Argonne NATIONAL LABORATORY

PERFORMANCE EVALUATION OF MPI ON CRAY XC40 XEON PHI SYSTEM

SCOTT PARKER, SUDHEER CHUNDURI, KEVIN HARMS – ARGONNE NATIONAL LAB

KRISHNA KANDALLA - CRAY

May 22, 2018

MPI PERFORMANCE ANALYSIS AND MODELING FOR THETA

- Quantify system MPI performance:
  - Baseline performance of MPI on Theta
  - Impact of various tunable MPI parameters
  - Track MPI performance over time to
    - monitor system health
    - impact of software updates
- Develop simplified models of MPI performance to:
  - Assist with application performance analysis and tuning
  - Provide input for application development and design
  - Project application performance on future system

THETA

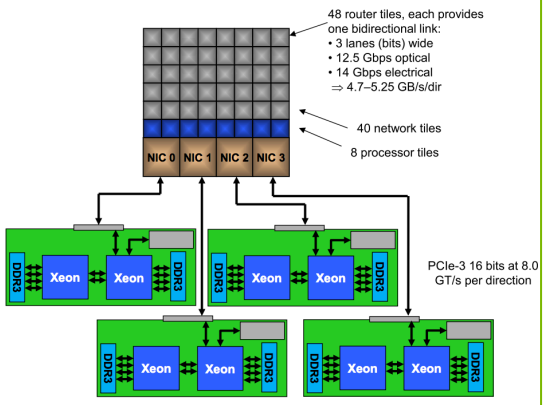
- System:
  - Cray XC40 system
  - 24 racks
  - 4,392 compute nodes/ 281,088 cores
  - 11.7 PetaFlops peak performance
  - Accepted Fall 2016
- Processor:
  - Intel Xeon Phi, 2nd Generation (Knights Landing) 7230
  - 64 Cores
  - 1.3 GHz base / 1.1 GHz AVX / 1.4-1.5 GHz Turbo
- Memory:
  - 16 GB MCDRAM per node
  - 192 GB DDR4-2400 per node
  - 913 TB of total system memory
- Network:
  - Cray Aries interconnect
  - Dragonfly network topology
  - 12 groups
- Filesystems:
  - Project directories: 10 PB Lustre file system
  - Home directories: GPFS

Argonne NATIONAL LABORATORY

ARIES DRAGONFLY NETWORK

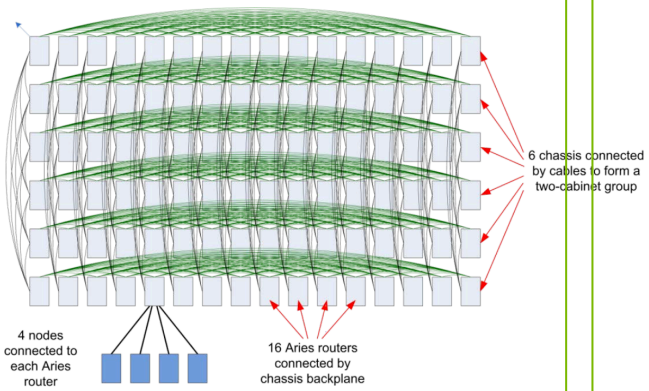
Aries Router:

- 4 Nodes connect to an Aries
- 4 NIC's connected via PCIe
- 40 Network tiles/links
- 4.7-5.25 GB/s/dir per link



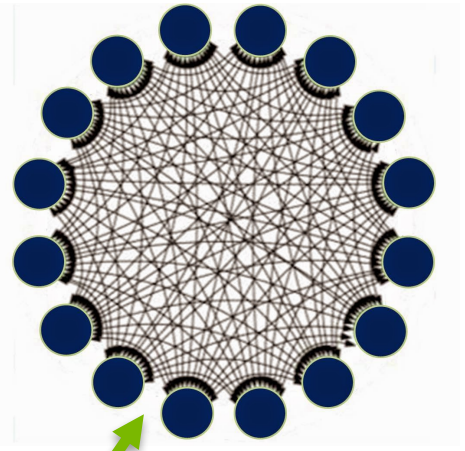
Connections within a group:

- 2 Local all-to-all dimensions
  - 16 all-to-all horizontal
  - 6 all-to-all vertical
- 384 nodes in local group



Connectivity between groups:

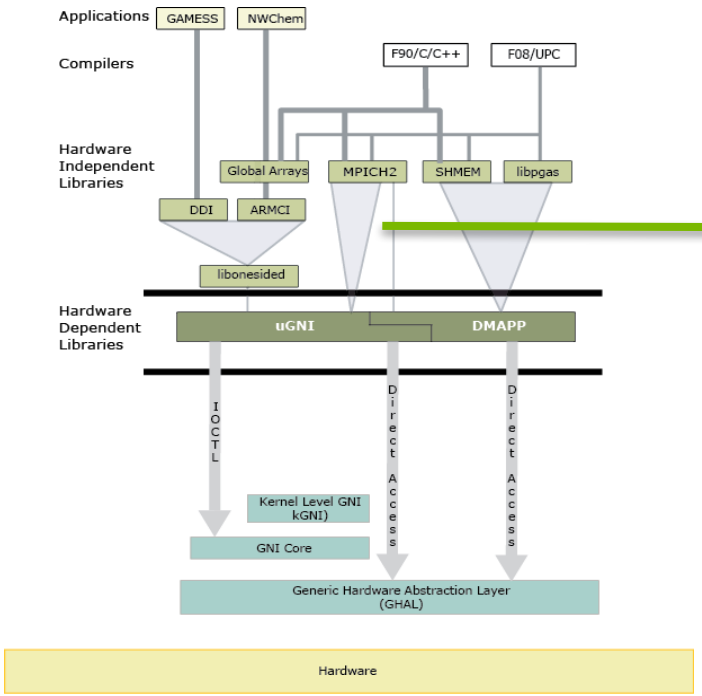
- Each group connected to every other group
- Restricted bandwidth between groups



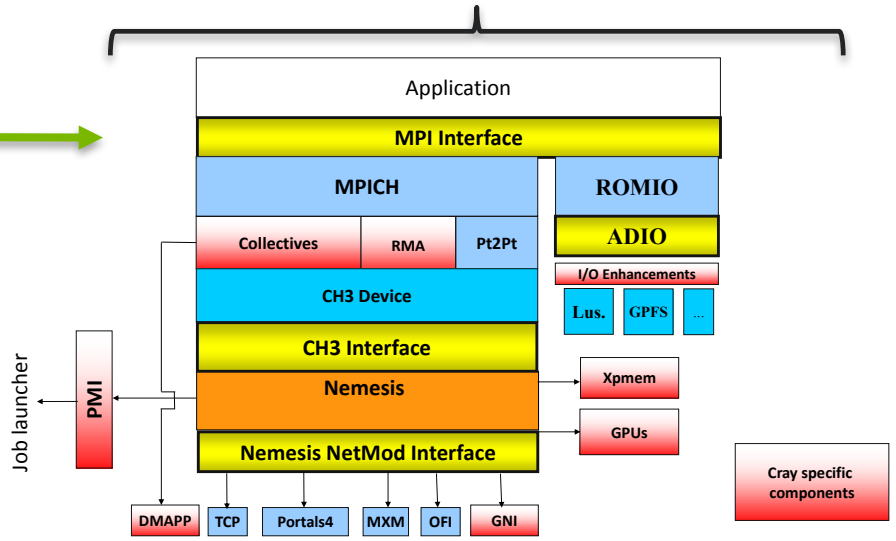
Theta has 12 groups with 12 links between each group

Argonne NATIONAL LABORATORY

CRAY MESSAGING SOFTWARE STACK



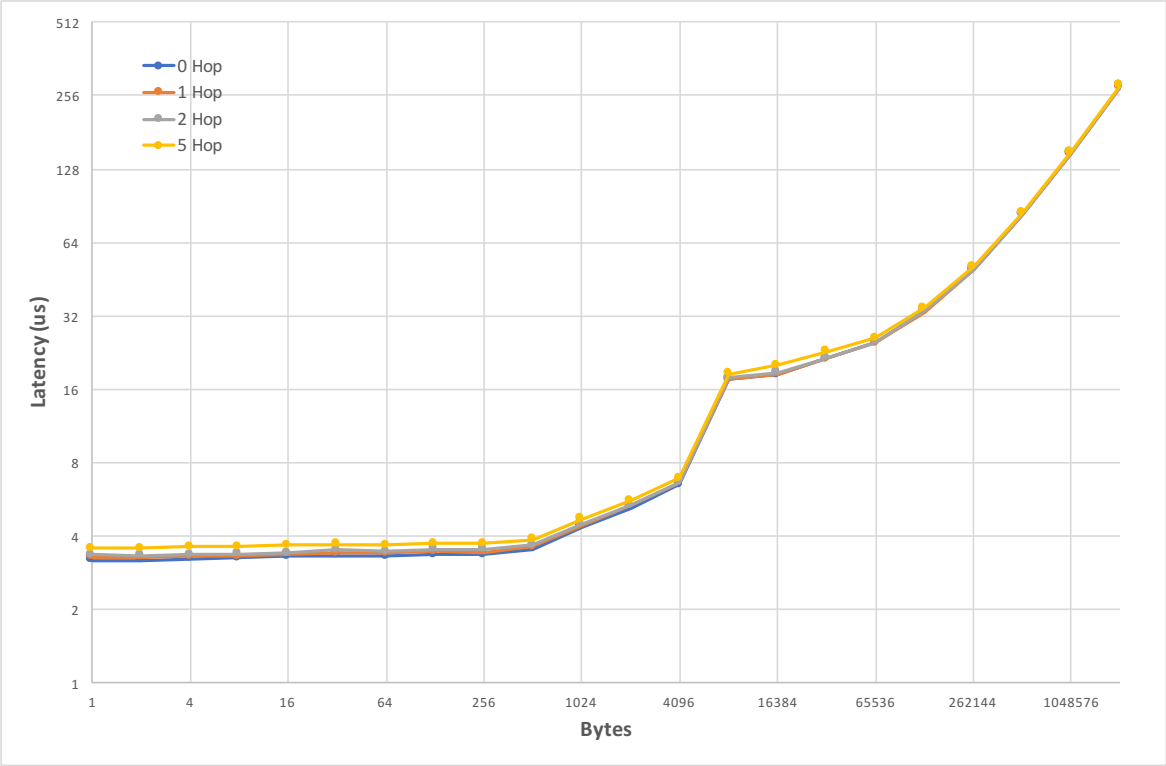
Cray MPI is derived from MPICH



MPI POINT-TO-POINT PERFORMANCE AND MODELS

MPI SEND AND RECEIVE LATENCY

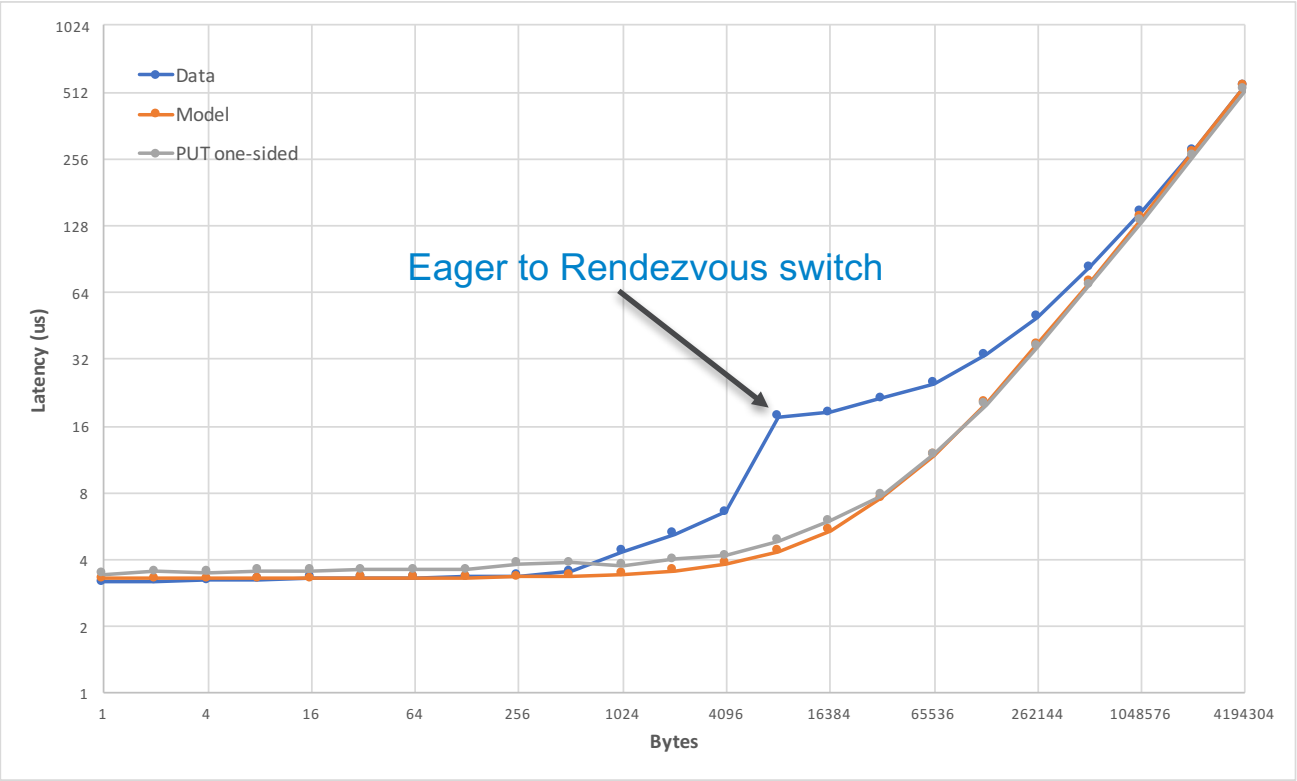
OSU PtoP MPI Latency on Theta



- Latency tested for pairs placed different distances or hops apart
  - 0 – on same Aries
  - 1 – same row/col
  - 2 – same groups
  - 5 – between groups
- Hop count does not strongly influence latency

MPI SEND AND RECEIVE MODEL

OSU PtoP MPI Latency on Theta



Simple (Hockney) model:

$$T = \alpha + \beta \cdot n$$

$$n = bytes$$

$$\propto = 3.3$$

$$\beta = 0.0013$$

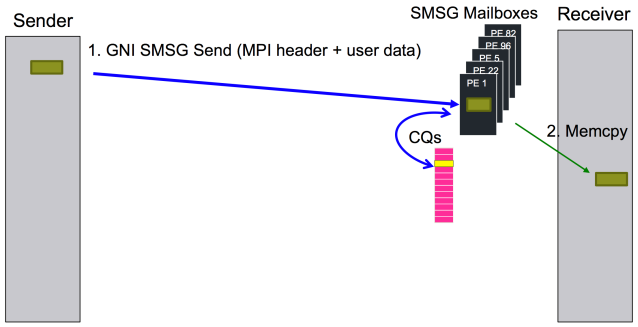
Model fits well for low and high byte counts

Eager to rendezvous protocol switch believed to be producing “bump” in latency

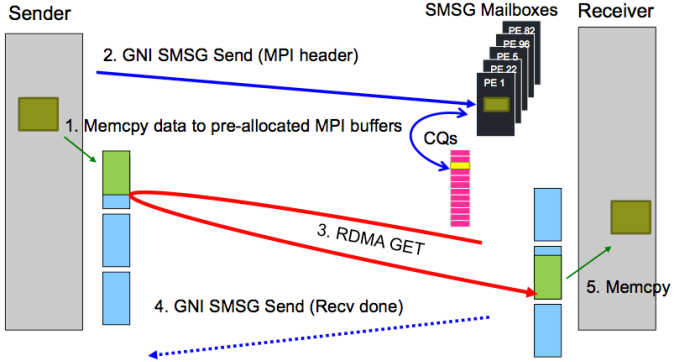
One sided PUT latency results lack “bump” and are close to the model

CRAY MPICH EAGER AND RENDEZVOUS PROTOCOLS

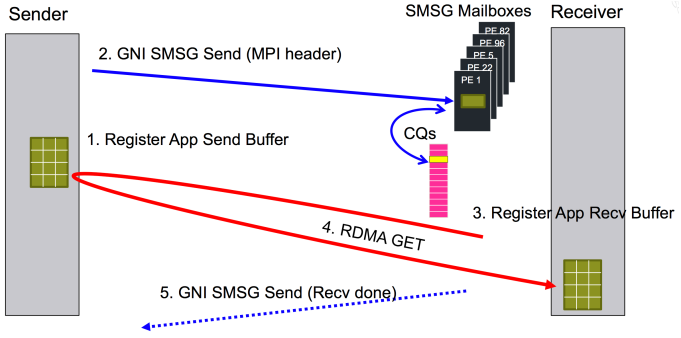
E0 - Eager



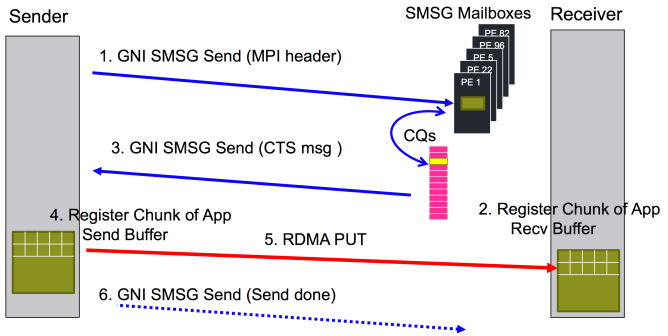
E1 - Eager



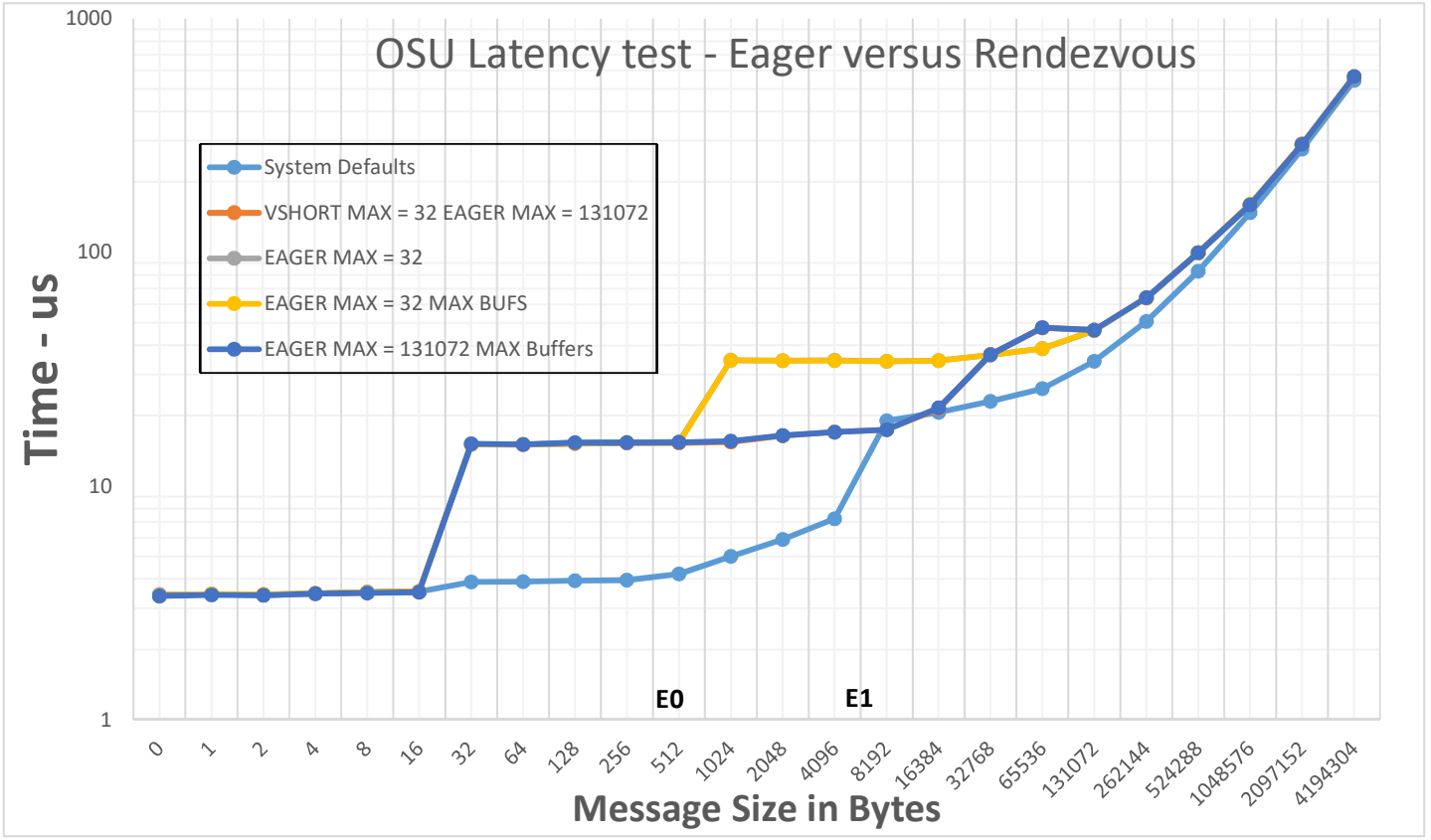
R0 - Rendezvous



R1 - Rendezvous



IMPACT OF EAGER AND RENDEZVOUS SETTINGS



MPI COLLECTIVES PERFORMANCE AND MODELS

MOST FREQUENTLY CALLED COLLECTIVE ROUTINES

Approximate relative call frequency from ALCF applications workload

Collectives Studied

Routine	Relative Call Frequency
Allreduce	5000
Bcast	2500
Barrier	500
Alltoall	500
Alltoallv	250
Reduce	75
Allgatherv	25
Everything else	<1

MPICH COLLECTIVES IMPLEMENTATION

- MPI collective routines are implemented using a series of point-to-point messages
- A variety of different algorithms are used for different collectives and within collectives for different messages sizes and rank counts
- There all well established time estimates for collective algorithms based on point-to-point models
- MPICH MP_Bcast example uses:
  - Binomial tree for small messages or small processor counts
    - $T = (\propto\cdot + \beta \cdot n) log_2(p)$
  - Scatter followed by recursive doubling allgather for messages sizes below set threshold and power of two ranks
    - $T = 2 \cdot \propto\cdot log_2(p) + 2 \cdot n \cdot \beta \cdot \frac{p-1}{p}$
  - Scatter followed by a ring allgather for everything else
    - $T = \propto\cdot (log_2(p) + p - 1) + 2 \cdot n \cdot \beta \cdot \frac{p-1}{p}$

CRAY OPTIMIZED COLLECTIVES

Software Optimizations:

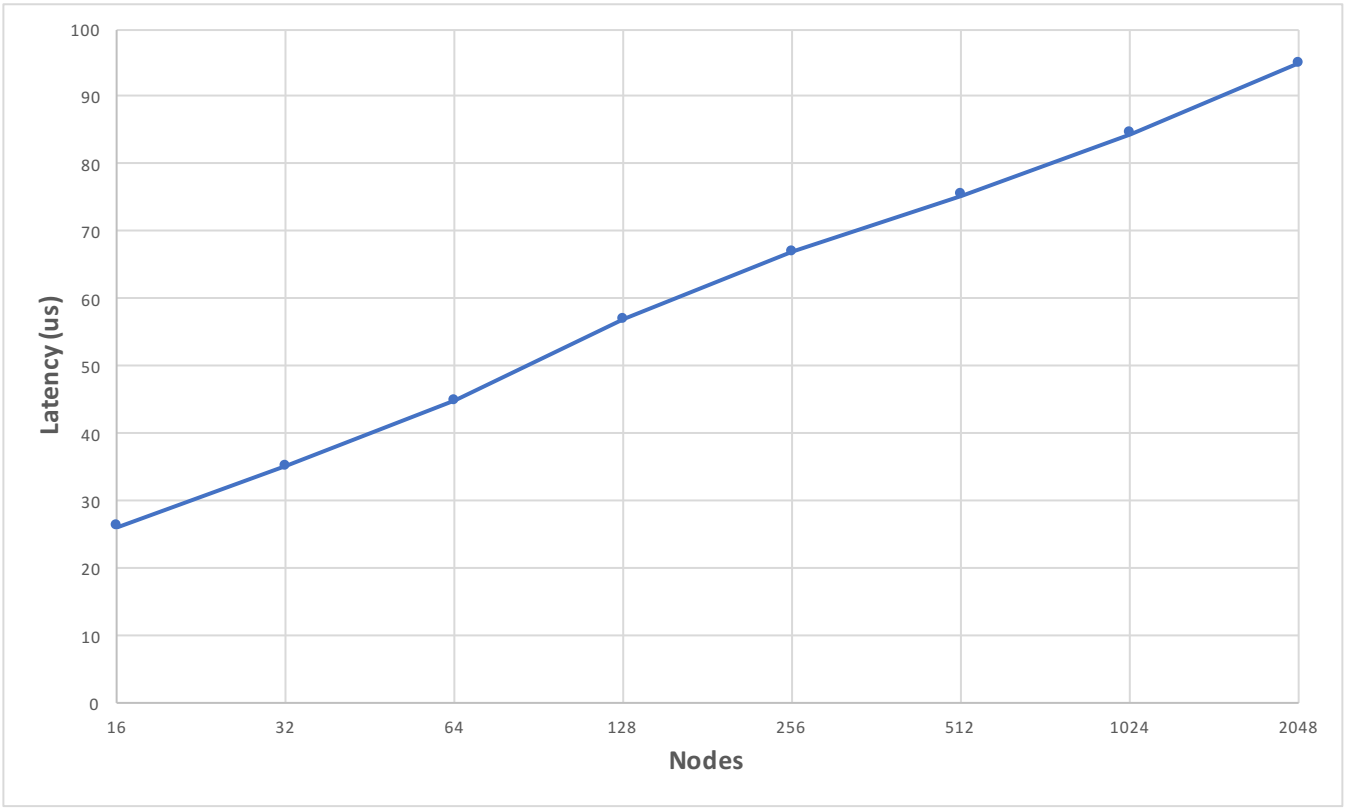
- MPI_Allreduce
- MPI_Bcast
- MPI_Barrier
- MPI_Alltoall, MPI_Alltoallv
- MPI_Allgather, MPI_Allgatherv
- MPI_Gatherv
- MPI_Scatterv
- MPI_Igatherv

Hardware Collective Engine Optimization:

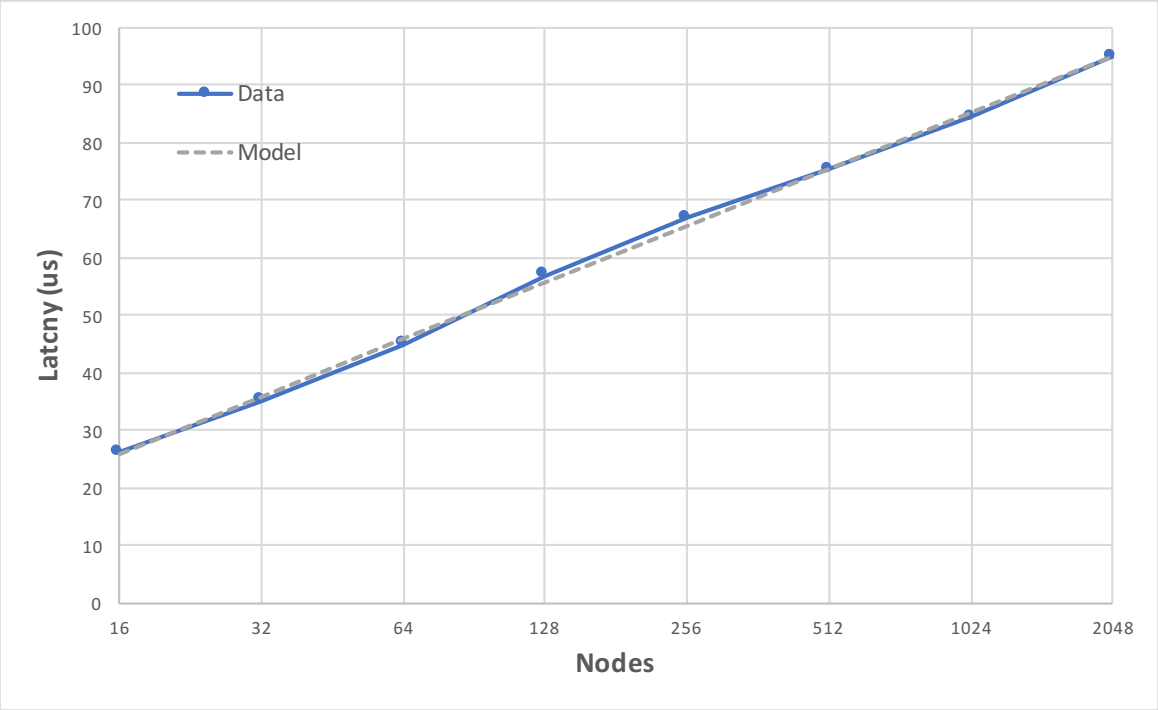
- Applicable for small message MPI_Bcast, MPI_Allreduce, MPI_allreduce, and MPI_Barrier
- Requires using DMAPP to enable the Aries HW Collective Engine
- MPI using just the standard uGNI library does not provide hardware acceleration

MPI BARRIER PERFORMANCE

OSU MPI Barrier Benchmarks



MPI BARRIER MODEL

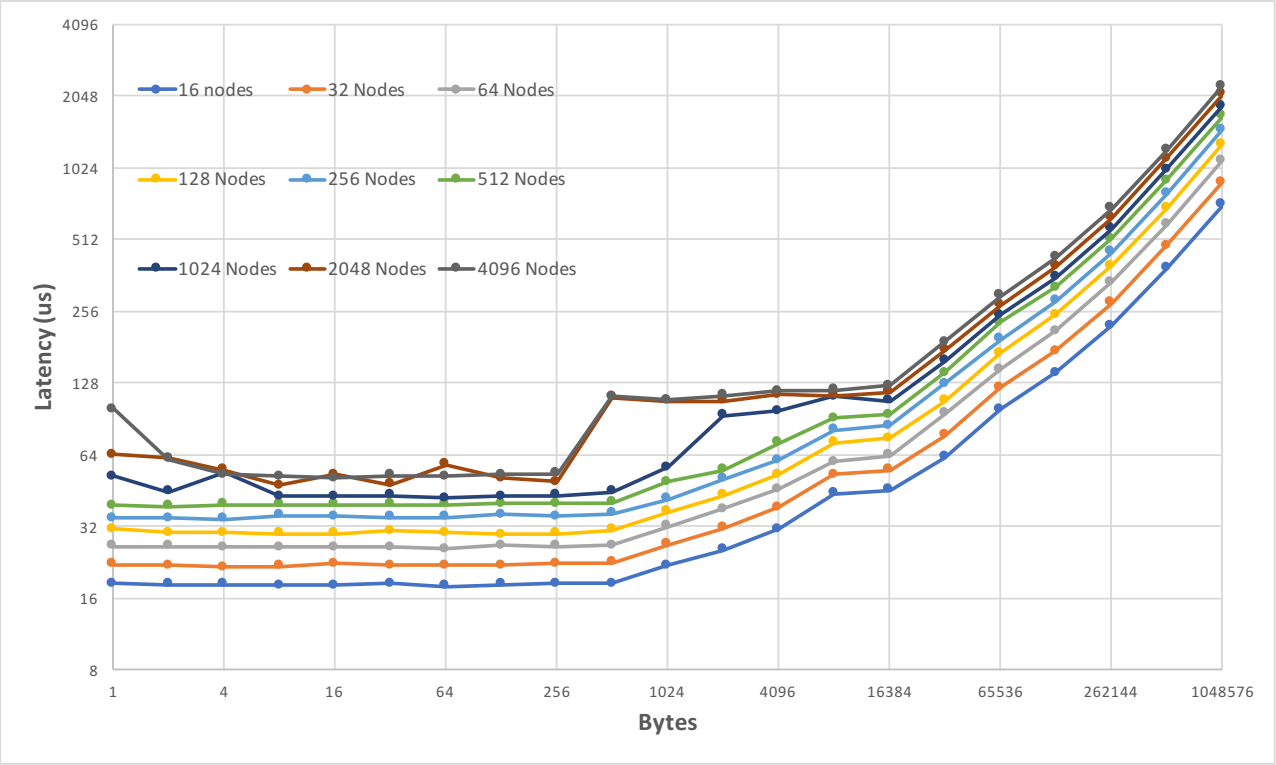


$$T = \alpha + \beta \cdot log_2(p)$$

$$p = nodes$$
$$\propto = -13.5$$
$$\beta = 9.87$$

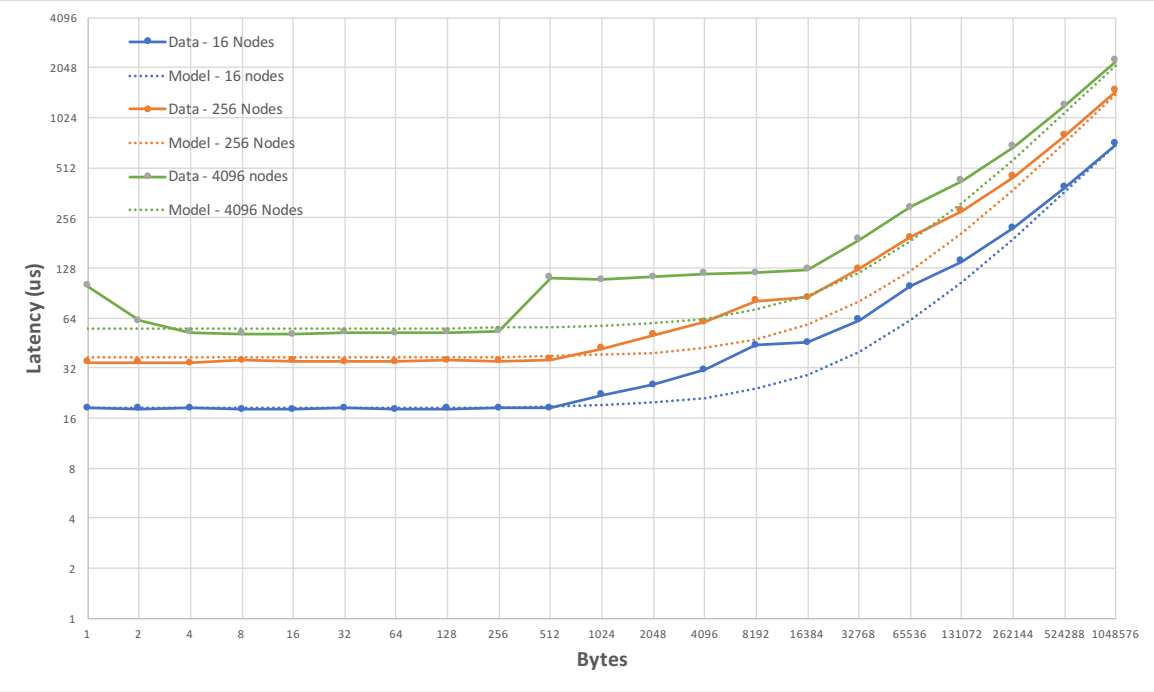
MPI BROADCAST PERFORMANCE

OSU MPI Bcast Benchmarks



Argonne NATIONAL LABORATORY

MPI BROADCAST MODEL



$$T = (\alpha + \beta \cdot n)Log_2(p)$$

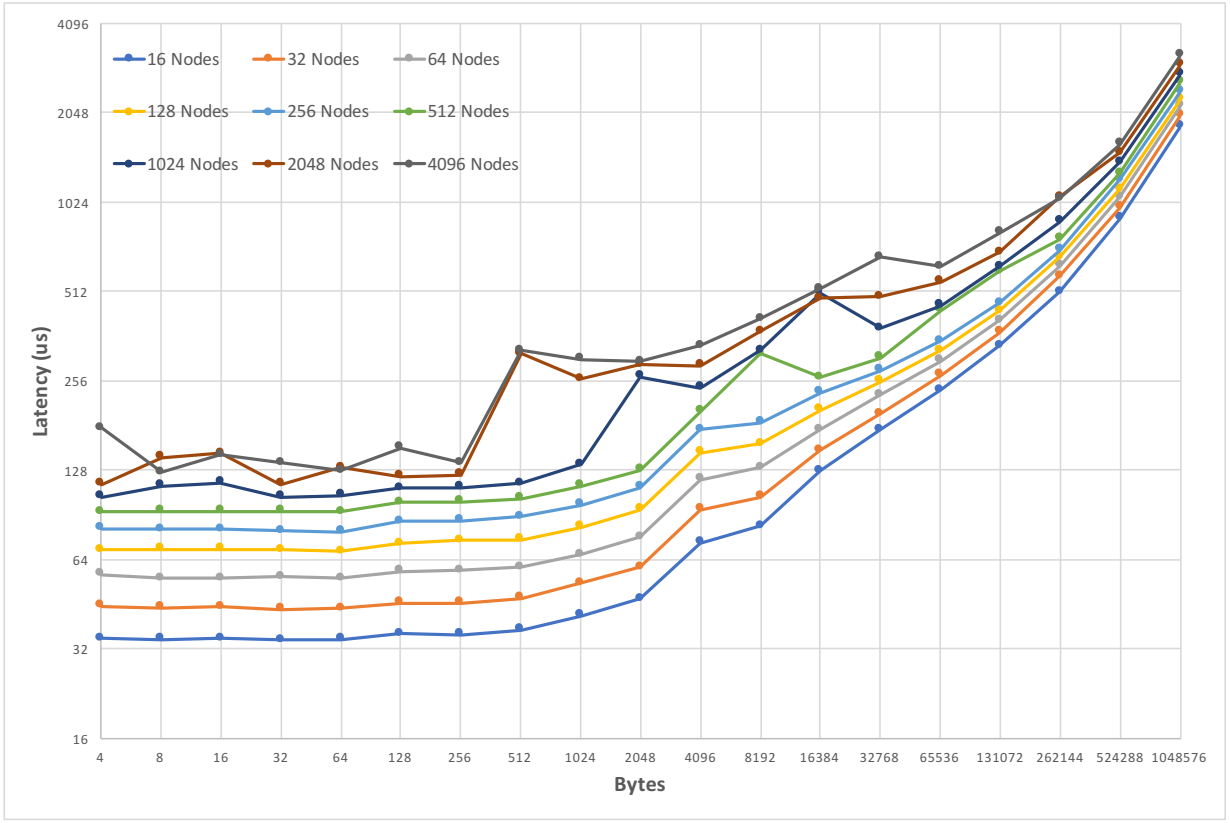
$$n = bytes$$
$$p = nodes$$
$$\propto = 4.6$$
$$\beta = 0.0016$$

Good fit at low and high byte ranges.

Errors centered around point of protocol switch

MPI ALLREDUCE PERFORMANCE

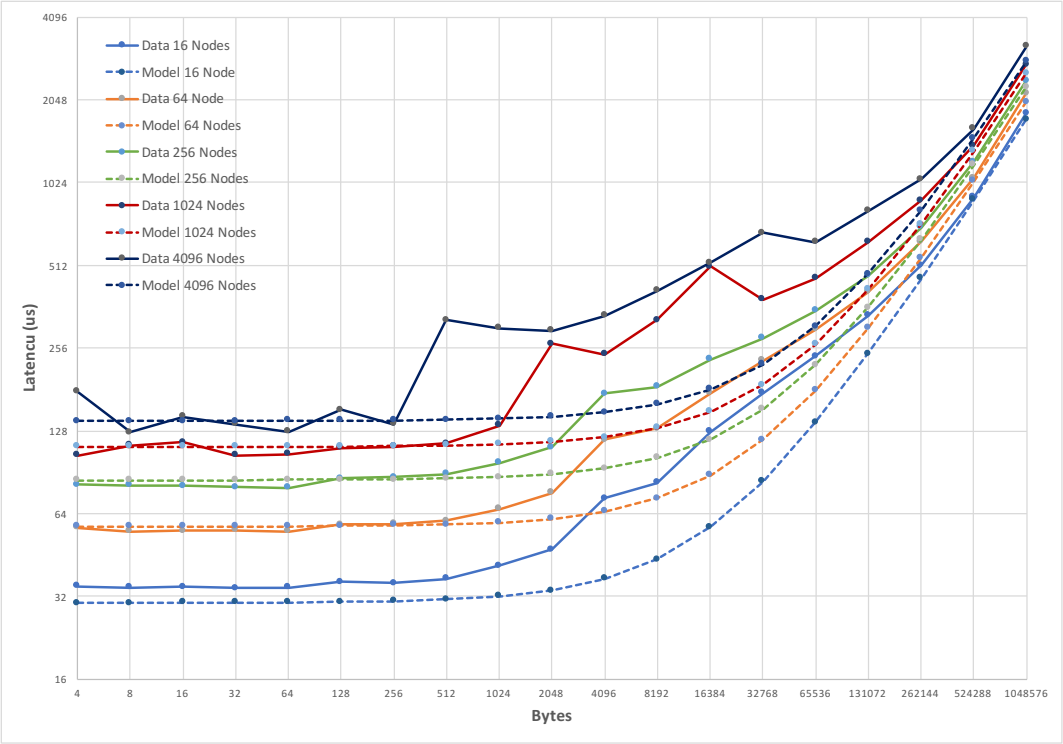
OSU MPI Allreduce Benchmark



Argonne NATIONAL LABORATORY

MPI ALLREDUCE MODEL

OSU MPI Allreduce Benchmark



$$T = \gamma + \delta n + (\alpha + \beta n) log_2(p)$$

$$n = bytes$$

$$p = nodes$$

$$\gamma = -24$$

$$\delta = 0.0012$$

$$\propto = 13.6$$

$$\beta = 0.00012$$

Good fit at low and high byte ranges.

Errors centered around point of protocol switch

CONTENTION, CONSISTENCY, AND VARIABILITY

NETWORK CONTENTION

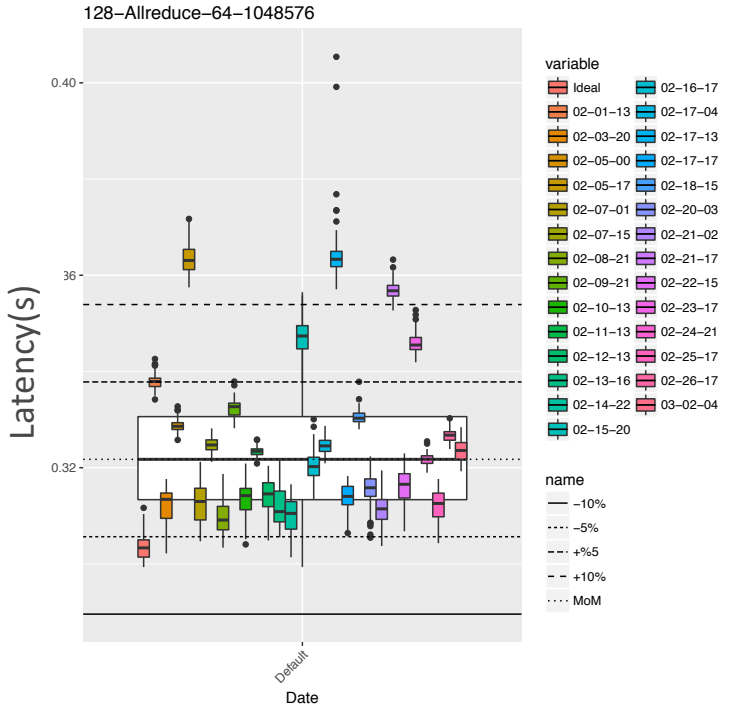
- Jobs on Theta typically consist of nodes distributed randomly across the network
- The Dragonfly topology on the XC40 does not provide traffic isolation between jobs
- Previously shown results were produced with no other network activity present and therefore represent a ”best case” result
- When multiple jobs are run concurrently there may be contention for network resources that reduces the network performance obtained by an individual job
- Theta has 12 optical network links between groups or ~56 GB/s of bi-directional bandwidth between any two groups
- The nodes in a group have a total network injection capacity of ~4 TB/s
- Traffic from multiple jobs between any two groups can lead to congestion due to limited inter-group bandwidth
- Indirect routing can alleviate some congestion impact

VARIABILITY ON THETA

- Identified four causes of variability (Chunduri, et al. "Run-to-run Variability on Xeon Phi Based Cray XC Systems". SC17, 2017)
  - Core level variability due to OS noise
  - Tile level variability due to shared resource contention on tile (L2)
  - Memory mode variability due to cache mode page conflicts
  - Network variability due to shared network resources
- Variability between runs on Theta:
  - frequently 15% or greater
  - can be up to 100%

NETWORK-LEVEL VARIABILITY

- **MPI_Allreduce** with 8 MB message on 128 nodes
- Repeated 100 times within a job
- Measured on several days
  - Changes in node placement and Job mix
- Isolated system run:
  - < **1%** variability (best observed)
- Variability is around **35%**
  - Much higher variability with smaller message sizes (not shown here)
- Each box shows the median, IQR (Inter-Quartile Range) and the outliers



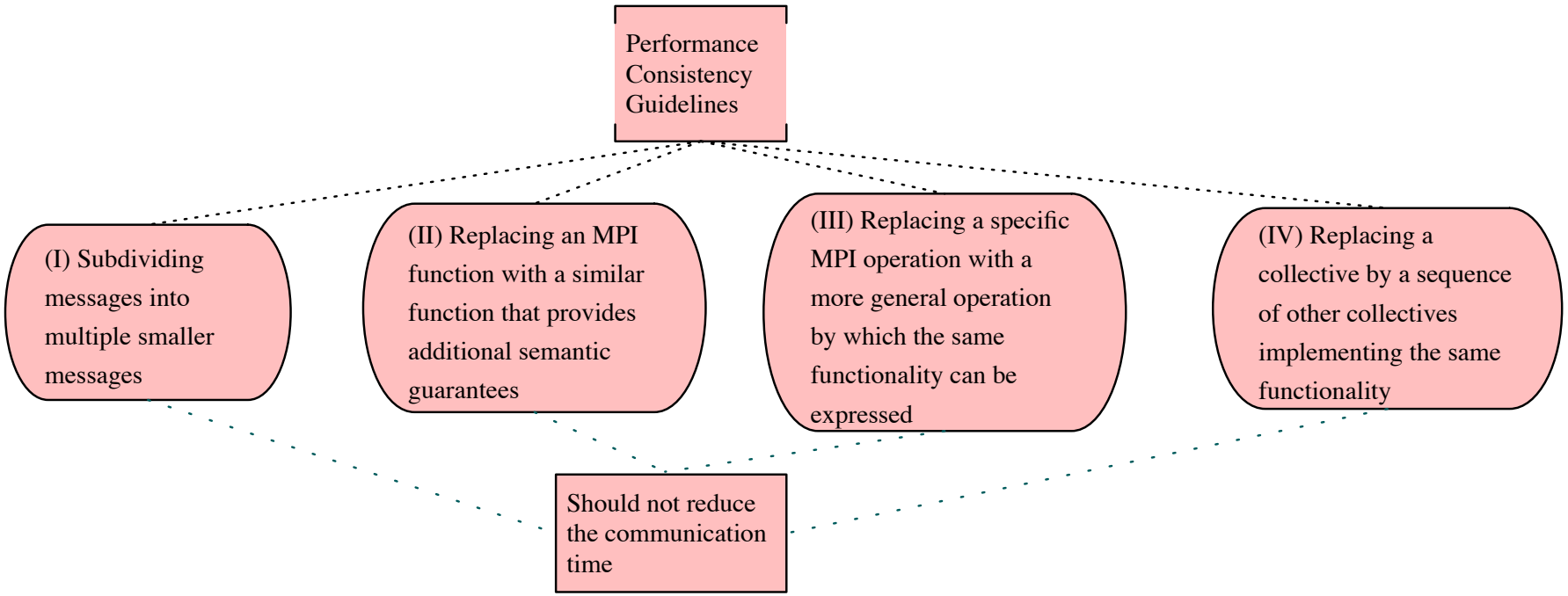
Different jobs

128 nodes Allreduce 8MB 64 PPN

MPI PERFORMANCE CONSISTENCY

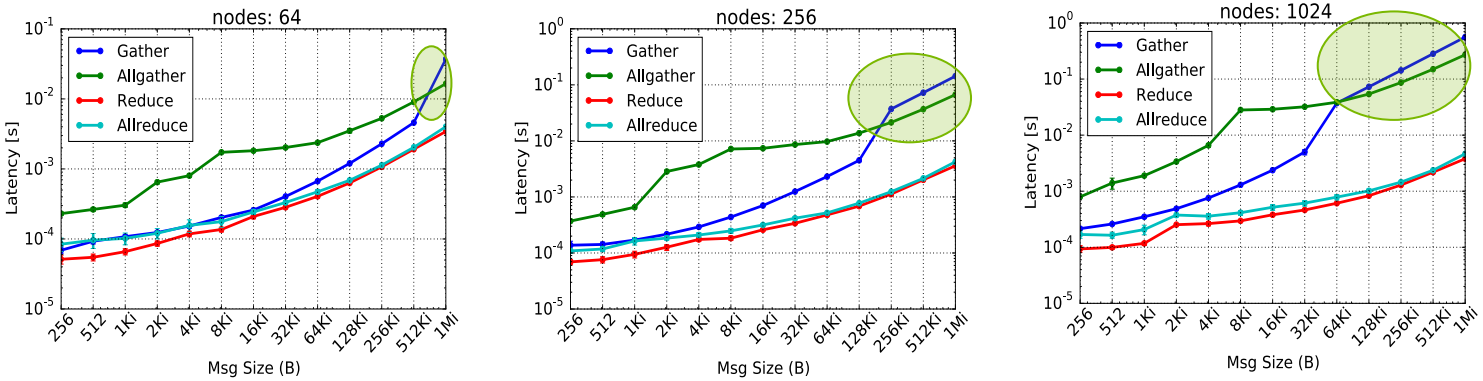
- Performance guidelines for the expected behavior of MPI collectives have been defined
  - Jesper Larsson Träff, William D Gropp, and Rajeev Thakur. 2010. Self-consistent MPI performance guidelines. IEEE Transactions on Parallel and Distributed Systems 21, 5 (2010), 698–709.
- A performance guideline usually defines a common-sense performance expectation based on sematic functionality of the collectives,
  - MPI_Allgather on n data elements should "not be slower" than a combination of a call to MPI_Gather with n data elements followed by a call to MPI_Broadcast with n data elements.

MPI PERFORMANCE CONSISTENCY



MPI PERFORMANCE CONSISTENCY RESULT ON THETA

- Collectives performance is generally found to be consistent when consistency tests are run with no other jobs on the machine
- Persistent inconsistencies arise when multiple jobs are running concurrently
- Example performance consistency violations found for Allgather:



On 256 Nodes	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288	1048576
MPI_Allgather ≤ Alltoall												✗	✗	✗	✗	✗	✗	✗		✗	
MPI_Allgather ≤ Allreduce						✗						✗									
MPI_Allgather ≤ Gather + Bcast	✗	✗	✗			✗	✗	✗				✗	✗	✗	✗						

Argonne NATIONAL LABORATORY

SUMMARY

- Simple model for point-to-point communication provides good accuracy except where protocol shifts occur in the underlying implementation (1-256k bytes)
- Some collectives (Allreduce, Bcast, Barrier, Alltoall) used significantly more frequently than other on ALCF systems
- Collective models capture overall collective performance trends well but errors are present in region where point-to-point protocols change
- Collective performance can vary be more than 35% due to congestion
- Some collectives performance consistency violations observed in the presence of congestion

QUESTIONS?