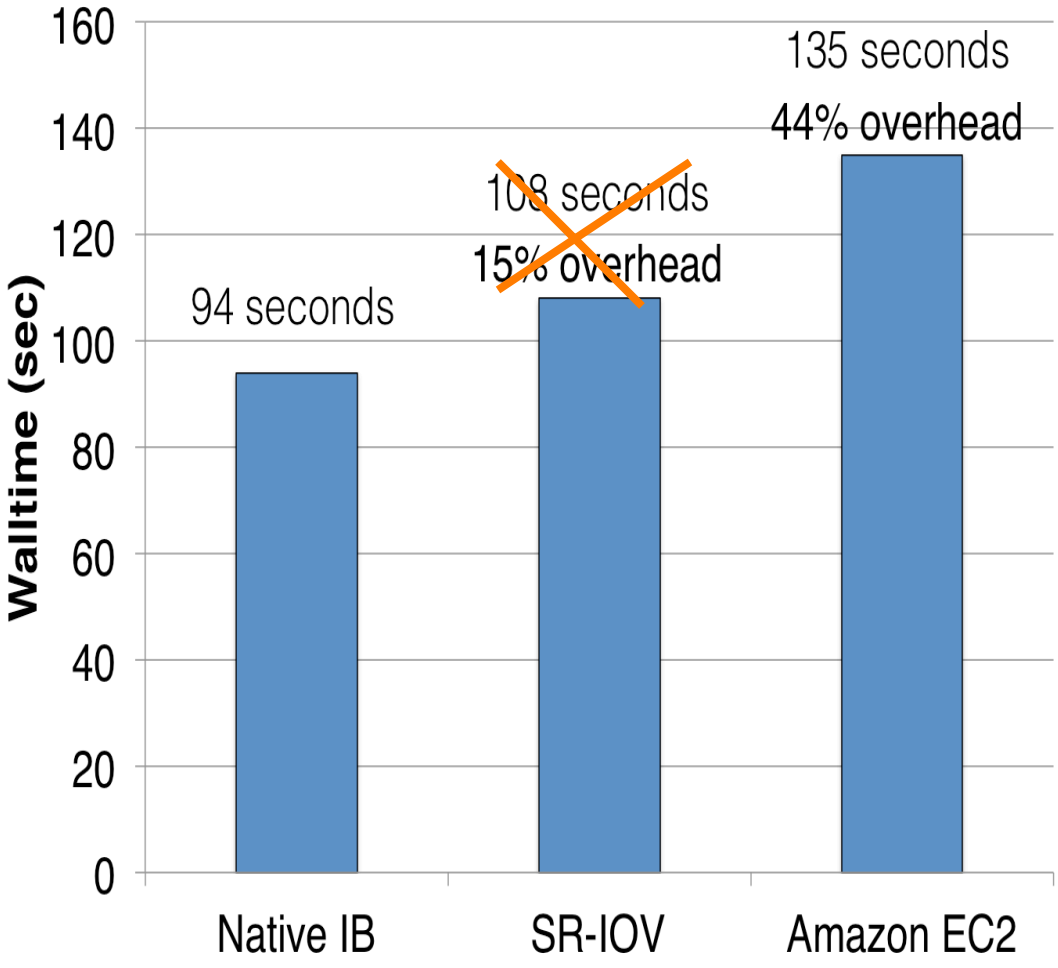


MPI latency slowdown from SR-IOV is at most 1.32 for small messages & negligible for large ones



WRF Weather Modeling – 15% Overhead with SR-IOV IB

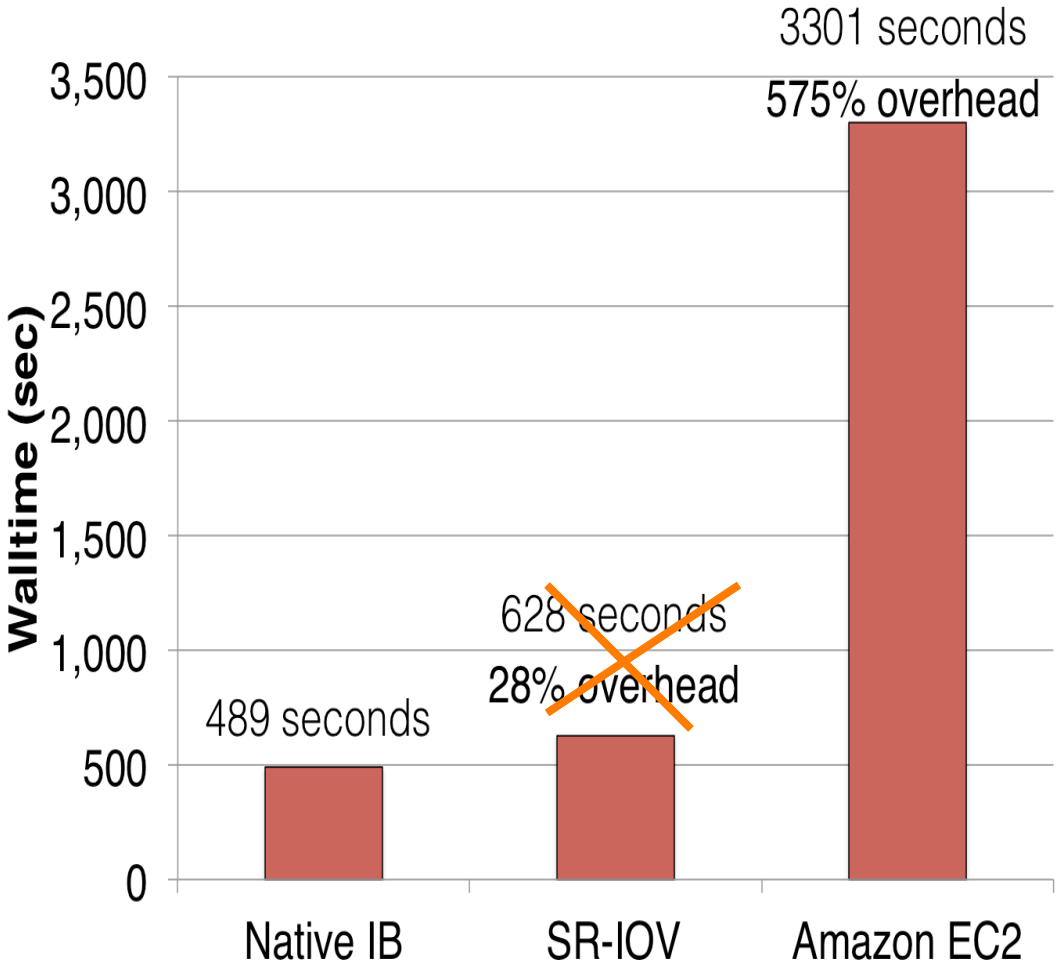
- 96-core (6-node) calculation
- Nearest-neighbor communication
- Scalable algorithms
- ~~SR-IOV incurs modest (15%) performance hit~~
- 2% slower w/ SR-IOV vs native IB!
- Still 20% faster than EC2 Despite 20% slower CPUs



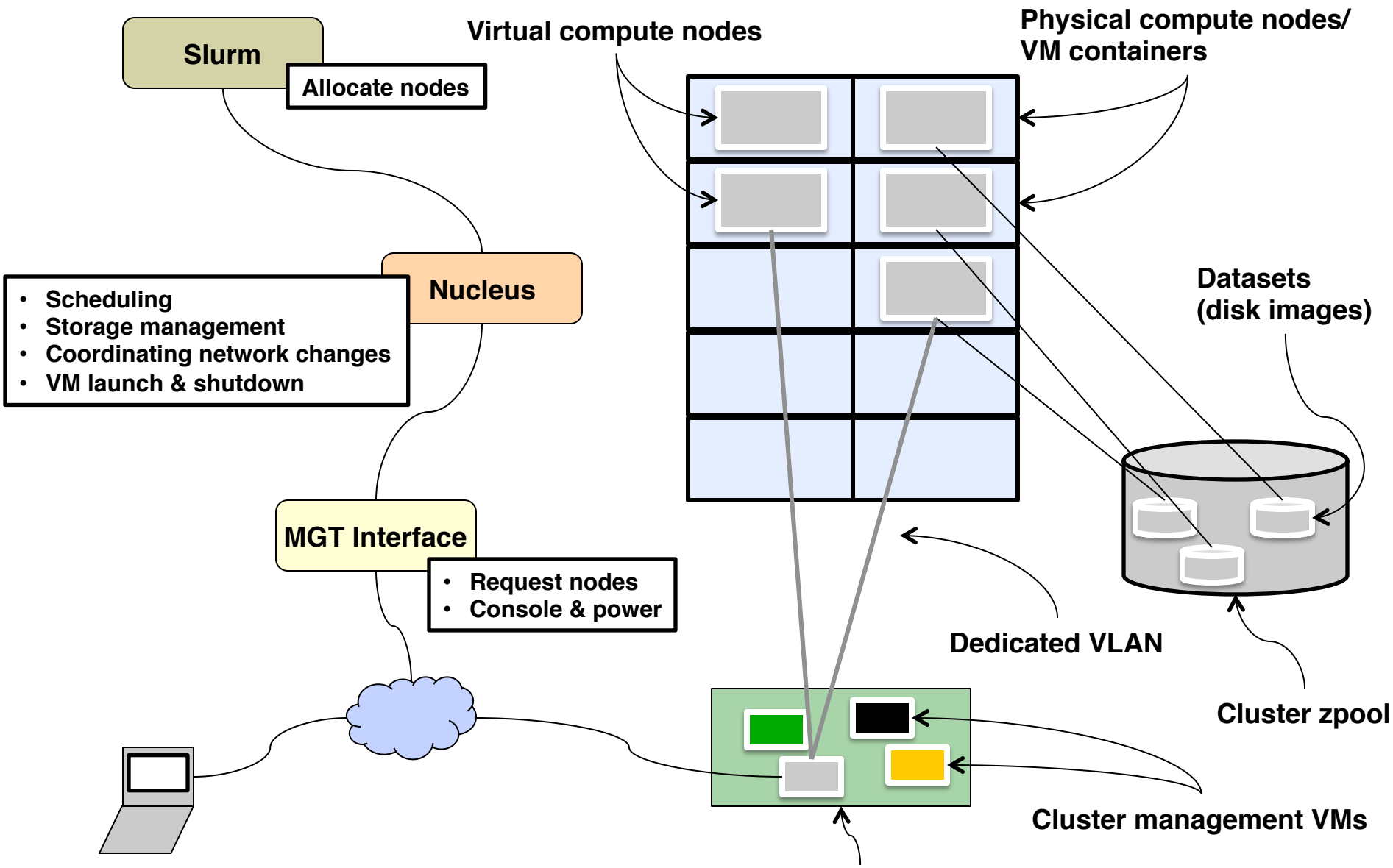
WRF 3.4.1 – 3hr forecast

Quantum ESPRESSO: 28% Overhead

- 48-core (3 node) calculation
- CG matrix inversion - irregular communication
- 3D FFT matrix transposes (all-to-all communication)
- ~~28% slower w/ SR-IOV vs native IB~~
- 8% slower w/ SR-IOV vs native IB!
- SR-IOV still > 500% faster than EC2 Despite 20% slower CPUs



Quantum Espresso 5.0.2 – DEISA AUSURF112 benchmark


Slurm
Allocate nodes
Virtual compute nodes
Physical compute nodes/
VM containers
Nucleus
• Scheduling
• Storage management
• Coordinating network changes
• VM launch & shutdown
Datasets
(disk images)
MGT Interface
• Request nodes
• Console & power
Dedicated VLAN
Cluster zpool
Cluster management VMs
Persistent VM container

Deploying and Using a Cluster (User View)

1. Provision front end (management node) from ISO
2. Request compute nodes
3. Install compute nodes automated or manually (PXE or from ISO)
4. Shut down compute nodes for later use
5. Request compute nodes to be booted as needed

Expected Users

Large projects and research groups that have dedicated IT staff and need more customization and control

Examples

- XSEDE SD&I
  - Testing software in different environments
- Campuses
  - Provide consistent environment
- Gateways
- SDSC
  - Continuous integration of cluster applications

Intended Usage Modes

Mode 1: Bursty

- Sporadic workload
- Asynchronous batch processing
- Request nodes as needed

Mode 2: Minimum Footprint

- Some nodes always running
- Continuous computing or data processing
- Can grow when needed

Allocations granted
and tracked just like
standard compute cycles.
Running a VM means spending SUs

Development Roadmap

- Now
  - Deploying development cluster
- August
  - Manual cluster provisioning
  - Scheduler integration
- September
  - Testing and usage by IU staff
  - Initial interface
- October
  - Network configuration
  - Refine interface
- November
  - Friendly user testing
- December
  - Deployment
- January 2016
  - Production
- Later
  - Expose API

Future Features

- Network storage
  - Lustre
  - NFS
- API
  - Needed for automated workflows

Potential ECSS Engagements

- Potentially, lots of new applications
  - VMs allow for new operating systems in HPC
- Performance tuning
- Workflows
- Data management
- Porting disk images