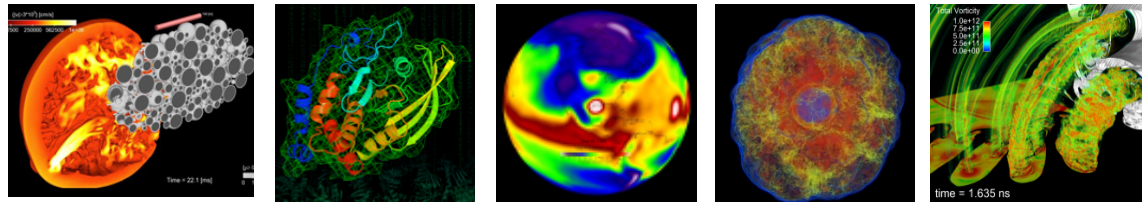


Trinity: Architecture and Early Experience



Scott Hemmert, Shaun Dawson, Si Hammond, Daryl Grunau, Rob Hoekstra, Mike Glass, Jim Lujan, Dave Morton, Hai Ah Nam, Paul Peltz Jr., Mahesh Rajan, Alfred Torrez, Manuel Vigil, Cornell Wright

Cray Users Group, May 2016

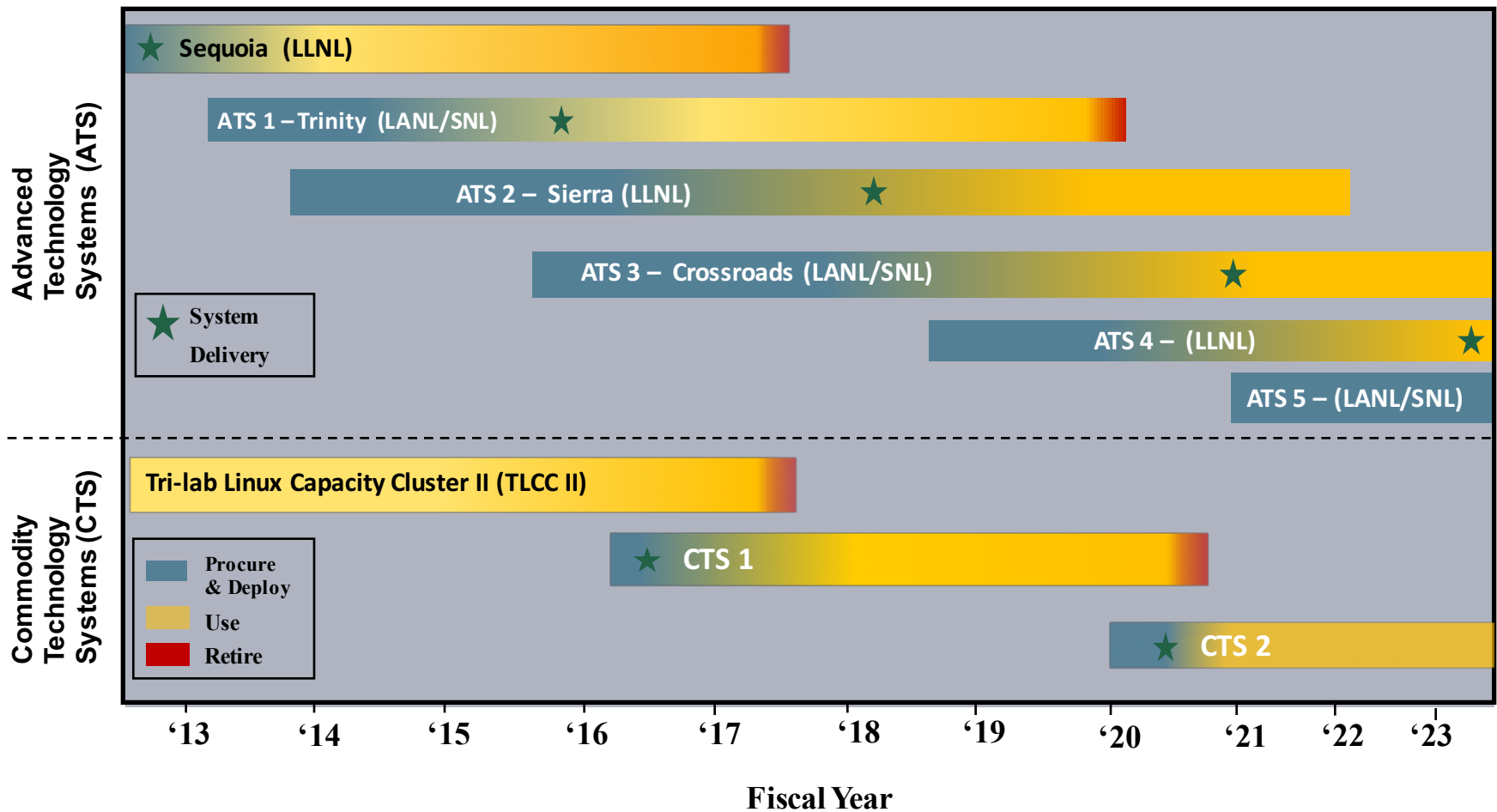


SAND2016-4374 C





ASC Platform Timeline



Jan 2016



Trinity Project Drivers

- Satisfy the mission need for more capable platforms
 - Trinity is designed to support the largest, most demanding ASC applications
 - Increases in geometric and physics fidelities while satisfying analysts' time-to- solution expectations
 - Foster a competitive environment and influence next generation architectures in the HPC industry
- Trinity is enabling new architecture features in a production computing environment
 - Trinity's architecture will introduce new challenges for code teams: transition from multi-core to many-core, high-speed on-chip memory subsystem, wider SIMD/vector units
 - Tightly coupled solid state storage serves as a "burst buffer" for checkpoint/restart file I/O & data analytics, enabling improved time-to-solution efficiencies
 - Advanced power management features enable measurement and control at the system, node, and component levels, allowing exploration of application performance/watt and reducing total cost of ownership
- Mission Need Requirements are primarily driving memory capacity
 - Over 2 PB of aggregate main memory



Trinity Architecture



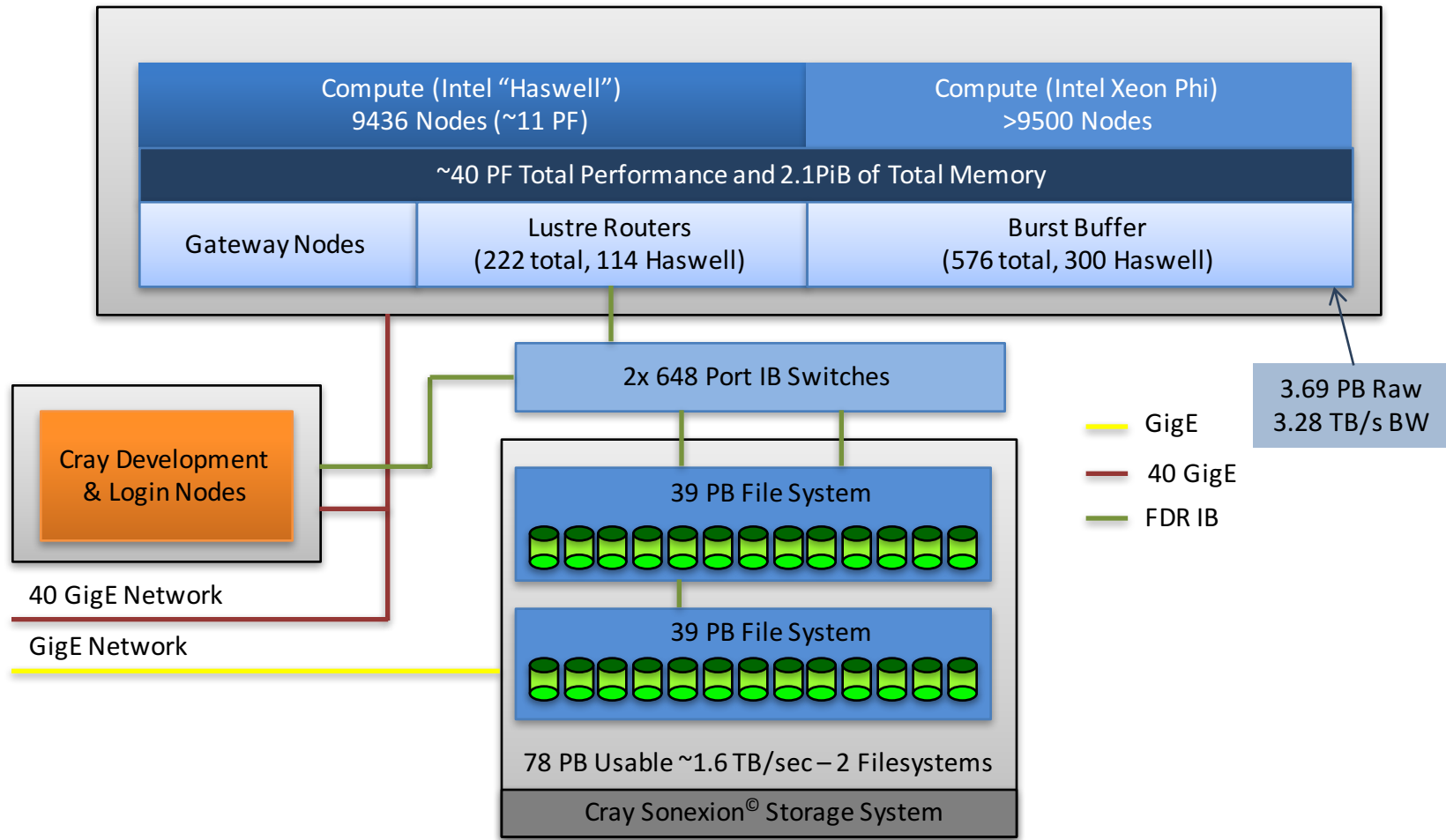


Trinity Platform

- Trinity is a single system that contains both Intel Haswell and Knights Landing processors
 - Haswell partition satisfies FY16 mission needs (well suited to existing codes).
 - KNL partition delivered in FY16 results in a system significantly more capable than current platforms and provides the application developers with an attractive next-generation target (and significant challenges)
 - Aries interconnect with the Dragonfly network topology
- Based on mature Cray XC30 architecture with Trinity introducing new architectural features
 - Intel Knights Landing (KNL) processors
 - Burst Buffer storage nodes
 - Advanced power management system software enhancements



Trinity Architecture





Cray Aries Interconnect

Cray Aries Blade

Blue Links (10x1)

To Other Groups,
10 Global Links
(4.7 GB/s per link)

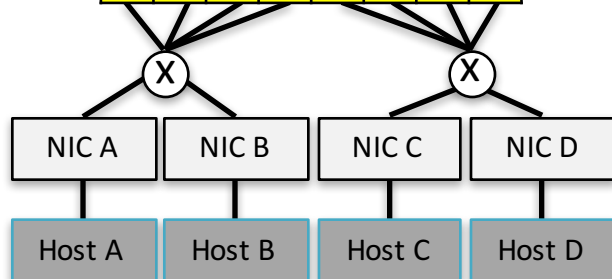
Green Links (15x1)

To 15 Other
Blades in Chassis,
1 Tile Each Link
(5.25 GB/s per link)

00	01	02	03	04	05	06	07
10	11	12	13	14	15	16	17
20	21	22	23	24	25	26	27
30	31	32	33	34	35	36	37
40	41	42	43	44	45	46	47
50	51	52	53	54	55	56	57

Black Links (5x3)

To 5 Other
Chassis in Group,
3 Tiles Each Link
(15.75 GB/s per link)



1. Chassis

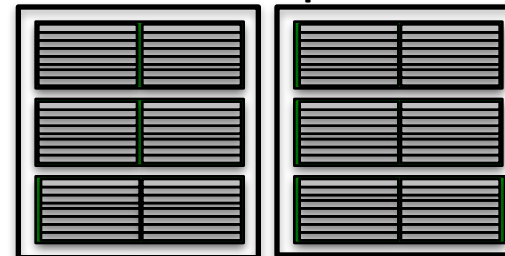


16 Blades Per Chassis

16 Aries, 64 Nodes

All-to-all Electrical Backplane

2. Group

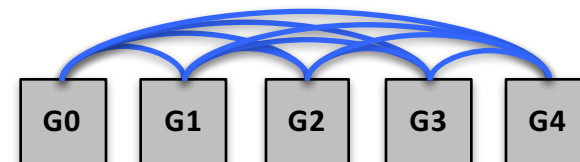


6 Chassis Per Group

96 Aries, 384 Nodes

Electrical Cables, 2-D All-to-All

3. Global



Up to 241 Groups

Up to 23136 Aries, 92544 Nodes

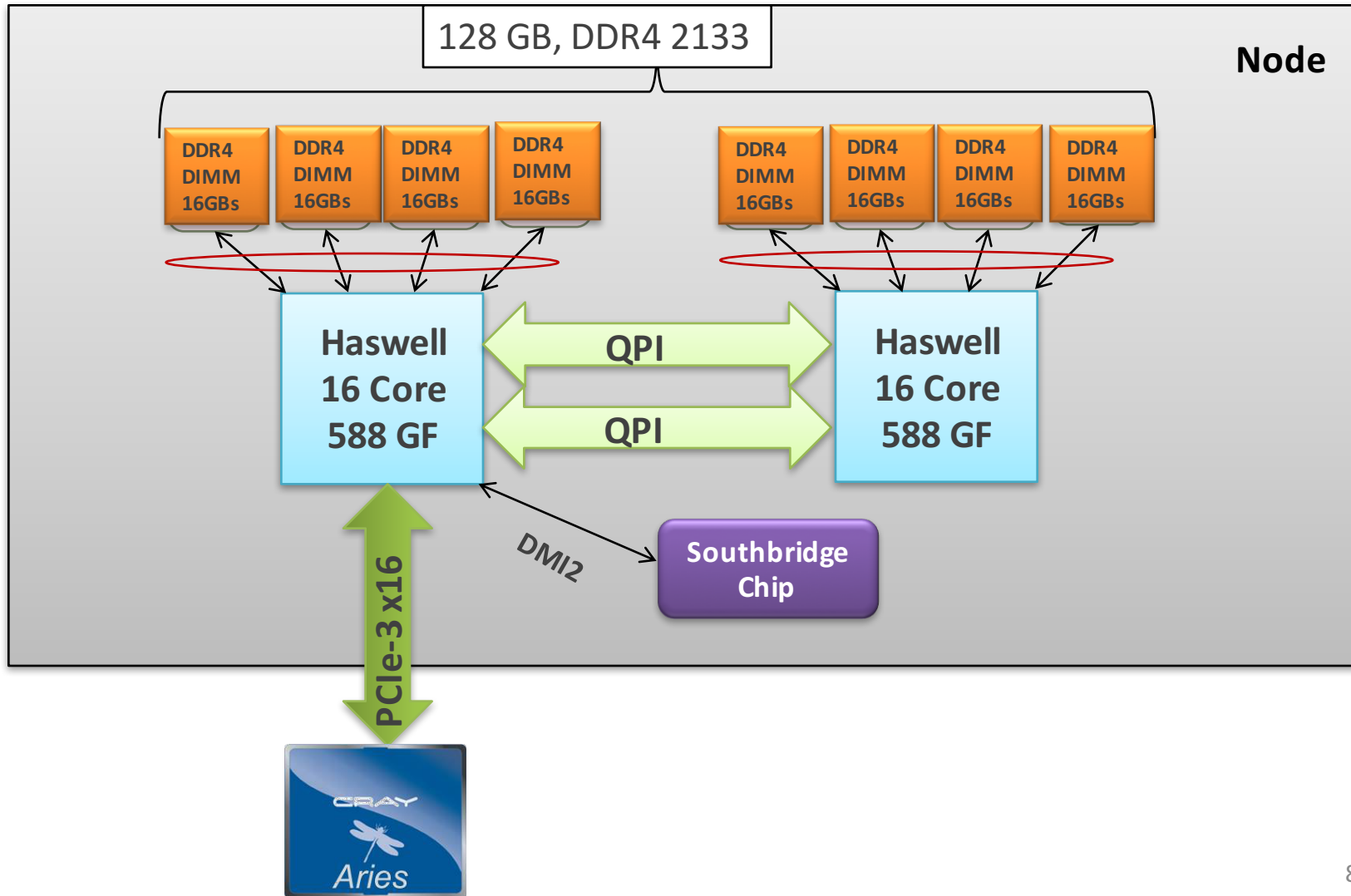
Optical Cables, All-to-All between Groups

Gemini: 2 nodes, 62.9 GB/s routing bw
Aries 4 nodes, 204.5 GB/s routing bw

Aries has advanced adaptive routing



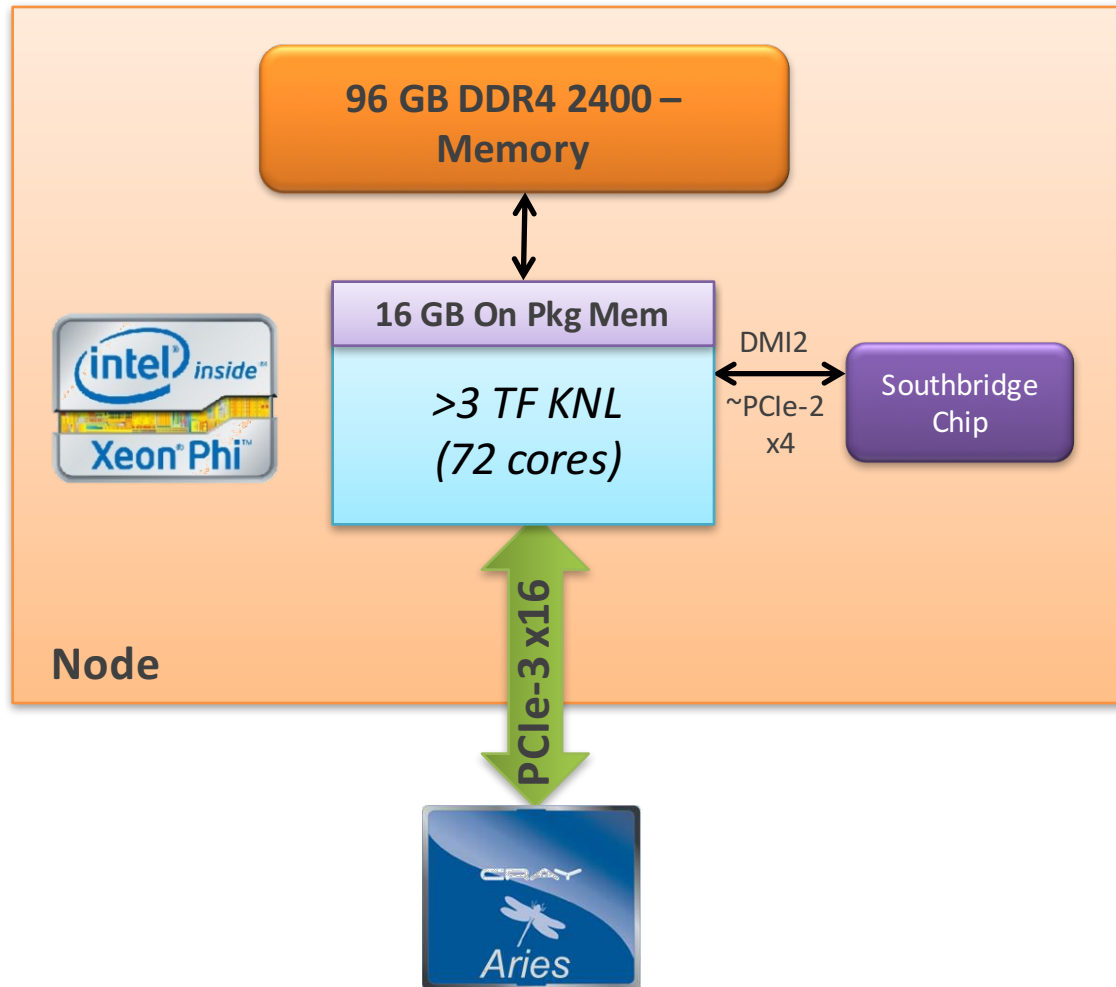
Trinity Haswell Compute Node





Trinity KNL Compute Node

Single Socket - Self Hosted Node





Test Bed Systems

- Gadget – Software Development Testbed
- Application Regression Testbeds
 - Configuration
 - 100 Haswell Compute Nodes
 - 720 TB / 15 GB/s Sonexion 2000 Filesystem
 - 6 Burst Buffer Nodes
 - Trinitite
 - LANL Yellow Network
 - Mutrino
 - Sandia SRN Network



Early Application Performance





Capability Improvement

- Defined as the product of an increase in problem size, and/or complexity, and an application specific runtime speedup factor over baseline measurement on NNSA's Cielo (a Cray XE6)
- Three applications chosen
 - Sierra Nalu
 - SIERRA/Nalu is a low Mach CFD code that solves a wide variety of variable density acoustically incompressible flows spanning from laminar to turbulent flow regimes.
 - Qbox
 - Qbox is a first-principles molecular dynamics code used to compute the properties of materials at the atomistic scale.
 - PARTISN
 - The PARTISN particle transport code [6] provides neutron transport solutions on orthogonal meshes in one, two, and three dimensions.



Capability Improvement Results

	Size/Complexity Increase	Relative Runtime	Capability Improvement
Sierra Nalu	1	4.009	4.009
Qbox	166.37	0.208	34.7
PARTISN	9.19	0.512	4.84



System Sustained Performance

Application Name	MPI Tasks	Threads	Nodes Used	Reference Tflops	Time (s)	Pi
miniFE (Total CG Time)	49152	1	1536	1065.151	49.5116	0.014005964
miniGhost (Total time)	49152	1	1536	3350.20032	17.7	0.122949267
AMG (GMRESSolve wall Time)	49152	1	1536	1364.51	66.233779	0.013412384
UMT (cumulativeWorkTime)	49184	1	1537	18409.4	454.057	0.026378822
SNAP (solve time)	12288	2	768	4729.66	177	0.034793285
miniDFT (Benchmark_time)	2016	1	63	9180.11	377.77	0.385726849
GTC (NERSC_TIME)	19200	1	300	19911.348	868.439	0.076425817
MILC (NERSC_TIME)	12288	1	384	15036.5	393.597	0.099486409
Geometric mean =						0.052990429
SSP =						500.0176846

Target = 400

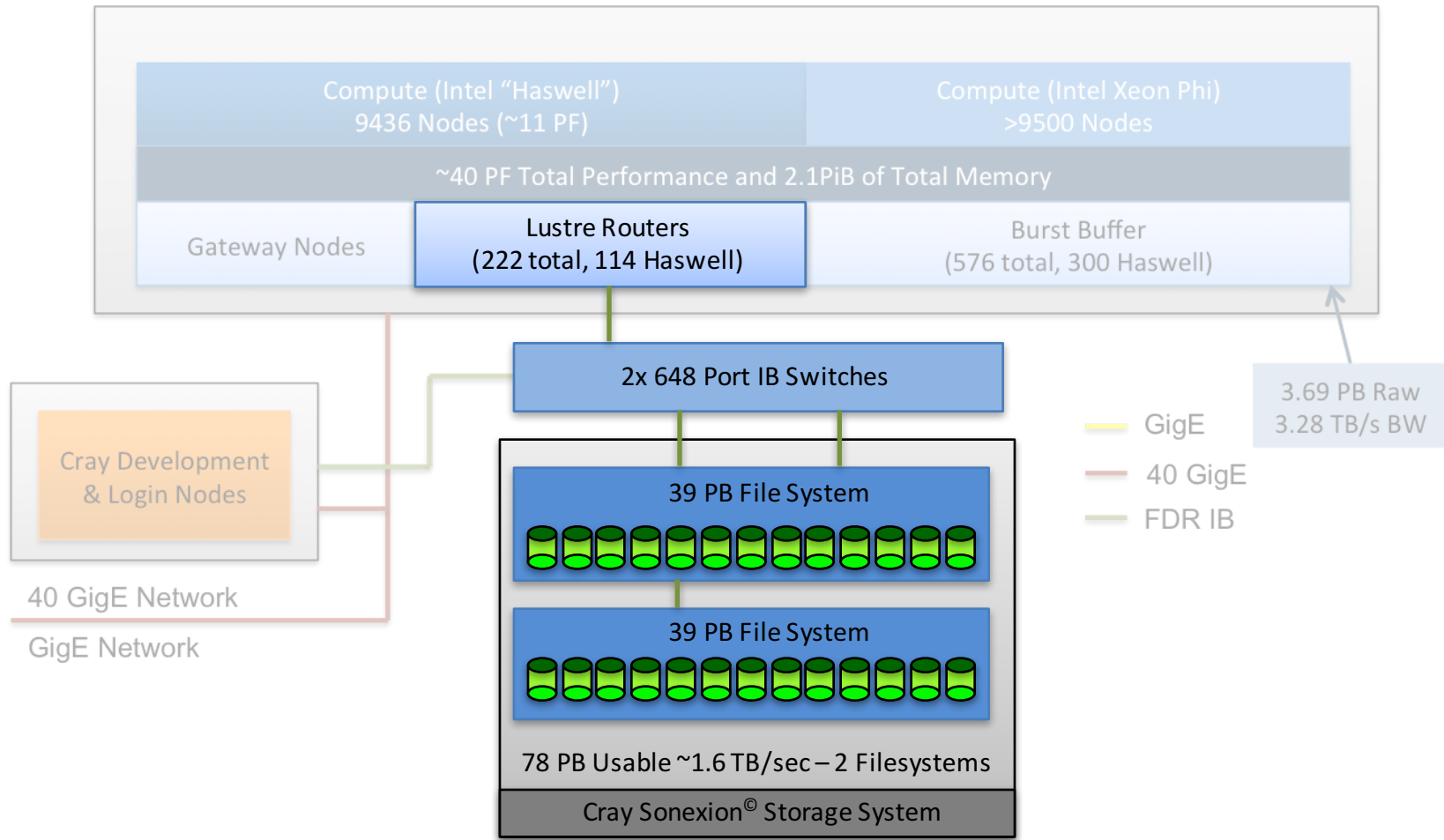


File System



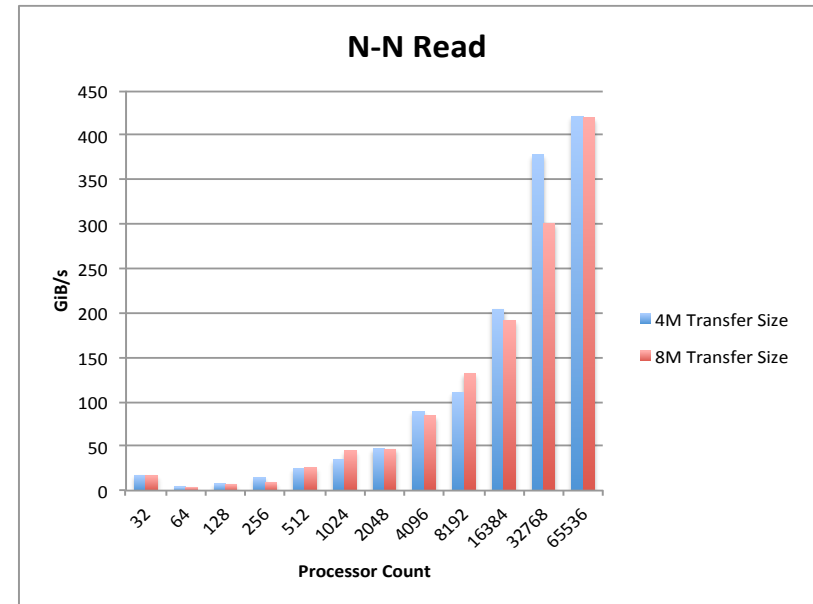
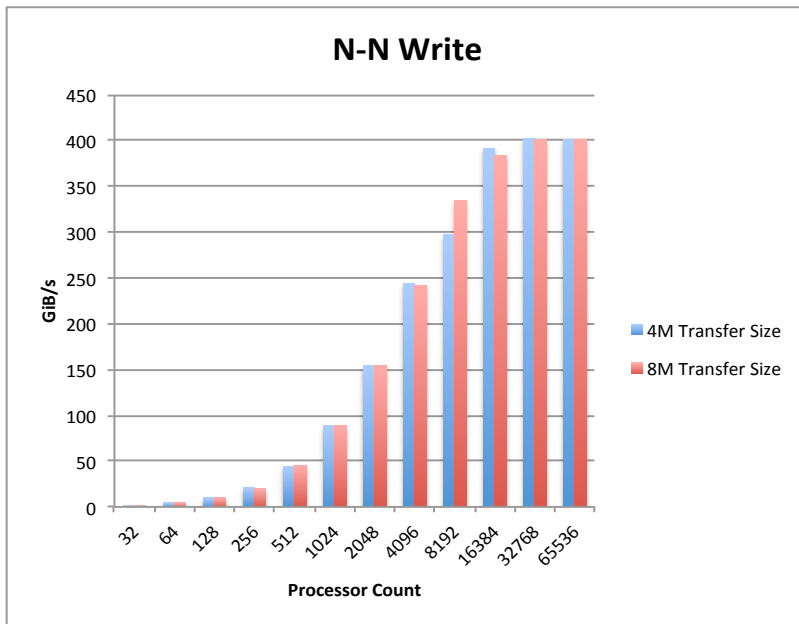


File System Configuration





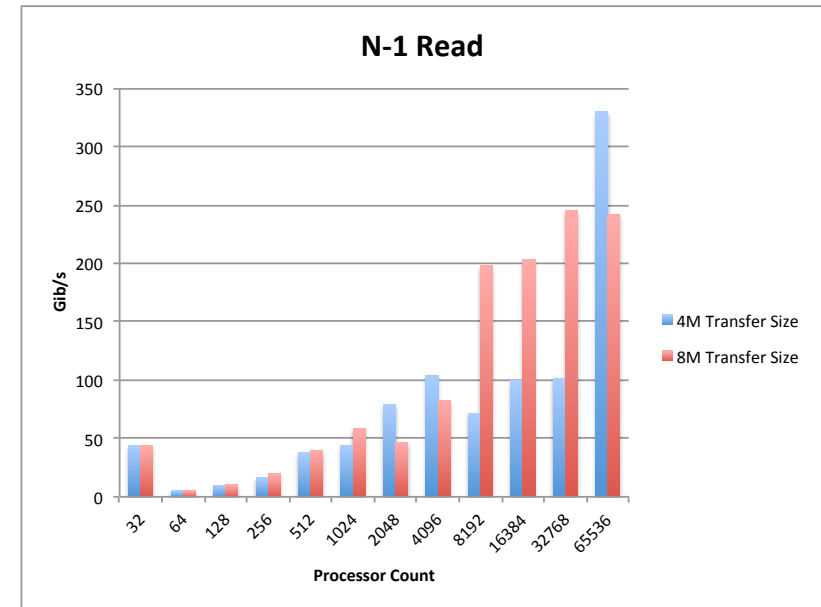
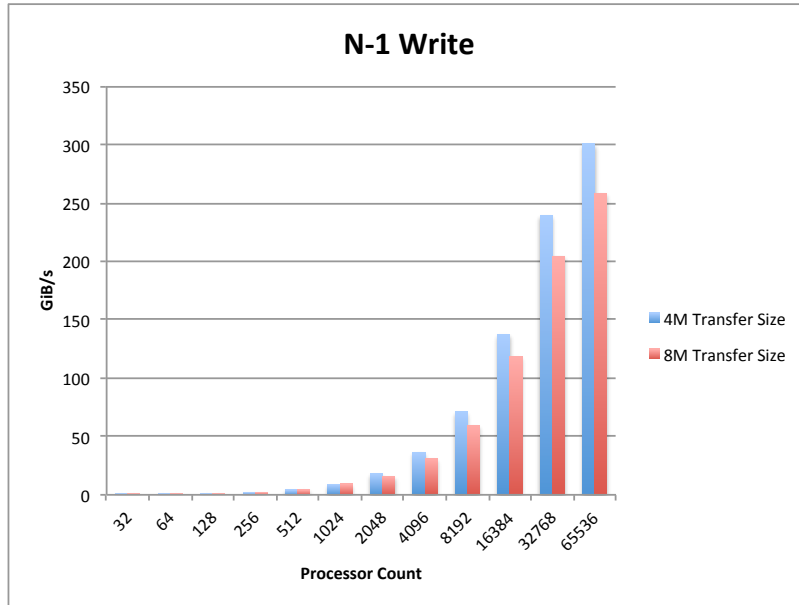
N-N Performance



- IOR, 32 processes per node, Each process writing 1 GiB
- Targeted 1 file system for these runs
- Max write: 401 GiB/s Max Read: 420GiB/s



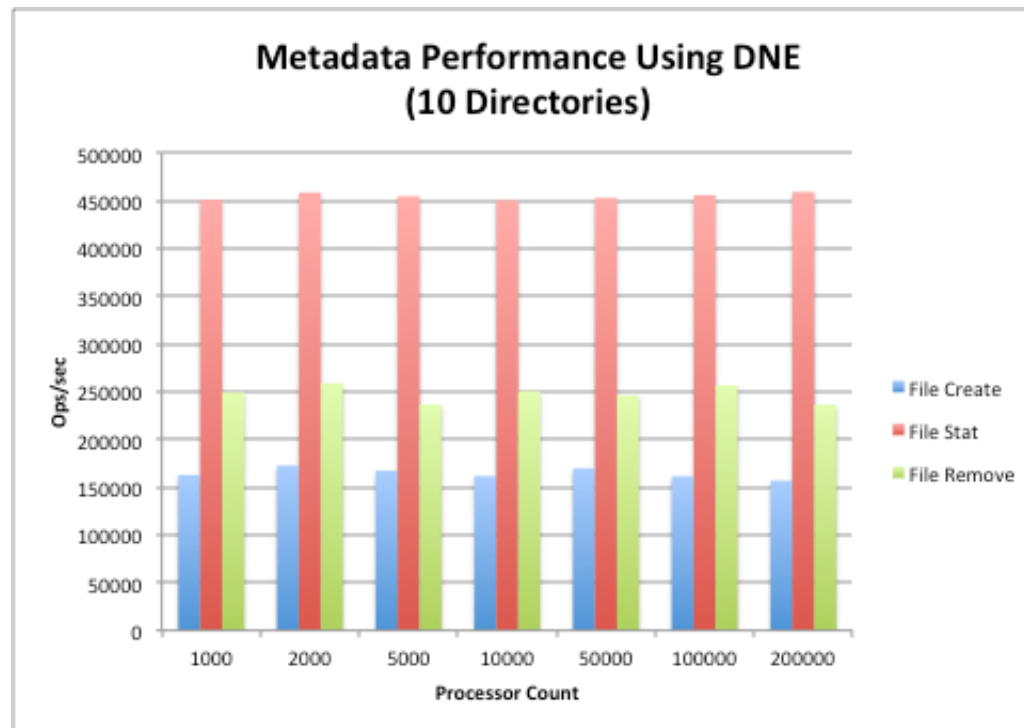
N-1 Performance



- IOR, 32 processes per node, Each process writing 1 GiB in strided pattern
- Target directory strip width set to OST count
- Target directory stripe size matched IOR transfer size
- Targeted 1 file system for these runs
- Max write: 301 GiB/s Max Read: 330 GiB/s



Metadata Performance



- Tested Lustre DNE phase 1 capability using 10 metadata servers each serving one directory
- mdtest, 32 procs per node
- Create, stat, delete 1 million files

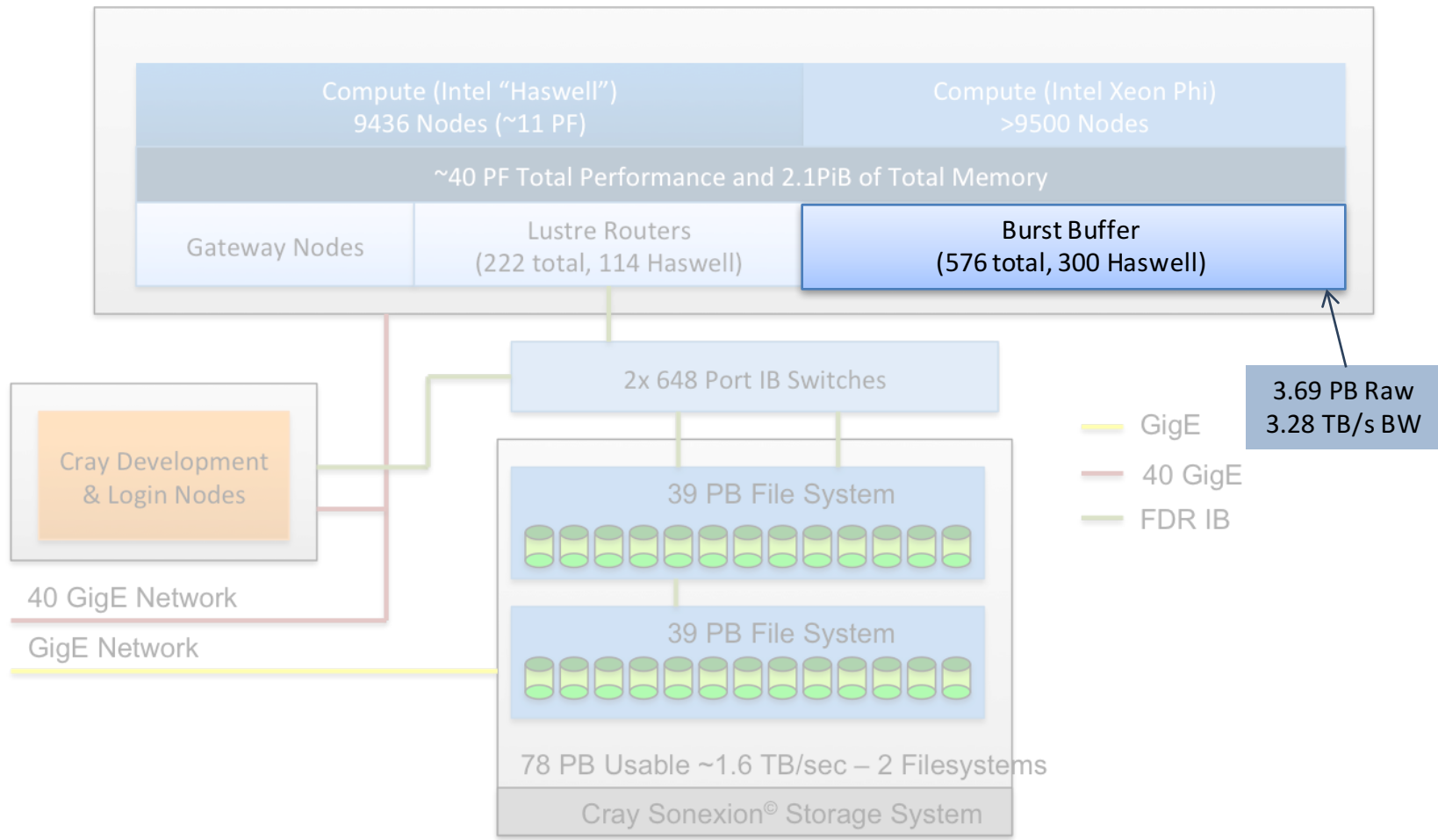


DataWarp/Burst Buffer





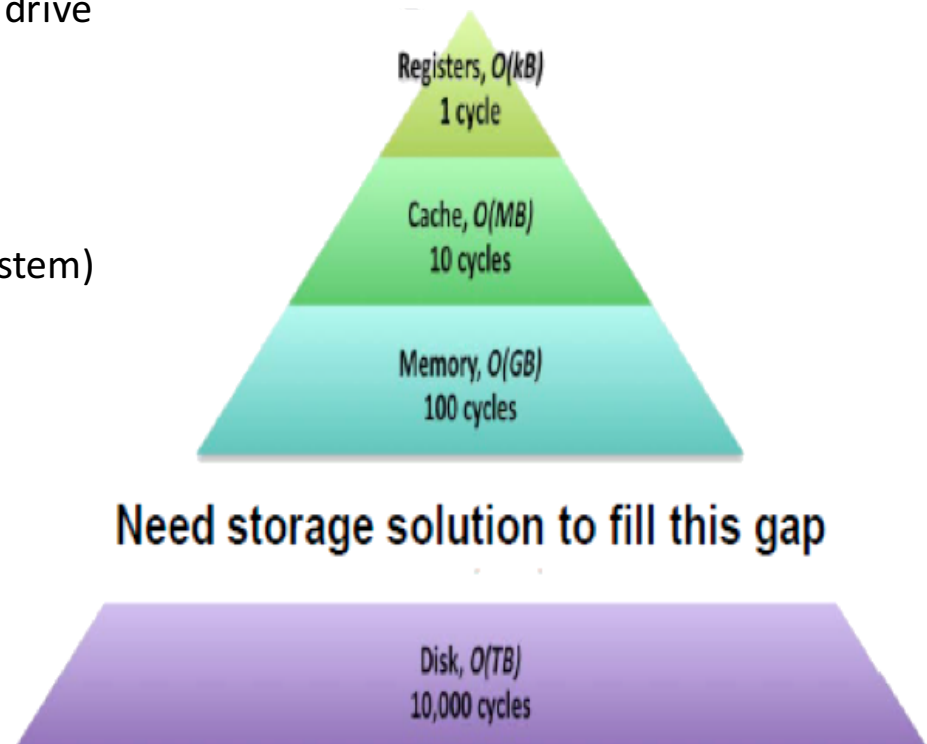
Burst Buffer Configuration





Burst Buffers will improve Productivity and Enable Memory Hierarchy Research

- Technology Drivers:
 - Solid State Disk (SSD) cost decreasing
 - Lower cost of bandwidth than hard disk drive
- Trinity Operational Plans:
 - SSD based 3 PB Burst Buffer
 - 3.28 TB/Sec (2x speed of Parallel File System)
- Burst Buffer will improve operational efficiency by reducing defensive IO time
- Burst Buffer fills a gap in the Memory and Storage Hierarchy and enables research into related programming models





Burst Buffer – more than checkpoint

- Use Cases:
 - Checkpoint
 - In-job drain, pre-job stage, post-job drain
 - Data analysis and visualization
 - In-transit
 - Post-processing
 - Ensembles of data
 - Data Cache
 - Demand load
 - Data staged
 - Out of core data
 - Data intensive workloads that exceed memory capacity

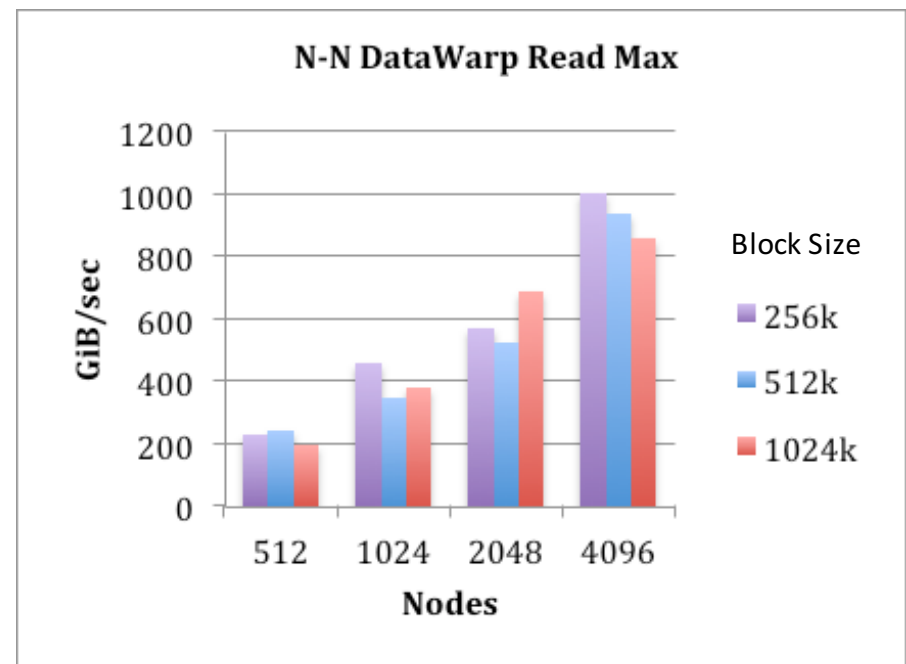
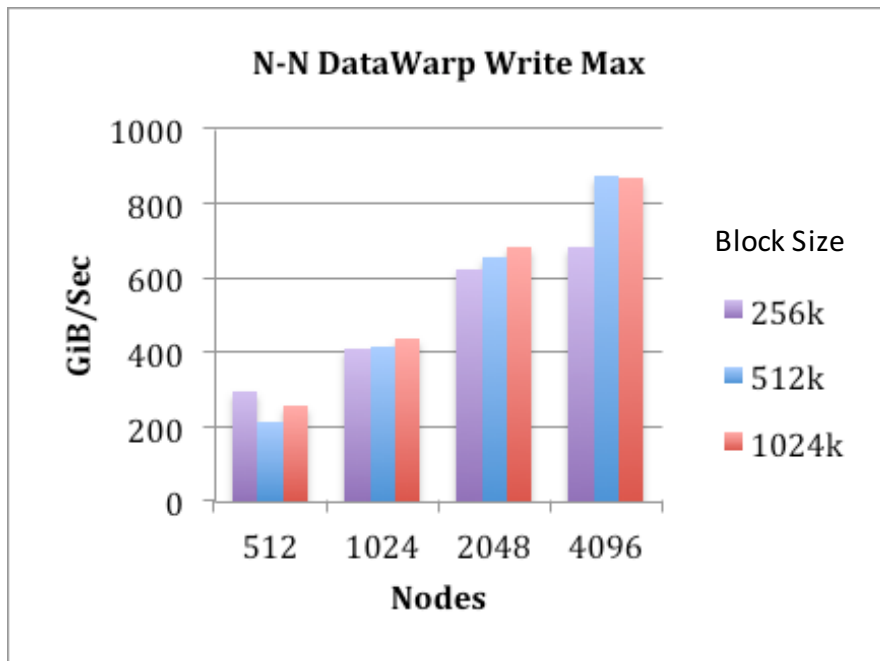


DataWarp Details

- DataWarp nodes built from Cray service nodes
 - 16-core Intel Sandy Bridge with 64 GiB memory
 - Two Intel P3608 SSD cards (4 TB per card)
 - Capacity overprovisioned to get to 10 drive writes per day endurance (standard is 3 DWPD)
- Usage modes
 - Striped scratch
 - Striped private
 - Paging (possible future mode)
 - Cache (possible future mode)
- Integrated with workload manager
 - Stage in/Stage out (single job lifetime)
 - Persistent allocations (accessible by multiple jobs)



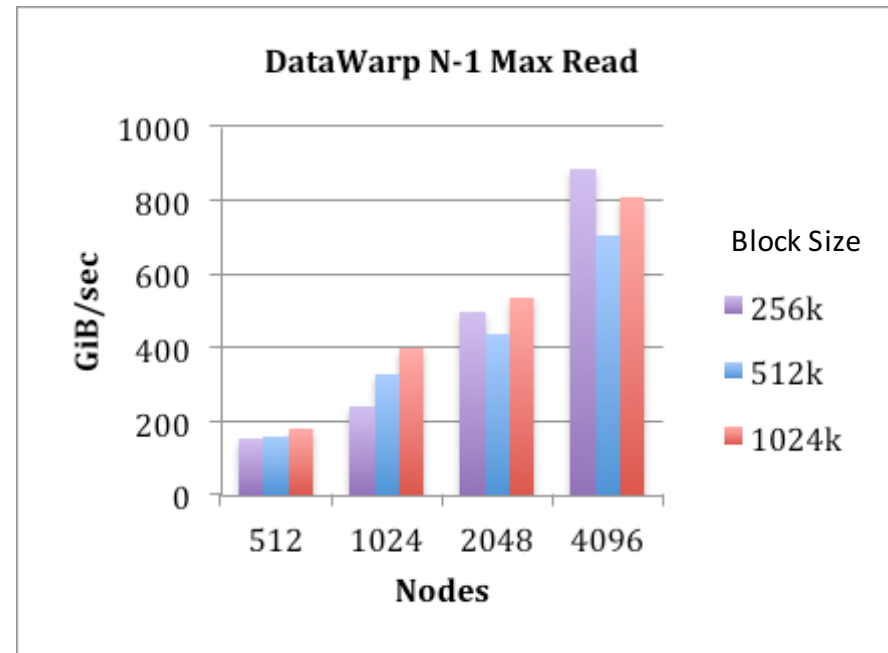
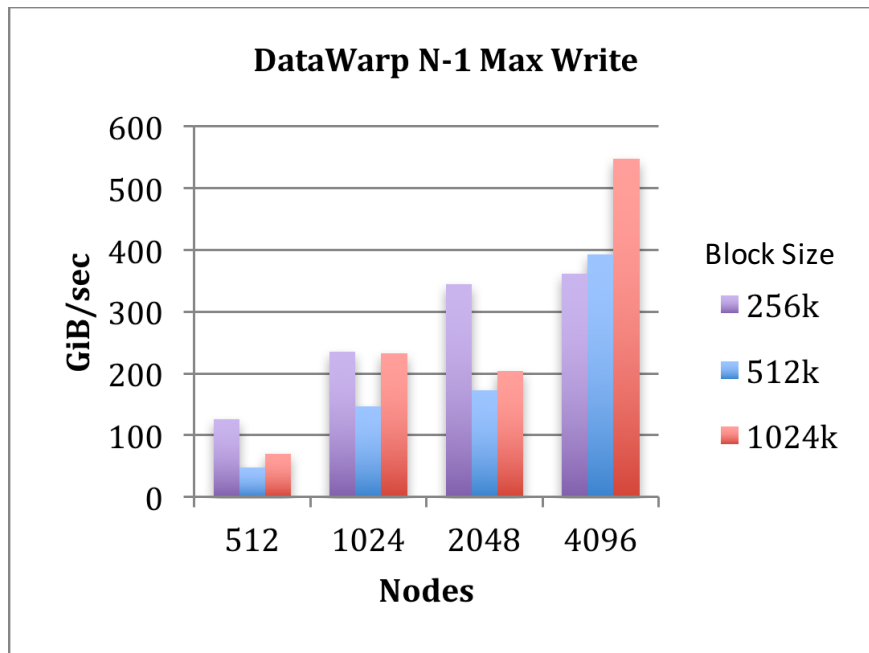
DataWarp N-N



- Test Configuration:
 - 1 reader or writer process per node
 - 32 GiB total data read or written per node
 - The DataWarp allocation striped across all 300 DataWarp nodes



DataWarp N-1



- Test Configuration:
 - 1 reader or writer process per node
 - 32 GiB total data read or written per node
 - The DataWarp allocation striped across all 300 DataWarp nodes



System Management and Integration





ACES/Cray Collaboration

- CLE6.0/SMW 8.0 (Rhine/Redwood)
 - Complete overhaul of the Imaging and Provisioning System
- Early Releases and Collaboration
 - Beta testing with Cray in June 2015
 - LANL was able to provide early feedback to Cray
 - Helped Cray develop a more mature and secure product



Early Experiences with CLE 6.0

- Trinity first to deploy CLE 6.0/SMW 8.0
- How is CLE 6.0 different?
 - Utilizes Ansible for node configuration
 - Utilizes industry standard Linux tools
 - Configurator tool to manage system configuration



Early Experiences with CLE 6.0

- Pre-Release Evaluation and Preparation
 - Significant time investment required for an install
 - SMW and Boot RAID must be reformatted (no upgrade path)
 - Configurator
 - Question and Answer interface for filling out system configuration
 - Tedious and cumbersome to use
 - Worksheets in later beta versions
 - Can be prepared ahead of time
 - For a large system this takes a considerable amount of time
 - Better than using Configurator



Early Experiences with CLE 6.0

- Configuration Management
 - Using Ansible effectively
 - Use Cray's Ansible site-hooks to fully prescribe the machine
 - Can break the boot process
 - Causes the boot process to run longer
 - Only runs at boot time
 - Separate Local Ansible plays developed by admins
 - Can be run via cron or at job epilogues
 - Cray's Ansible plays are lengthy and resource-intensive
 - Playing nicely with Cray's Ansible plays
 - Difficult to manage files that Cray also wants to manage
 - Workarounds in place, but is still an ongoing issue



Early Experiences with CLE 6.0

- External Login Nodes
 - Replacement for Bright
 - Utilizes OpenStack
 - Commonality Between Internal login and eLogin
 - Builds eLogin images from same source
 - Uses the same Programming Environment
 - OpenStack Concerns
 - Harder to manage and debug OpenStack
 - Securing OpenStack can be a challenge



Integrating New Technologies

- Sonexion 2000
 - Lustre Appliance
 - First deployment of Distributed Namespace (DNE Phase 1)
 - Multiple MDT for better metadata performance
 - Directories on MDTs created for users on a case by case basis
 - Continually working with Seagate to fix issues
- DataWarp
 - Learning how to manage DataWarp
 - Debugging when things go wrong is a challenge
 - Many challenges integrating DataWarp with Adaptive's Moab scheduler



Current Challenges

- Debugging boot failures
 - It is almost always Ansible that fails
 - Sometimes rerunning Ansible will fix it
 - Some Ansible logs are only on the end node
 - If the node's ssh is not configured yet it can be difficult to get to the logs
 - Cray's xtcon can work but only if there is a password set
- DataWarp at Scale
 - Testing done mostly on smaller systems
 - Seeing issues with stage-out performance to Lustre
 - Communication issues with Moab and DataWarp under high load
 - Currently ssh, but a RESTful interface has been requested



Ongoing Collaboration with Cray

- CLE 6.0 UP00 to CLE 6.0 UP01
 - UP01 will be the first public release of CLE 6.0/SMW 8.0
 - Many of the bugs and enhancements requested will be in the new release
 - UP01 required for KNL deployment in Phase 2
 - Installation of UP01 on LANL TDS systems end of May



Trinity Center of Excellence





Trinity Advanced Technology System

COMPUTE NODES

Intel "Haswell" Xeon E5-2698v3	Intel Xeon Phi "Knights Landing"
9436 nodes	> 9500 nodes
Dual socket, 16 cores/socket, 2.3 GHz	1 socket, 60+ cores, > 3 Tflops/KNL
128 GB DDR4	96 GB DDR4 + 16GB HBM

#6 on Top500
November 2015
8.1 PFlops
(11 PF Peak)



Cray Aries 'Dragonfly' Interconnect

Advanced Adaptive Routing
All-to-all backplane & between groups



Cray Sonexion Storage System

78 PB Usable, ~1.6 TB/s



Cray DataWarp

576 Burst Buffer Nodes
3.7 PB, ~3.3 TB/s



Trinity - Performance (Portable) Challenges

COMPUTE NODES

Intel "Haswell" Xeon E5-2698v3

9436 nodes

Dual socket, 16 cores/socket,
2.3 GHz

128 GB DDR4

#6 on Top500
November 2015
8.1 PFlops
(11 PF Peak)

Intel Xeon Phi "Knights Landing"

> 9500 nodes

1 socket, 60+ cores,
> 3 Tflops/KNL

96 GB DDR4 +
16GB HBM

- **Enabling (not hindering) Vectorization**
- **Increase parallelism, cores/threads**
- **High Bandwidth Memory**
- **Burst Buffer – reduce I/O overhead**



Cray Aries 'Dragonfly' Interconnect

Advanced Adaptive Routing
All-to-all backplane & between groups



Cray Sonexion Storage System

78 PB Usable, ~1.6 TB/s

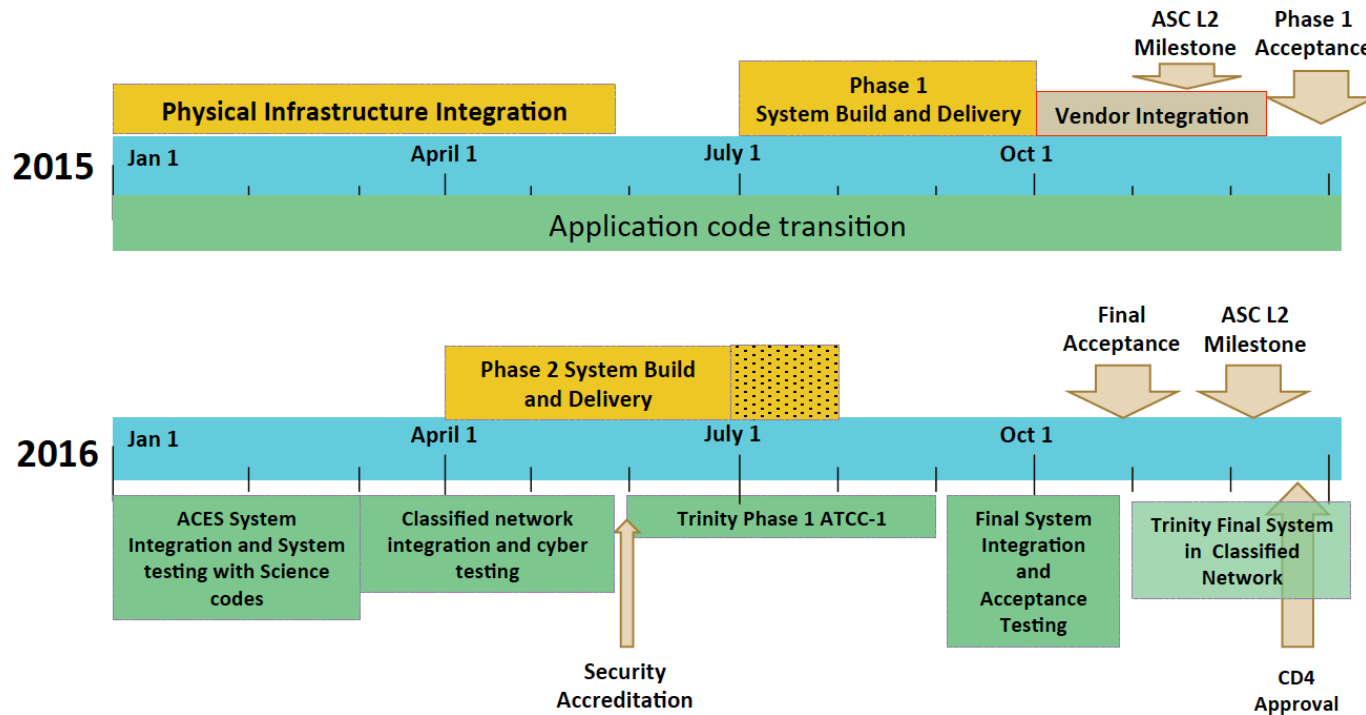


Cray DataWarp

576 Burst Buffer Nodes
3.7 PB, ~3.3 TB/s



Access to Early HW/SW



- Application Regression Test Beds x2 (Cray) ~100 nodes (June 2015), Software Dev. Testbed < 100 nodes – Phase I, upgrades for Phase II
- White Boxes (Intel) ~ few nodes (Sept 2015/April 2016)



COE Collaborations

- Cray

- John Levesque (50%)
- Jim Schwarzmeier (20%)
- Gene Wagenbreth (100%) - new
- Mike Davis (SNL), Mike Berry (LANL) on-site analyst
- SMEs (Performance & Tools)
- Acceptance team

- Intel

- Ron Green, on-site analyst (SNL/LANL)
- Discovery Session, Dungeons - SMEs

- ASC codes are often export controlled, large and complex = a lot of paperwork
- Embedded vendor support/expertise is needed = US citizenship
- Original projects focus on a single code/lab



CoE Projects/Highlights

- SNL
 - Focused on preparing the Sierra engineering analysis suite for Trinity
 - Proxy Codes: miniAero (explicit Aerodynamics), miniFE (implicit FE), miniFENL, BDDC (Domain Decomp. Solver)
 - ‘Super’ Dungeon Session including
 - More realistic code/stack
 - NALU (proxy application for FEM assembly for low Mach CFD) + Trilinos multi-grid solver, Kokkos + BDDC
 - 6 weeks preparation leading up to Dungeon session
 - Expose Intel to ‘real’ codes & issues – long compile times, long tools analysis times, compiler issues, MKL issues.
 - Great for relationship/collaboration building
 - More embedded support from Cray (Gene Wagenbreth, March 2016)



CoE Projects/Highlights

- LLNL
 - Developed Proxy Code: Quicksilver (Monte Carlo transport)
 - Dynamic neutron transport problem (MPI or MPI+threads)
 - Use in performance portability activities
 - Proxy codes are not an example of efficient source code, rather a representation of a larger application
 - Discovery Sessions (x2) with proxy applications & performance portable abstraction layer

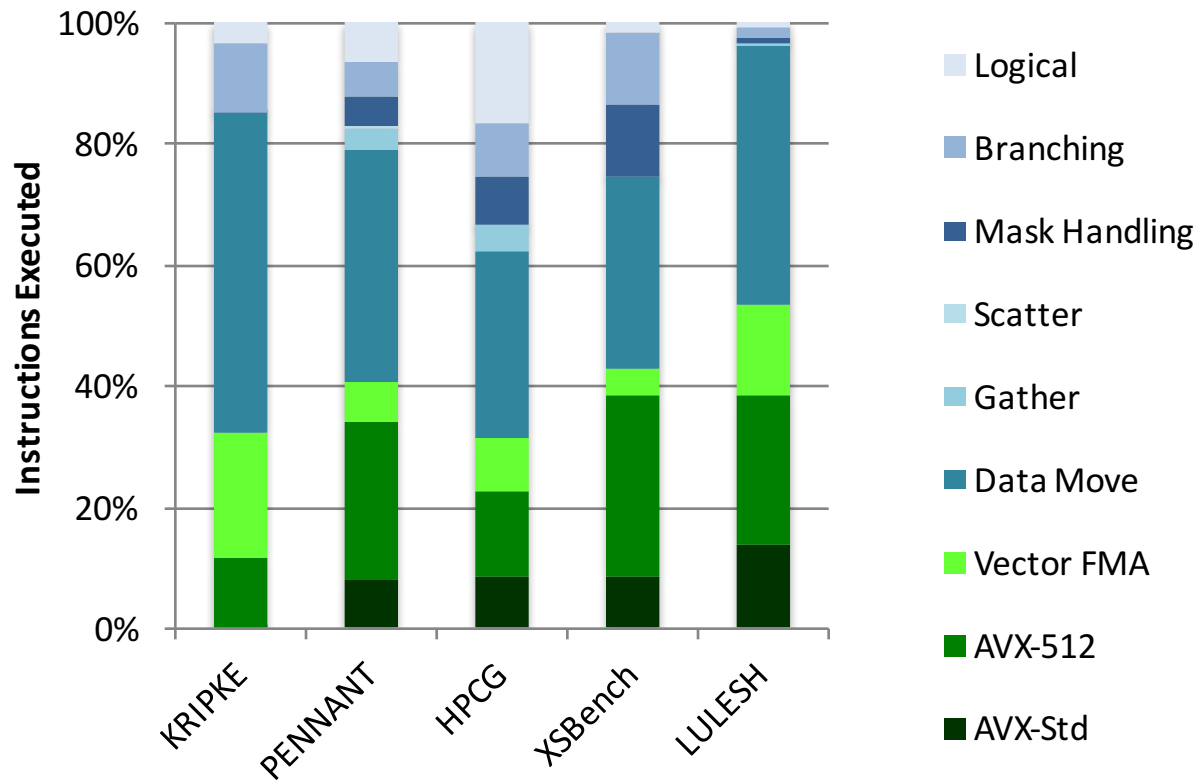


CoE Projects/Highlights

- LANL
 - Full application exploration – very large, multi-physics, multi-material AMR application (MPI-only)
 - Discovery session (Intel) & Deep dive (Cray) – on-site
 - Prototyping SPMD in radiation diffusion package as an option in code threading implementation
 - Addressing performance bottlenecks in solvers library (HYPRE) & code
 - Addressing technical debt
 - Broadening scope of COE projects to include deterministic Sn transport (full application and proxy)
 - Discovery sessions & deep dive activities



Vectorization



AVX-512 Vectorization Levels in DOE Benchmarks and Mini-Apps



Experiences on KNL

- **Initial work on KNL with mini-applications and some performance kernels (from Trillinos) going very well**
- For some applications, greater improvement than the hardware specifications moving between memory
- Strongest application performance for some kernels on any GA-hardware we have ever seen
- API (memkind) bring up going well but we expect this to be low-level (users do not like this and want it hidden away)
- Lots under NDA but results will most likely be shown at ISC'16



Questions?

