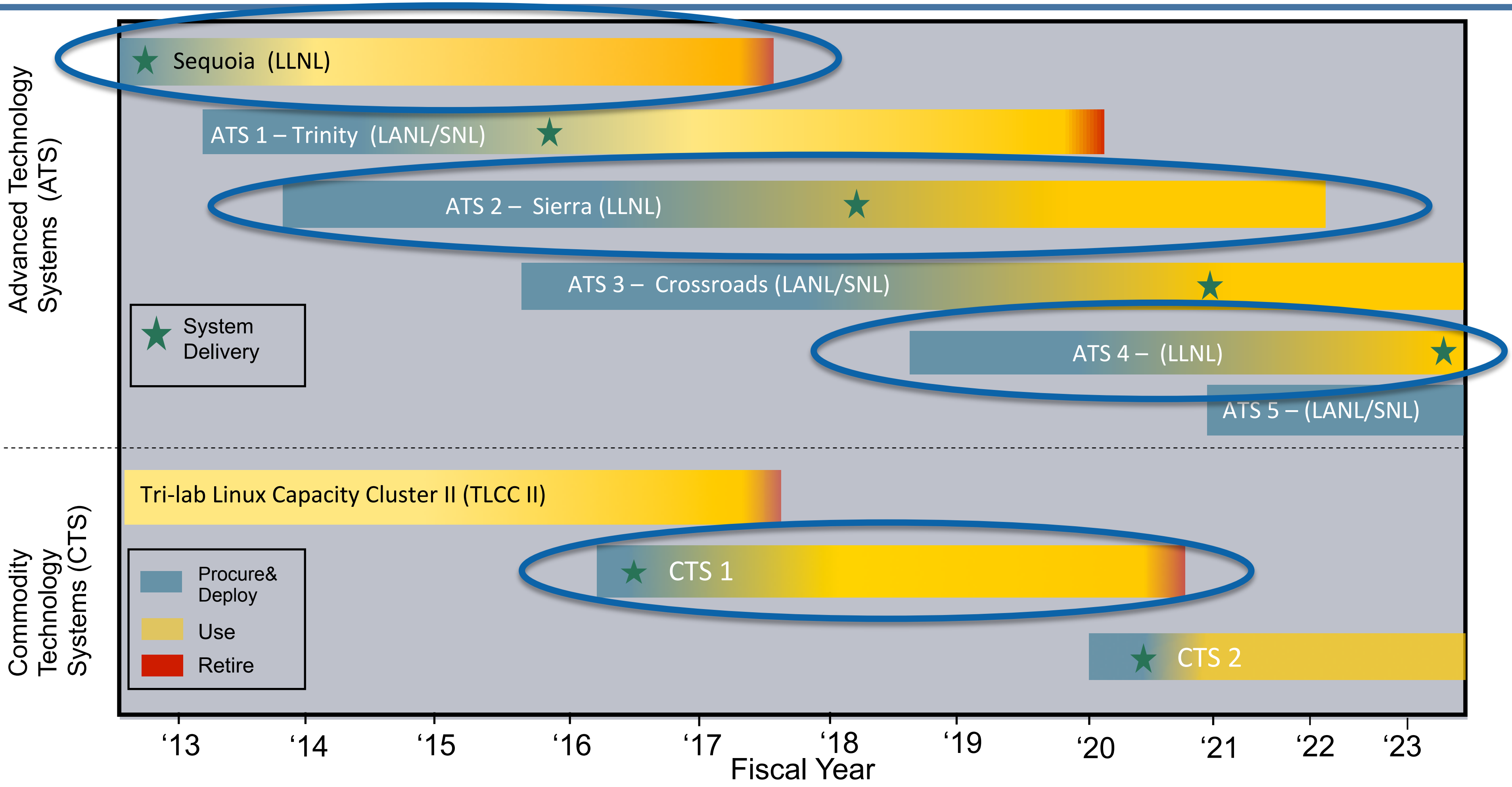
Optimizing Network Usage on Sequoia and Sierra

Bronis R. de Supinski
Chief Technology Officer
Livermore Computing

June 23, 2016

LLNL-PRES-695482
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

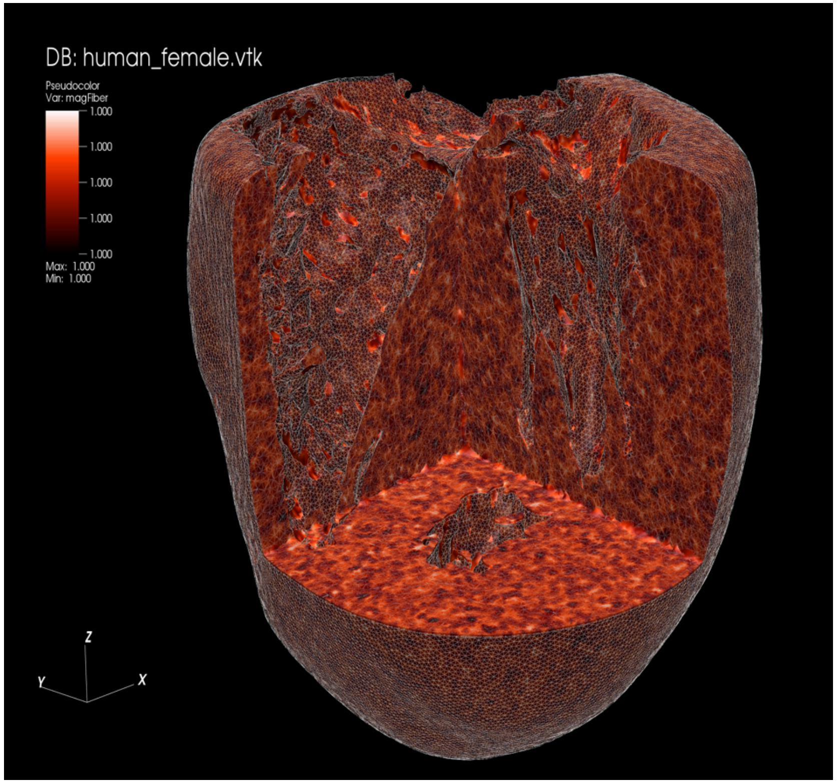
My focus is NNSA ASC ATS platforms at LLNL

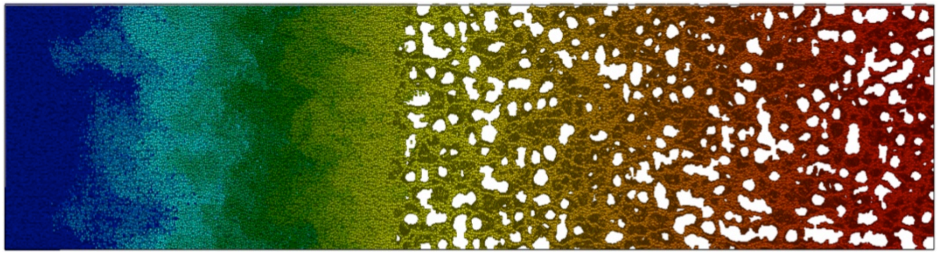


Sequoia and Sierra are the current and next-generation Advanced Technology Systems at LLNL

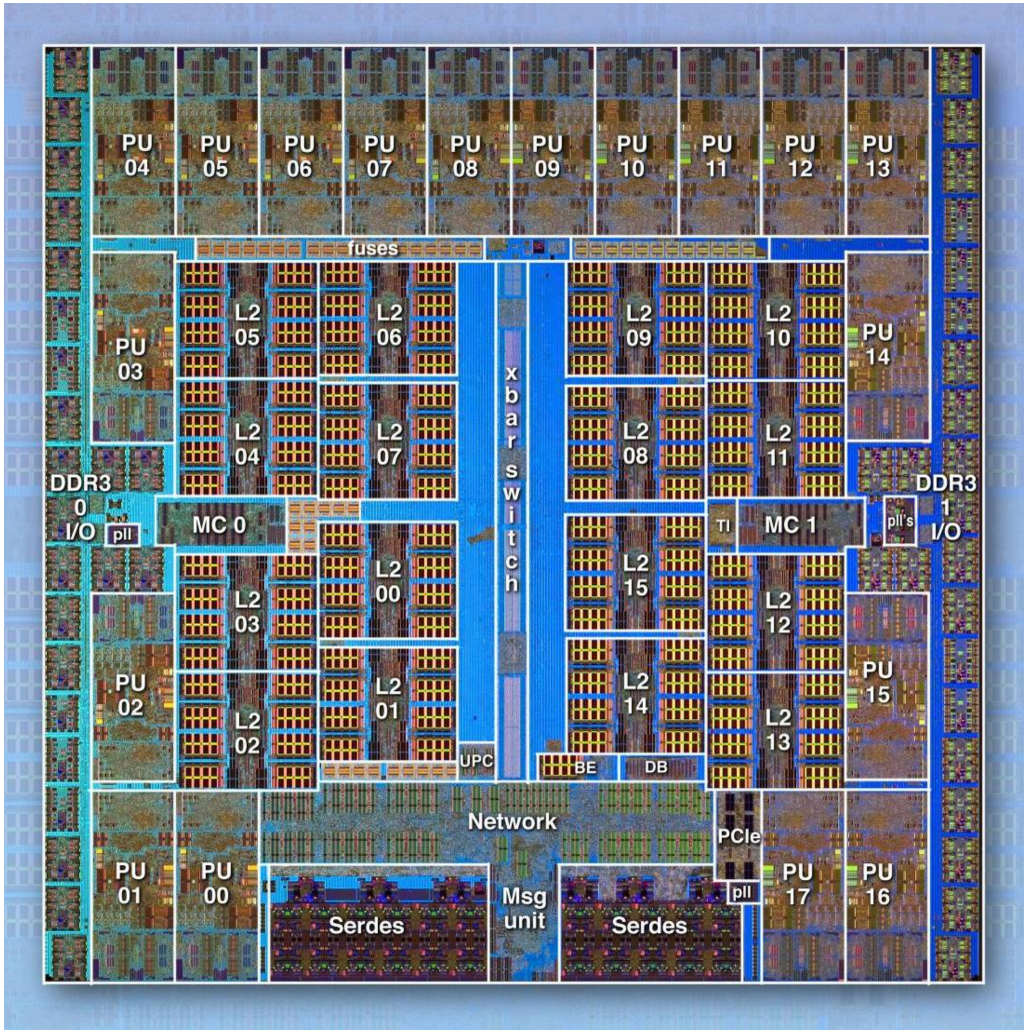
Sequoia provides previously unprecedented levels of capability and concurrency

- Sequoia statistics
  - 20 petaFLOP/s peak
  - 17 petaFLOP/s LINPACK
  - Memory 1.5 PB, 4 PB/s bandwidth
  - 98,304 nodes
  - 1,572,864 cores
  - 3 PB/s link bandwidth
  - 60 TB/s bi-section bandwidth
  - 0.5−1.0 TB/s Lustre bandwidth
  - 50 PB disk
- 9.6MW power, 4,000 ft^2
- Third generation IBM BlueGene





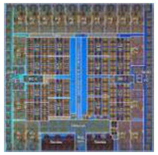
The BG/Q compute chip integrates processors, memory and networking logic into one chip



- 16 user + 1 OS + 1 redundant cores
  - 4-way multi-threaded, 1.6 GHz 64-bit
  - 16kB/16kB L1 I/D caches
  - Quad FPUs (4-wide DP SIMD)
  - Peak: 204.8 GFLOPS @ 55 W
- Shared 32 MB eDRAM L2 cache
  - Multiversioned cache
- Dual memory controller
  - 16 GB DDR3 memory (1.33 Gb/s)
  - 2 * 16 byte-wide interface (+ ECC)
- Chip-to-chip networking
  - 5D Torus topology + external link
  - Each link 2 GB/s send + 2 GB/s receive
  - DMA, put/get, collective operations

Traditional BlueGene overall system integration results in small footprint

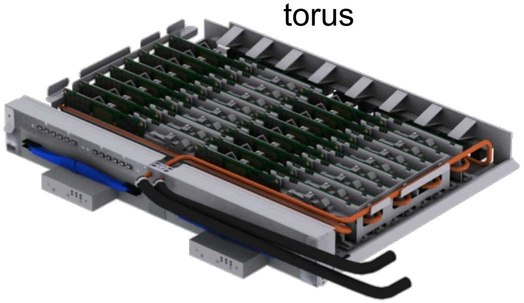
1. Chip
16 cores



2. Module
Single chip

3. Compute card
One single chip module,
16 GB DDR3 memory

4. Node card
32 compute cards,
Optical modules, link chips,
torus

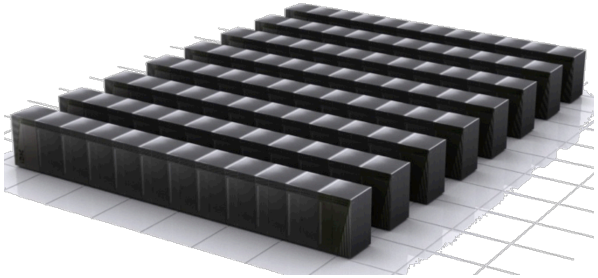


5b. I/O drawer
8 I/O cards
8 PCIe Gen2 slots

5a. Midplane
16 node cards

6. Rack
2 midplanes
1, 2, or 4 I/O drawers

7. System
20 PF/s



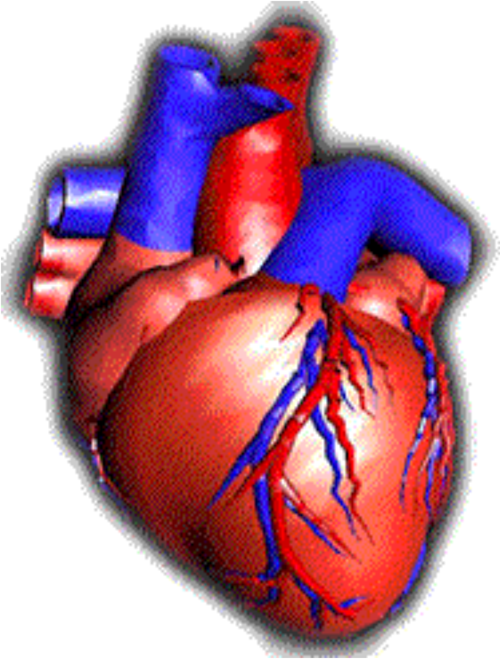
Sequoia and BlueGene/Q reflect lessons from previous BlueGene generations

- Communication locality through optimized MPI process placement is critical on 3D torus networks
  - Use of 5D torus reduces network diameter and reduces the importance of MPI process placement
- Support for hardware optimized collectives should apply to subcommunicators as well as global operations
  - Increased network communication contexts allows more applications to exploit hardware support for collective operations
- Hardware support for network partitioning minimizes jitter
- Multiple networks provide many benefits but also increase costs

These examples are network-centric; others reflect lessons throughout the system hardware and software architecture

LLNL, with IBM, has developed Cardiod, a state-of-the-art cardiac electrophysiology simulation

- Mechanisms that lead to arrhythmia are not well understood; Contraction of heart is controlled by electrical behavior of heart cells



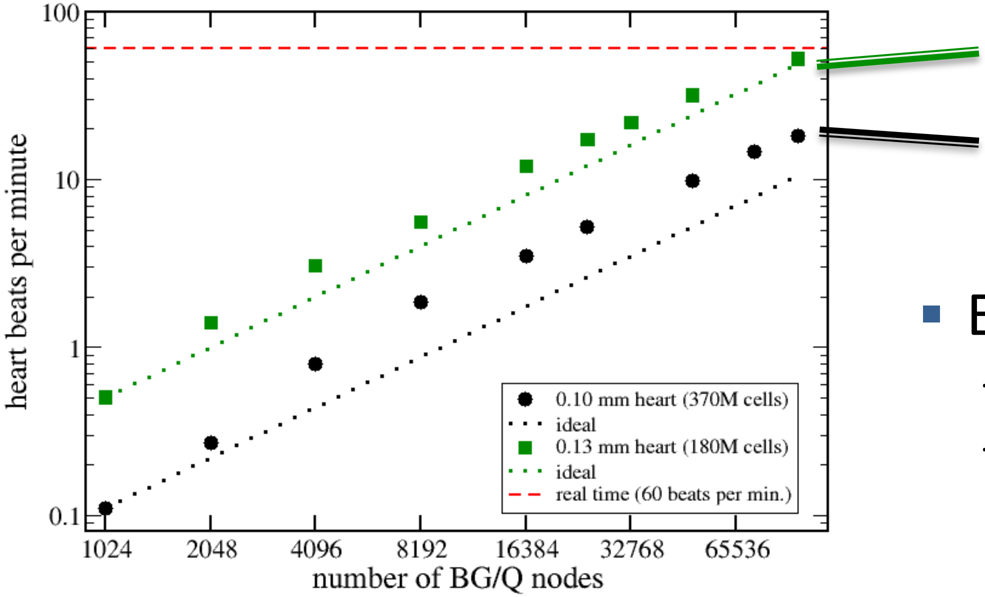
- Mathematical models reproduce component ionic currents of the action potential
  - System of non-linear equation ODEs
  - TT06 includes 6 species and 14 gates
  - Negative currents depolarize (activate)
  - Positive currents repolarize (return to rest)

- Ability to run, at high resolution, thousands instead of tens of heart beats enables detailed study of drug effects

2012 Gordon Bell finalist typifies the effort
required to exploit supercomputer capability fully

Cardoid achieves outstanding performance that enables nearly real-time heart beat simulation

- Measured peak performance: 11.84 PFlop/s (58.8% of peak)
  - — 0.05 mm resolution heart (3B tissue cells)
  - — Ten million iterations, dt = 4 usec
  - — Performance of full simulation loop, including I/O, measured with HPM



60 beats in 67.2 seconds

60 beats in 197.4 seconds

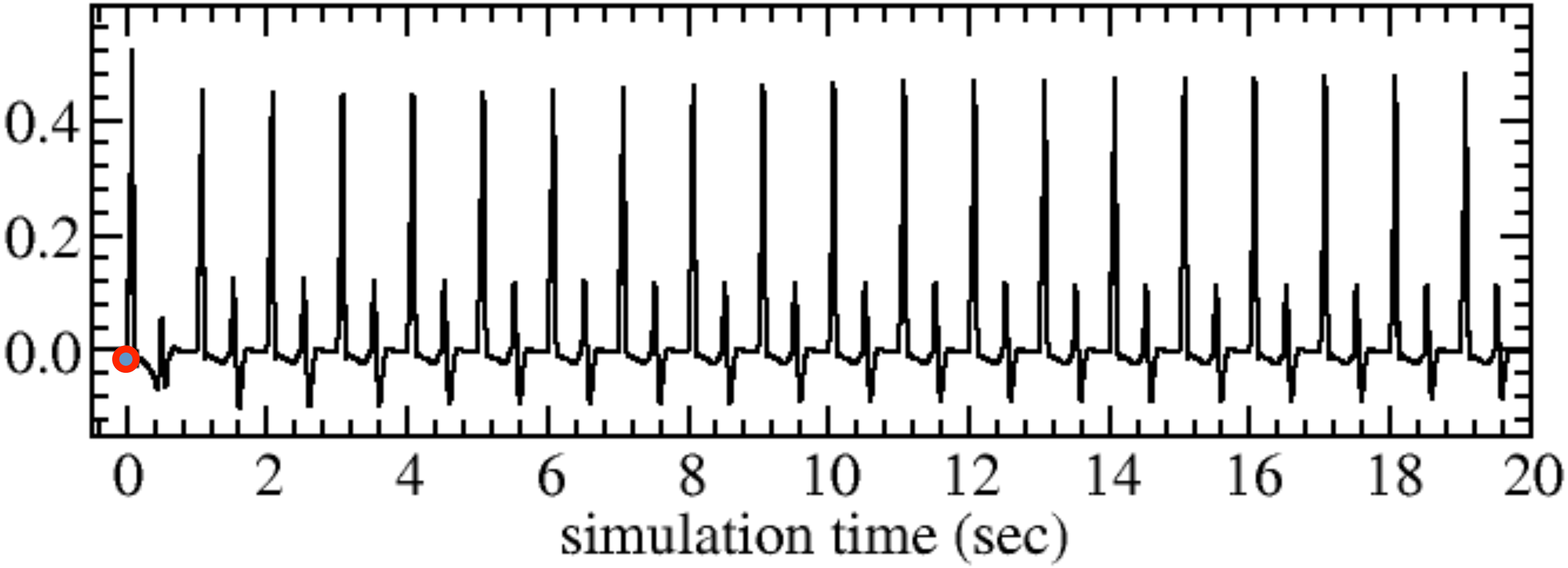
- Extreme strong scaling limit:
  - — 0.10 mm: 236 tissue cells/core.
  - — 0.13 mm: 114 tissue cells/core

Optimized Cardioid is 50x faster than “naive” code

Cardiod represents a major advance in the state of the art of human heart simulation

One minute of wall time

0.1 mm heart (370M tissue cells)



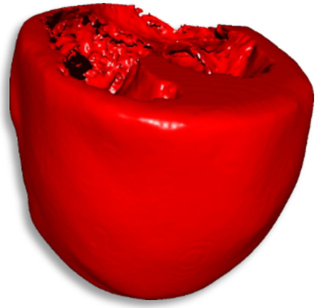
- ○ Cardioid
- ● Previous state of the art

18.2 seconds of simulation time

Cardoid achieves outstanding performance through detailed tuning to Sequoia’s architecture

- Partitioned cells over processes with an upper bound on time (not on equal time)
- Assigned diffusion work and reaction work to different cores
- Transformed the potassium equation to remove serialization
- Expensive 1D functions in reaction model expressed with rational approximates
- Single precision weights to reduce diffusion stencil use of L2 bandwidth
- Hand unrolled to SIMDize loops over cells
- Sorted by cell type to improve SIMDization
- Sub-sorting of cells to increase sequential/ vector load and storing of data.
- log function from libm replaced with custom inlined functions
- On the fly assembly of code to optimize data movement at runtime
- Memory layout tuned to improve cache performance
- Use of vector intrinsics and custom divides
- Moved integer operations to floating point units to exploit SIMD units
- **No explicit network barrier**
- L2 on node thread barriers
- **Use low level SPI for halo data exchange between tasks (DMA)**
- Application managed threads
- SIMDized diffusion stencil implementation
- Zero flux boundary conditions approximated by method with no global solve
- **High performance I/O is aware of BG/Q network topology**
- Low overhead in-situ performance monitors
- Assignment of threads to diffusion/reaction dependent on domain characteristics
- Co-scheduled threads for improved dual issue
- Multiple diffusion implementations to obtain optimal performance for various domains
- Remote & local copies separated to improve bandwidth utilization

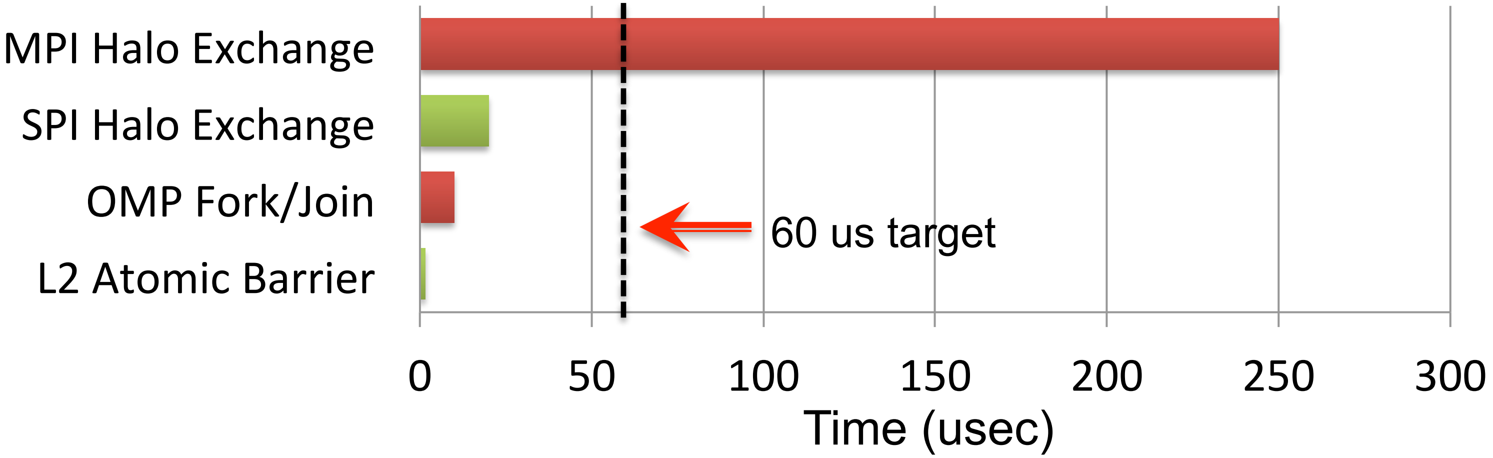
At largest scales, small software overheads can significantly impact performance



370 Million Cells 1.6 Million Cores

1600 Flops/cell

60 us per iteration

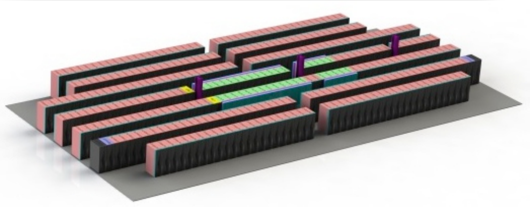


Direct use of message units and L2 atomic operations minimizess overhead

The Sierra system that will replace Sequoia features a GPU-accelerated architecture

Compute System

2.1 – 2.7 PB Memory
120 -150 PFLOPS
10 MW



Compute Rack

Standard 19”
Warm water cooling

Compute Node

POWER® Architecture Processor
NVIDIA®Volta™
NVMe-compatible PCIe 800GB SSD
> 512 GB DDR4 + HBM
Coherent Shared Memory

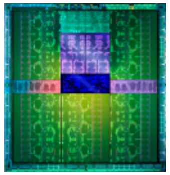
Components

IBM POWER

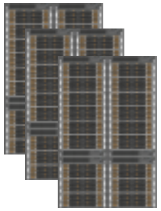
- NVLink™

NVIDIA Volta

- HBM
- NVLink



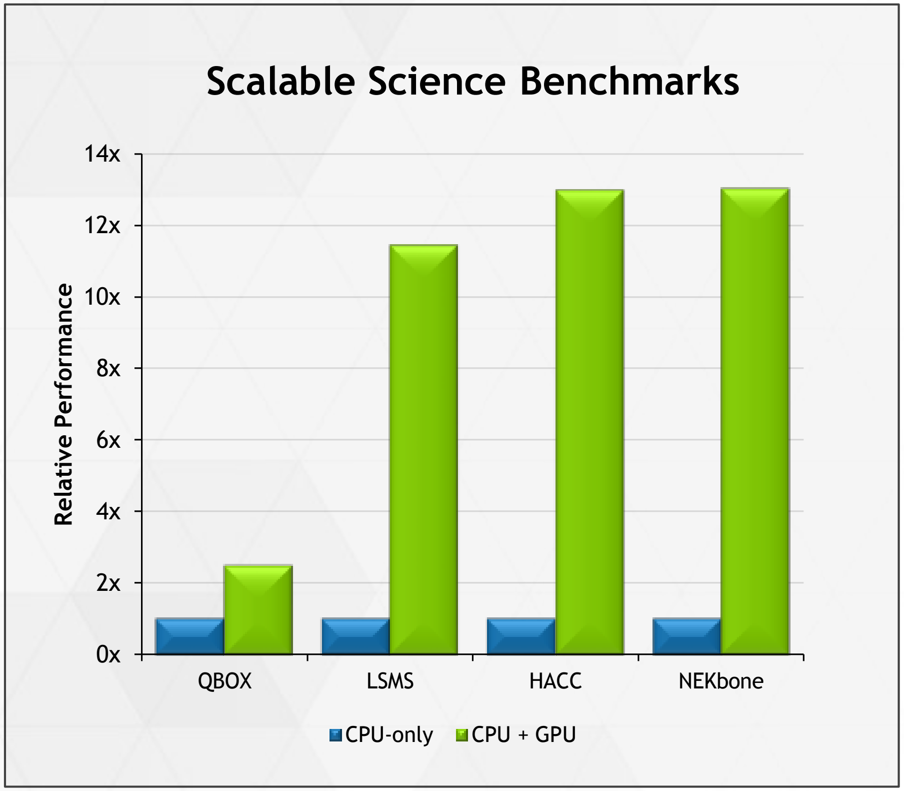
Mellanox® Interconnect
Dual-rail EDR Infiniband®

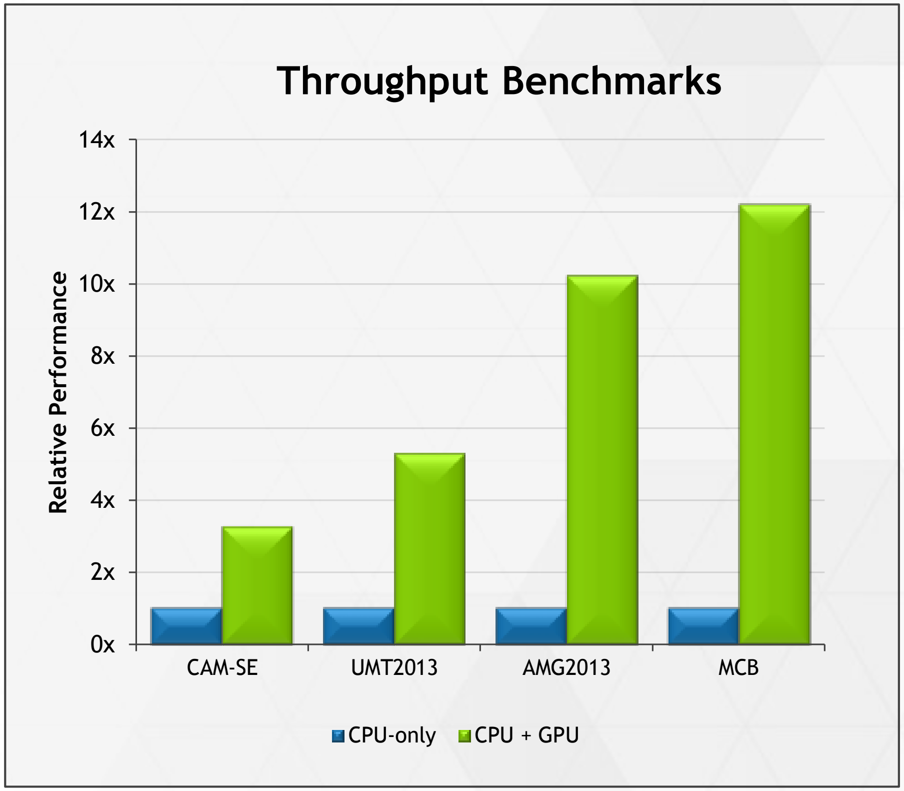


GPFS™ File System

120 PB usable storage
1.2/1.0 TB/s R/W
bandwidth

Outstanding benchmark analysis by IBM and NVIDIA demonstrates the system's usability





Projections included code changes that showed tractable annotation-based approach (i.e., OpenMP) will be competitive

Sierra NRE will provide significant benefit to the final system

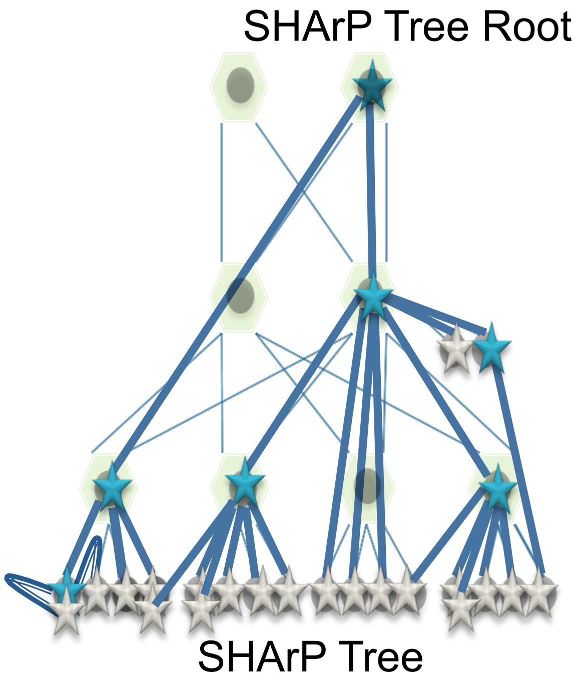
- Center of Excellence
- Motherboard design
- Water cooled compute nodes
- HW resilience studies/ investigation (NVIDIA)
- **Switch based collectives**
- **Hardware tag matching**
- **GPU Direct and NVMe**
- Open source compiler infrastructure
- System diagnostics
- System scheduling
- Burst buffer
- GPFS performance and scalability
- Cluster management
- Open source tools

Switch-based support for collectives further improves critical functionality

SHArP

Scalable Hierarchical Aggregation Protocol

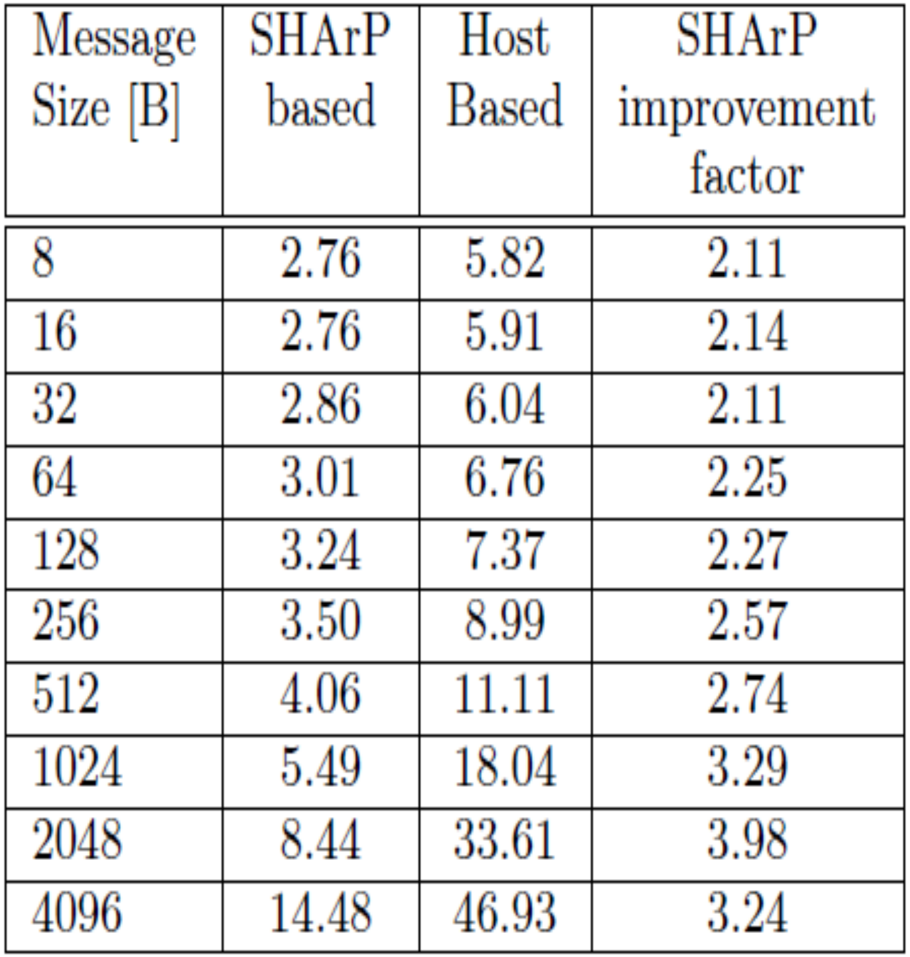
- Reliable, scalable, general purpose primitive, applicable to multiple use cases
  - In-network Tree based aggregation mechanism
  - Large number of groups
  - Multiple simultaneous outstanding operations
- High performance collective offload
  - Barrier, Reduce, All-Reduce
  - Sum, Min, Max, Min-loc, max-loc, OR, XOR, AND
  - Can overlap communication and computation
- Flexible mechanism reflects lessons learned from BlueGene systems



★ SHArP Tree Aggregation Node (Process running on HCA)

☆ SHArP Tree Endnode (Process running on HCA)

Initial results demonstrate that SHArP collectives improve performance significantly



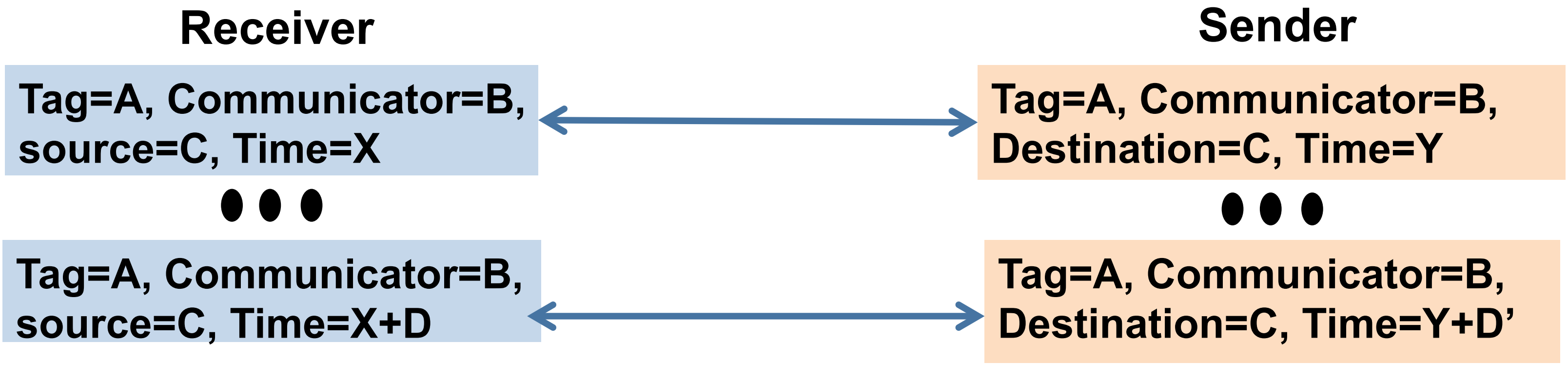
Message Size [B]	SHArP based	Host Based	SHArP improvement factor
8	2.76	5.82	2.11
16	2.76	5.91	2.14
32	2.86	6.04	2.11
64	3.01	6.76	2.25
128	3.24	7.37	2.27
256	3.50	8.99	2.57
512	4.06	11.11	2.74
1024	5.49	18.04	3.29
2048	8.44	33.61	3.98
4096	14.48	46.93	3.24

- OSU Allreduce 1PPN, 128 nodes

As seen with Cardoid, MPI software overhead critically limits realized network performance

- Realistic applications, particularly in C++ often use small messages
  - Realized message rate often the key performance indicator
  - MPI provides little ability to COALESCE these messages

- MPI matching rules heavily impact realized message rate
  - Message envelops must match
    - Wild cards (MPI_ANY_SOURCE, MPI_ANY_TAG) increase envelop matching complexity and, thus, cost
  - Posted received must be matched in-order against the in-order posted sends



Receiver

Tag=A, Communicator=B, source=C, Time=X

• • •

Tag=A, Communicator=B, source=C, Time=X+D

Sender

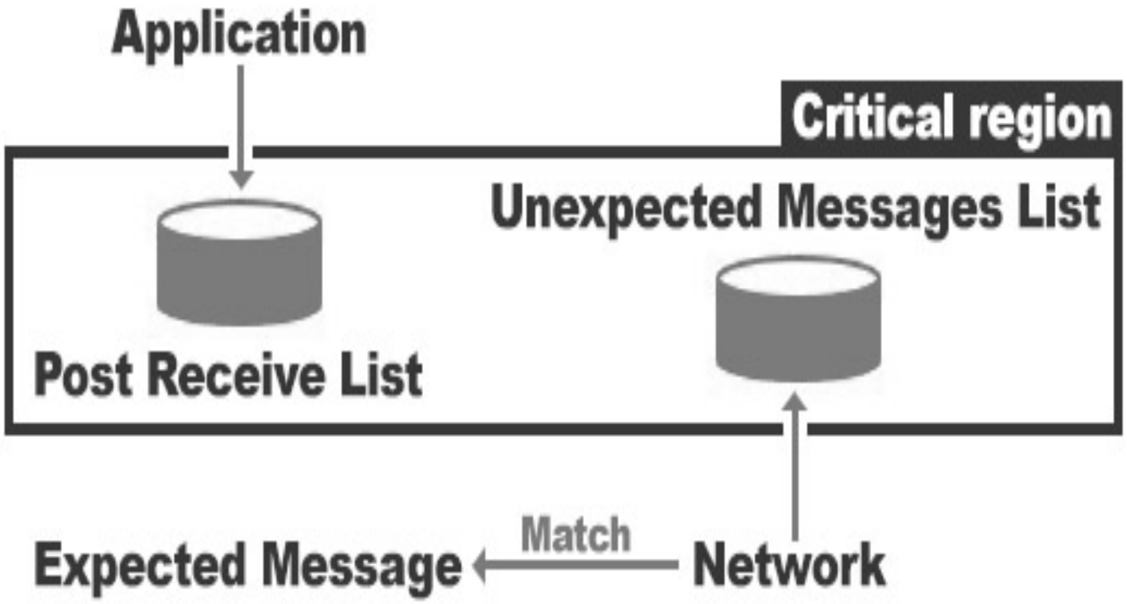
Tag=A, Communicator=B, Destination=C, Time=Y

• • •

Tag=A, Communicator=B, Destination=C, Time=Y+D'

Hardware message matching support can alleviate software overhead

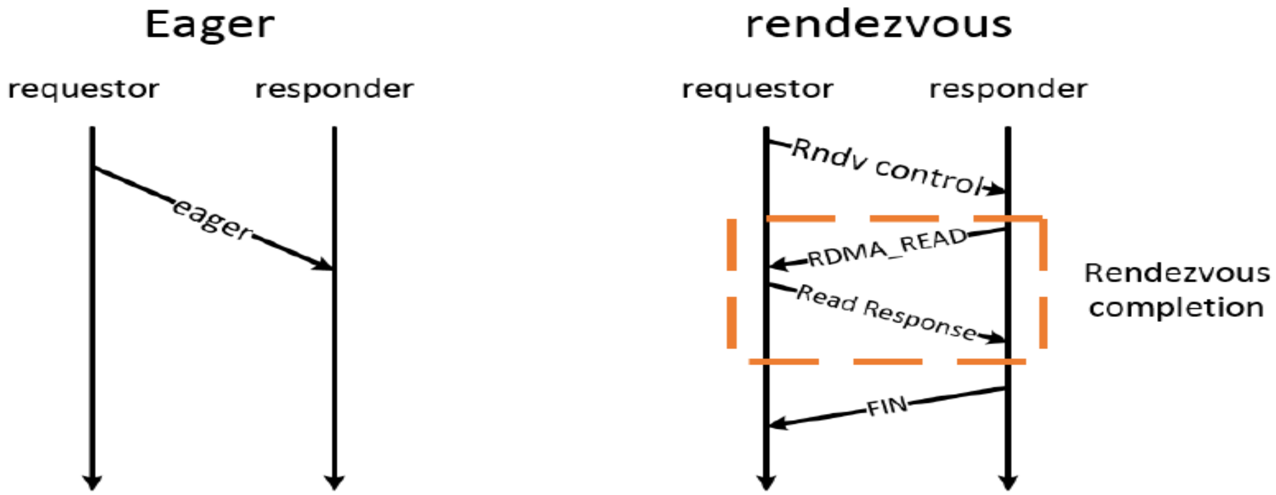
MPI tag matching operations must appear to be performed atomically



- Complexity/serialization of message matching limits the processing that can be performed on GPUs

Mellanox hardware will support efficient MPI tag matching

- Offloaded to the ConnectX-5 HCA
  - Enables more of hardware bandwidth to translate to realized message rate
  - Full MPI tag matching as compute progresses
  - Rendezvous offload: large data delivery as compute progresses
- Control can be passed between hardware and software
- Verbs tag matching support is being up-streamed

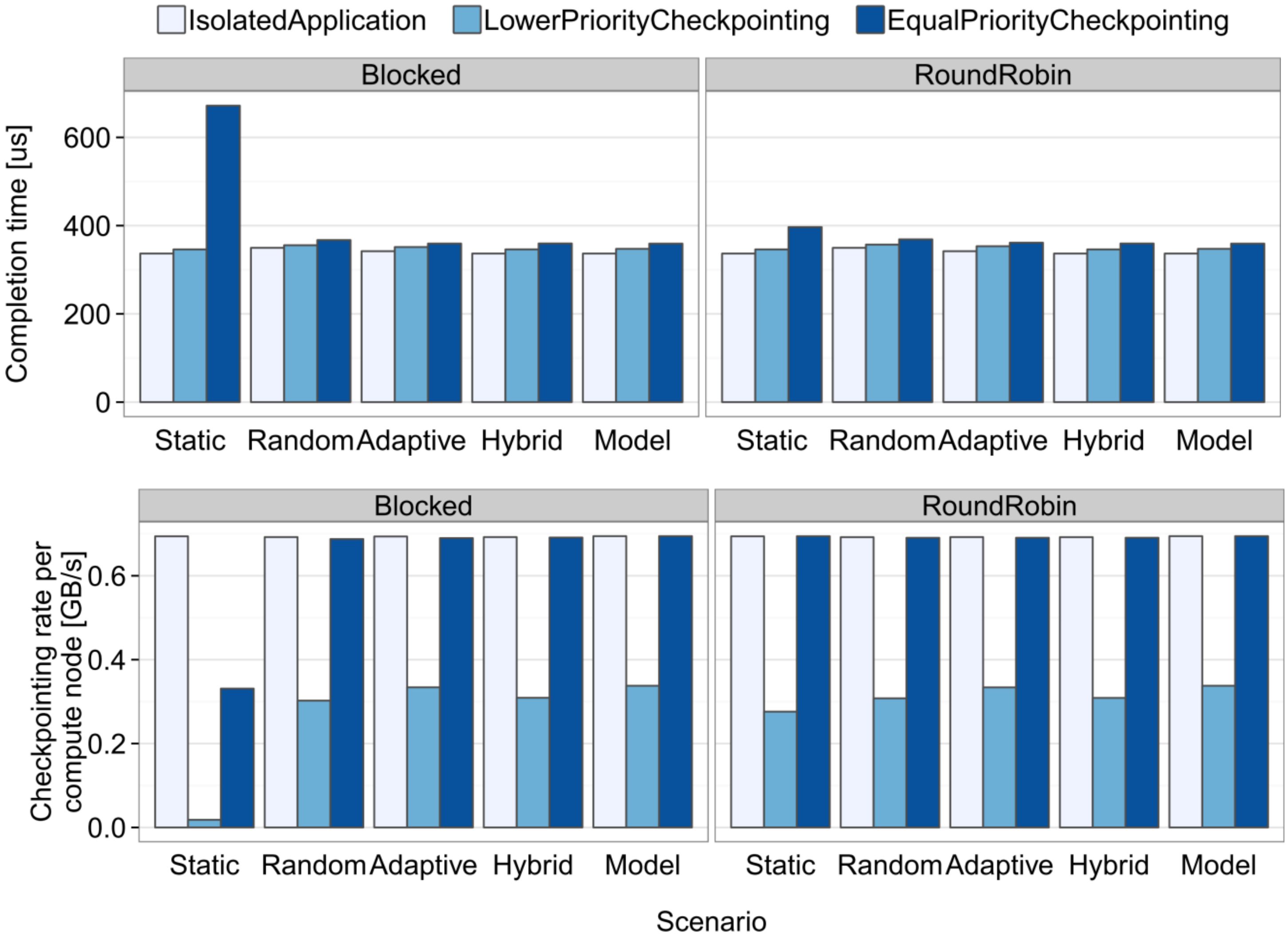


Deploying multiple networks exacerbates network hardware costs, which are already too high in large-scale systems

- Sierra uses commodity cluster network solution
  - Separate management (ethernet) and user (IB) networks
  - Single network for user traffic (point-to-point, collectives & file system traffic)
- A single network for user traffic saves money but has other costs
  - Jitter impact of other jobs' file system traffic can be severe
  - Burst buffer strategy smooths file system bandwidth demand
    - File system traffic of a job now competes with its MPI traffic
- Different types of network traffic are not equally critical
  - File system traffic "only" needs a guarantee of eventual completion
  - Collectives often critically limit overall performance
  - Other traffic classes also exist

Quality-of-Service (QoS) mechanisms are necessary to achieve a network solution that reduces network hardware costs while providing acceptable, consistent performance for all traffic classes

Preliminary results indicate that IB priority levels compensate for checkpoint traffic



Sierra network hardware addresses lessons learned from previous LLNL systems

- Flexibility of SHArP switch-based collectives will accelerate subcommunicators collectives and will allow jobs to share network
- HCA MPI tag matching will reduce software cost on critical path
  - Future systems should further accelerate message passing software
- QoS mechanisms are essential with burst buffers or systems that shared network resources across jobs
  - Multiple networks might still provide best solution in some cases
  - Network partitioning could still be valuable on future systems
- GPU Direct and NVMe reduce within node messaging impact
- High capability nodes lead to a smaller network
  - Reduces importance of network partitioning
  - LLNL CTS-1 with 2-to-1 tapered fat tree still require careful task mapping

Substantial research questions still remain for systems after Sierra

Lawrence Livermore
National Laboratory