Designing Topology-Aware Collective Communication Algorithms for Large Scale InfiniBand Clusters: Case Studies with Scatter and Gather

Krishna Kandalla, Hari Subramoni, Abhinav Vishnu and Dhabaleswar K. (DK) Panda

IPDPS 2010

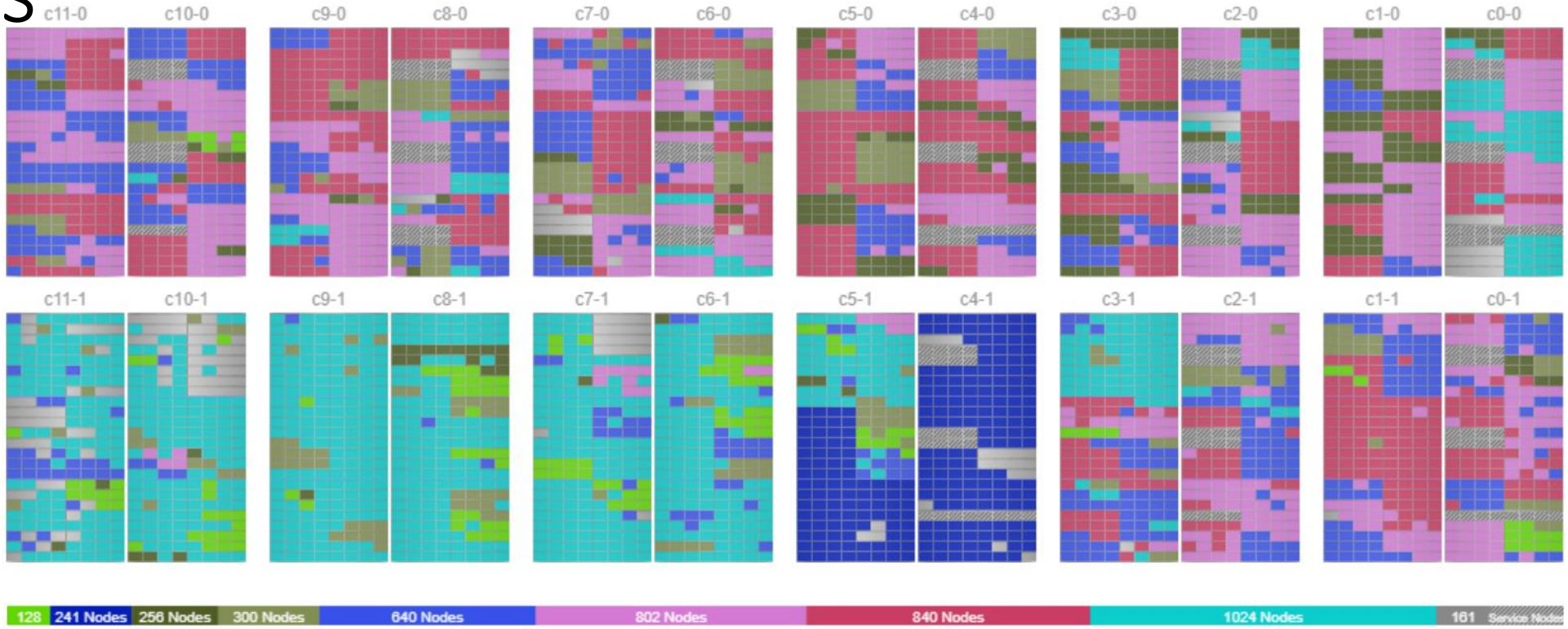
March 6, 2024

Source: hpe.com


IBM
Blue Gene/P


भारत सरकार की एक पहल
राष्ट्रीय सुपरकंप्यूटिंग मिशन
NSM
NATIONAL SUPERCOMPUTING MISSION
An Initiative of Government of India
परम
संगणक
PARAM
Sanganak

Theta Status



Running | Starting | Queued | Reservations

Total Running Jobs: 8

Job Id		Project	Nodes	Start Time	Run Time	Walltime	Queue	Mode
512304		EstopSim_2	1024	9:13:30 AM	00:12:13	16:00:00	default	script
512623		TurbShockWalls	840	7:57:42 AM	01:28:01	1d 00:00:00	default	script
498557		TurbShockWalls	802	9:43:57 PM	11:41:46	1d 00:00:00	default	script
513358		PSFMat_2	640	8:29:27 AM	00:56:16	12:00:00	default	script
511830		ReconDepth	300	7:50:53 AM	01:34:50	06:00:00	default	script
514000		HighLumin	256	8:50:22 AM	00:35:21	06:00:00	default	script
514114		CVD_CityCOVID	241	1:28:47 AM	07:56:56	1d 12:00:00	CVD_Research	script
514178		FDTD_Cancer_2a	128	8:00:51 AM	01:24:52	03:00:00	default	script

status.alcf.anl.gov -> Theta (now retired)

2023

gpu thetagpu14-gpu3

Total Running Jobs: 8

Job Id		Project	Nodes ▼	Start Time	Run Time	Walltime	Queue	Mode
651326		TRB	1800	9:46:25 PM	15:23:19	1d 00:00:00	default	script
651396		WallBoundedMHDTurb	1024	9:46:11 PM	15:23:34	1d 00:00:00	default	script
651447		3DWholeGenome	930	9:46:29 PM	15:23:16	1d 00:00:00	default	script
651582		PTLearnPhoto	256	7:43:26 AM	05:26:18	06:00:00	default	script
651541		QMCPACK_aesp	256	11:31:38 AM	01:38:06	06:00:00	backfill	script
651605		Catalyst	4	12:29:55 PM	00:39:49	01:00:00	debug-cache-quad	script
10133030		datascience	1	7:48:03 AM	05:21:41	12:00:00	full-node	script
10133042		AI-based-NDI-Spirit	1	11:14:56 AM	01:54:49	12:00:00	bigmem	script

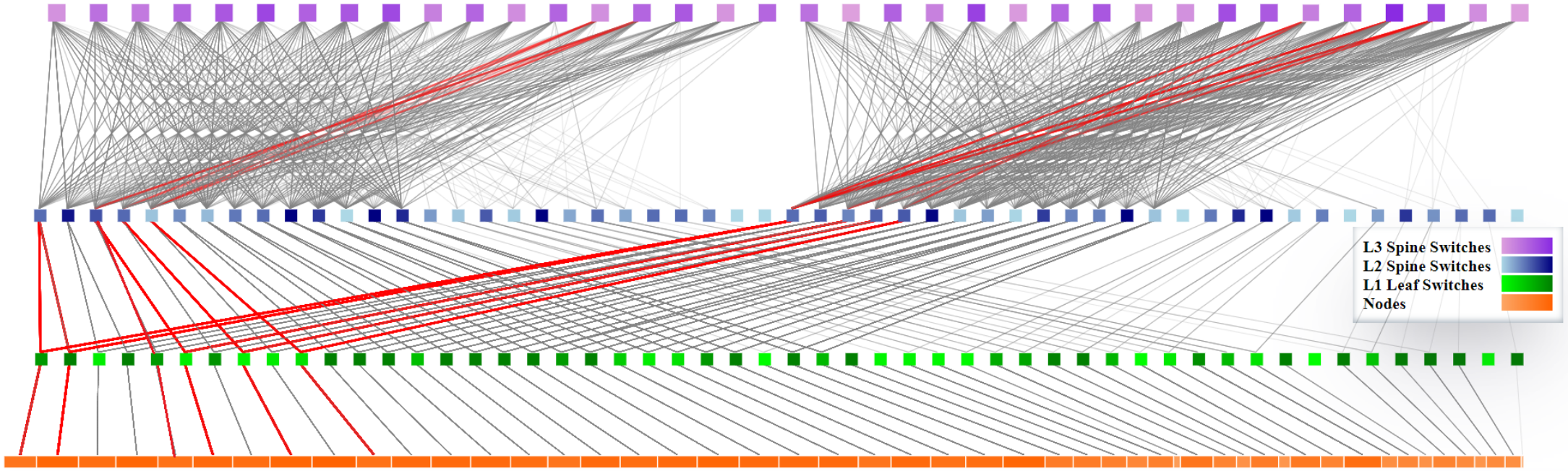
Total Queued Jobs:

Job Id	Project	Score	Walltime	Queued Time	Queue	Nodes	Mode
488198	NucStructReact_6	40.00	06:00:00	114d 02:05:56	backfill	2048	script
488200	NucStructReact_6	40.00	06:00:00	114d 02:05:33	backfill	2048	script
488201	NucStructReact_6	40.00	06:00:00	114d 02:05:15	backfill	2048	script
488202	NucStructReact_6	40.00	06:00:00	114d 02:04:52	backfill	2048	script
488612	NextGenReac	40.00	06:00:00	110d 10:35:36	backfill	1024	script
505064	TurbShockWalls	86.10	09:00:00	33d 20:13:16	default	512	script
509215	ClimateEnergy_4	59,231.06	01:00:00	20d 05:04:52	default	128	script
511951	TurbShockWalls	1,074.68	1d 00:00:00	8d 02:01:15	default	802	script
512878	CSC249ADSE16	397.24	09:30:00	5d 00:47:21	default	1024	script
513149	NanoReactive_3	1,120.58	14:00:00	4d 06:39:18	default	1041	script
513267	TurbShockWalls	620.78	1d 00:00:00	3d 17:12:12	default	840	script
513422	CSC249ADCD08	15.44	06:00:00	3d 03:43:45	backfill	256	script
513426	CSC249ADCD08	15.39	06:00:00	3d 03:25:28	backfill	256	script
513434	CSC249ADCD08	15.18	06:00:00	3d 02:05:25	backfill	256	script
513435	CSC249ADCD08	15.18	06:00:00	3d 02:05:10	backfill	256	script
513436	CSC249ADCD08	15.18	06:00:00	3d 02:04:42	backfill	256	script
513437	CSC249ADCD08	15.18	06:00:00	3d 02:04:03	backfill	256	script
513593	UltrafastMat	1,341.82	1d 00:00:00	2d 02:27:31	default	4096	interactive
513598	DirectFusion	119.55	09:00:00	2d 00:23:41	default	520	script
513613	spentFuel	51.00	01:00:00	1d 23:48:13	default	256	script
513654	TurbShockWalls	106.47	1d 00:00:00	1d 16:48:06	default	832	script
513656	TurbShockWalls	105.66	1d 00:00:00	1d 16:36:07	default	832	script
513722	HHPMT_5	51.07	06:00:00	1d 13:40:04	default	256	script
513736	HierChemSep	64.65	09:00:00	1d 13:13:09	default	512	script
513758	PSFMat_2	146.57	06:00:00	1d 12:13:35	default	512	script
513759	PSFMat_2	146.44	06:00:00	1d 12:12:36	default	512	script
513760	PSFMat_2	65.85	12:00:00	1d 12:09:34	default	640	script
513768	IonTransES	124.08	06:00:00	1d 11:42:09	default	409	script
513769	IonTransES	79.53	09:00:00	1d 11:40:36	default	540	script
513771	IonTransES	87.89	09:00:00	1d 11:20:44	default	718	script
513823	HierChemSep	122.00	03:00:00	1d 08:48:33	default	512	script

Batch Queueing Systems – Features

- Has full knowledge of queued, running jobs
- Has full knowledge of the resource usage
- Typically FCFS with backfilling
- Node allocation may not be contiguous

PARAM Sanganak Real Time Job Path



	JobID	UserName	QueueName	TotalNodes	TotalCores	RequiredTime	JobState	ElaspedTime
⊕	1004182.un05	[redacted]	medium	4	80	96:00	R	44:53
⊕	1004941.un05	[redacted]	medium	4	80	96:00	R	76:53
⊖	1004949.un05	[redacted]	large	6	120	96:00	R	49:32
	NodeList: hpc013,hpc018,hpc027,hpc104,hpc139,hpc169							
⊕	1005316.un05	[redacted]	small	2	40	120:0	R	48:22

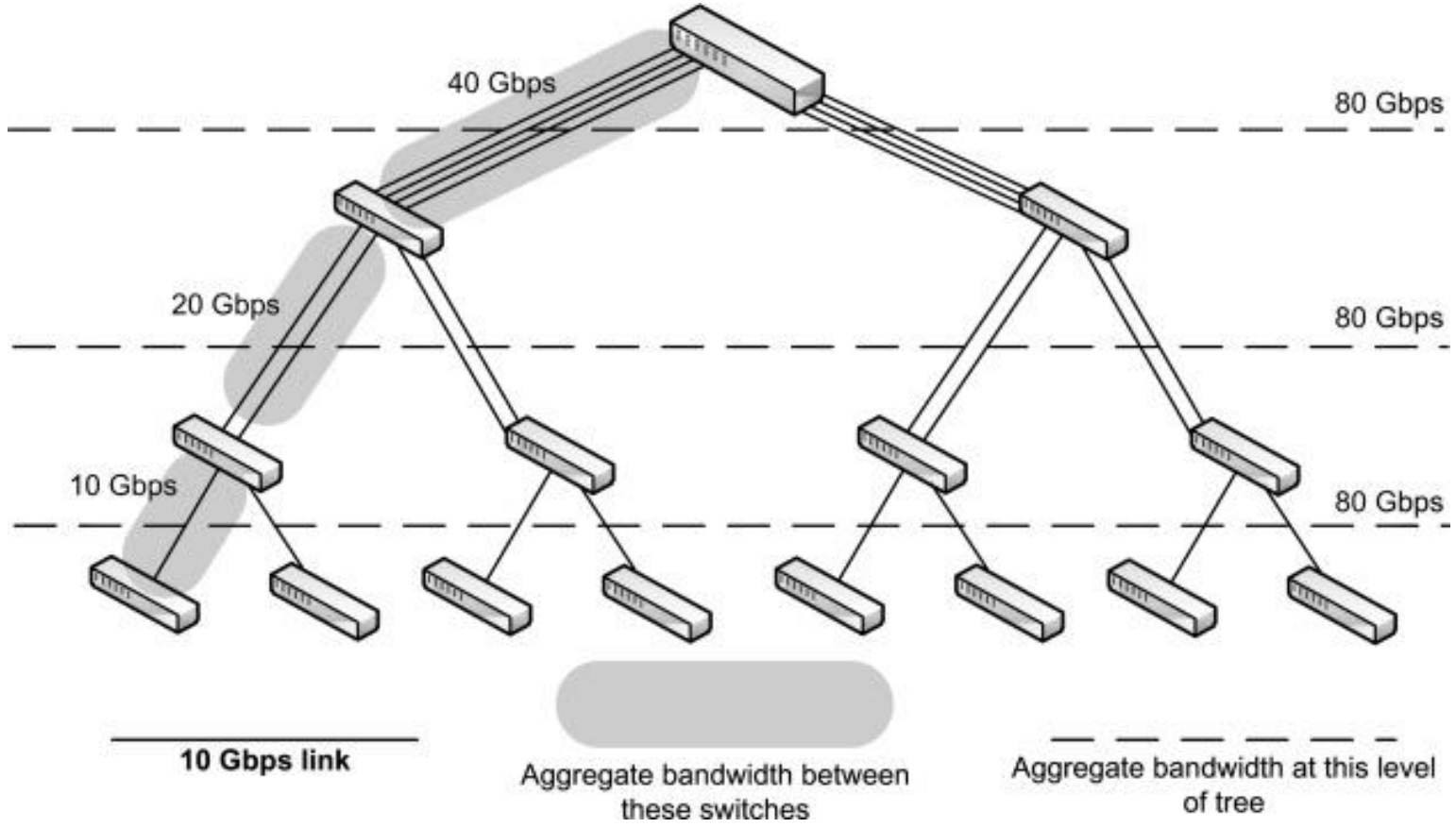
Showing 1 to 150 of 150 entries

Visual Analysis of Congestion and Interference in Supercomputers, A Sharma, D Sahu, P Malakar, HiPC SRS'23

Research Questions

- How do we efficiently discover the topology of a large scale InfiniBand cluster?
- What are the challenges involved in designing efficient collective algorithms that are aware of the network topology?
- Can we derive communication cost models for collective operations on large-scale systems with several levels of hierarchies?
- What is the effect of the background traffic on the performance of collective operations? Can we leverage the topology information to design algorithms that are resilient to network contention?

Network Topology



Fat-tree topology

[www.sciencedirect.com]

A typical topology of large-scale systems (TACC Ranger system)

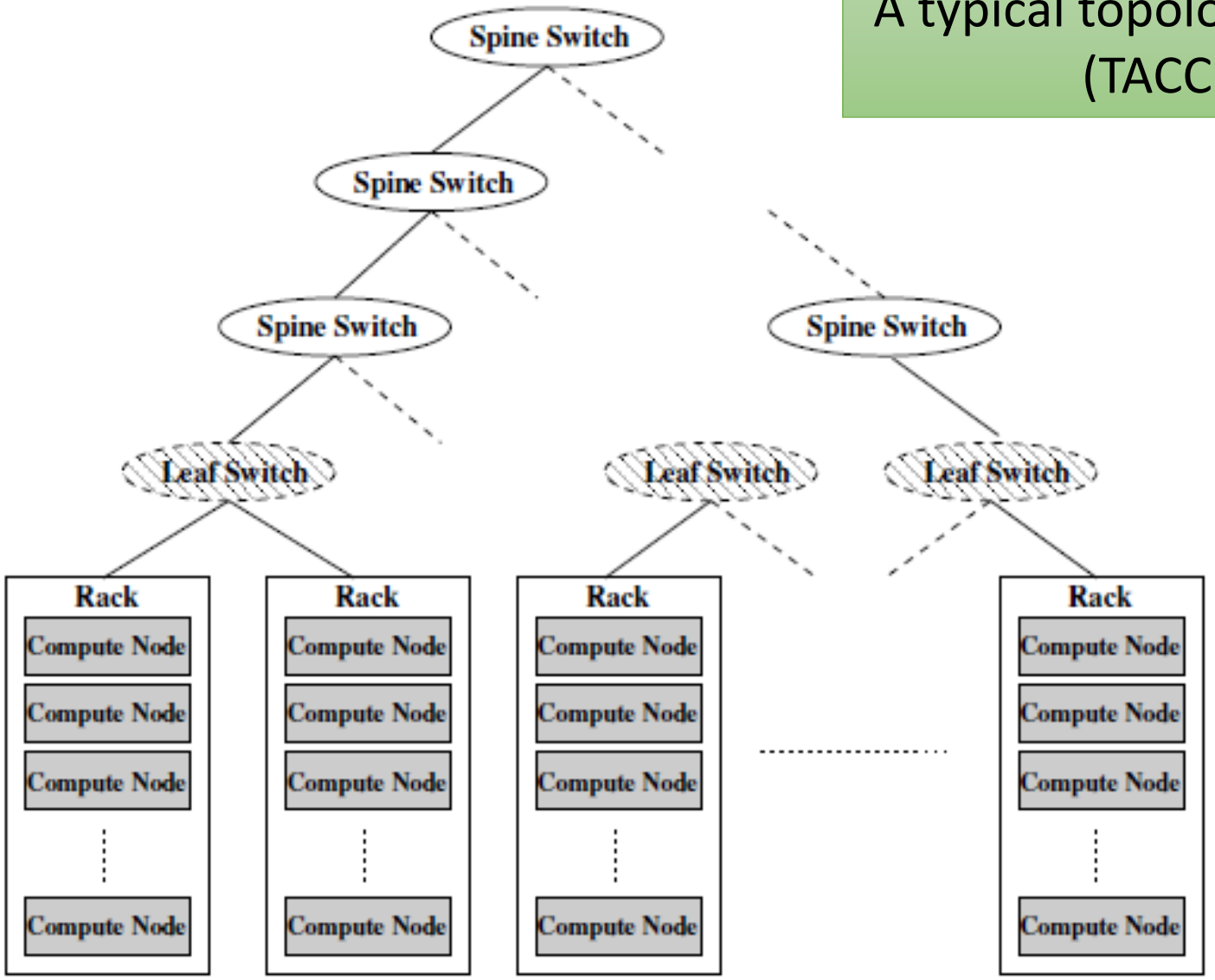
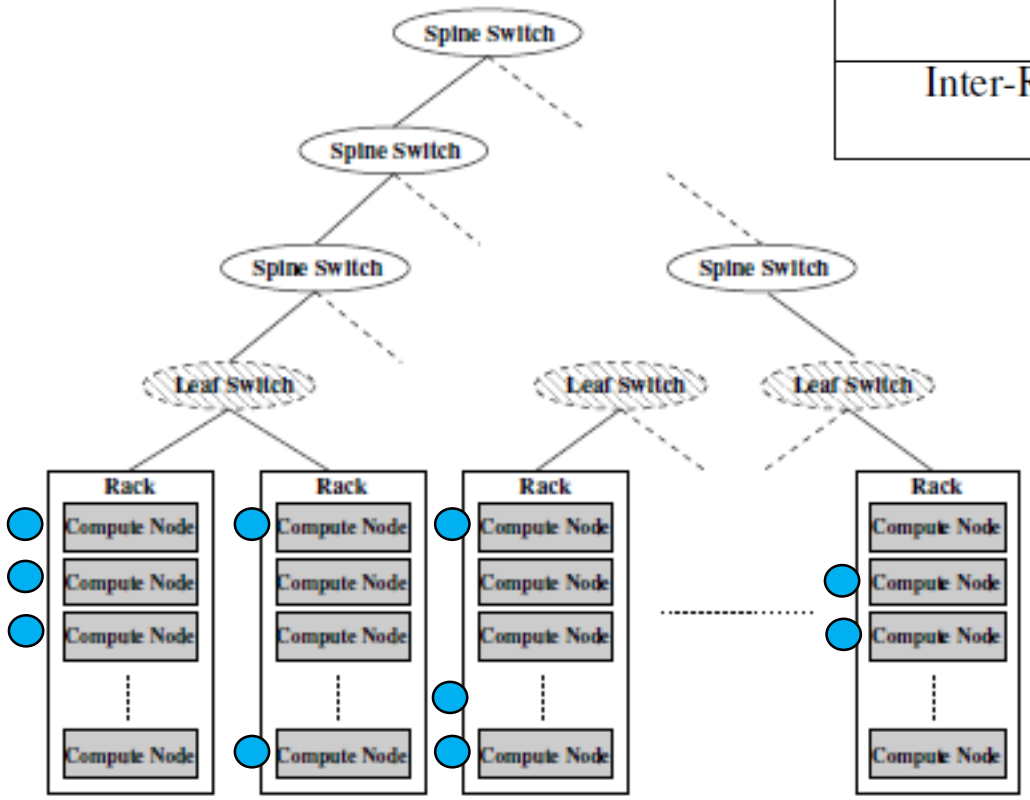


Figure 1. A Typical Topology

Effect of Topology on Latency

Process Location		Number of Hops	MPI Latency (*us*)
Intra-Rack	Intra-Chassis	0 Hops in Leaf Switch	1.57
	Inter-Chassis	1 Hop in Leaf Switch	2.04
Inter-Rack		3 Hops Across Spine Switch	2.45
		5 Hops Across Spine Switch	2.85



A typical topology of large-scale systems

(TACC Ranger system)

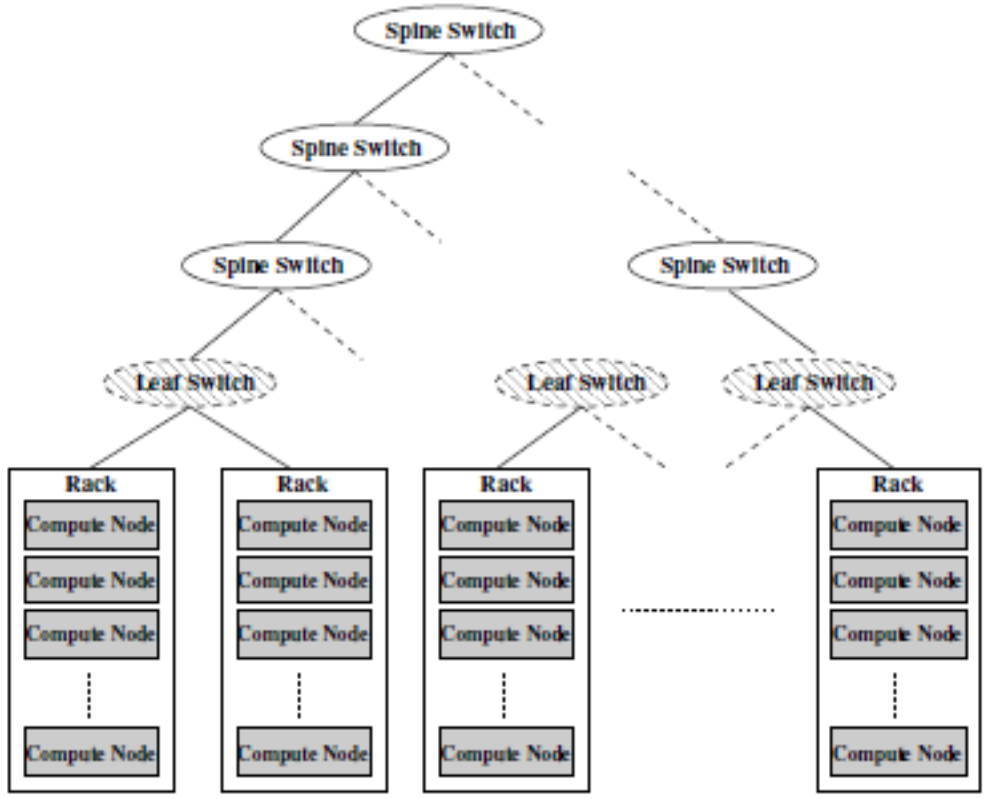
- Allocated nodes are usually scattered in the system
- Different job request sizes and durations
- Contiguous node allocation may increase queue waiting times

Contribution: Topology-aware Collectives

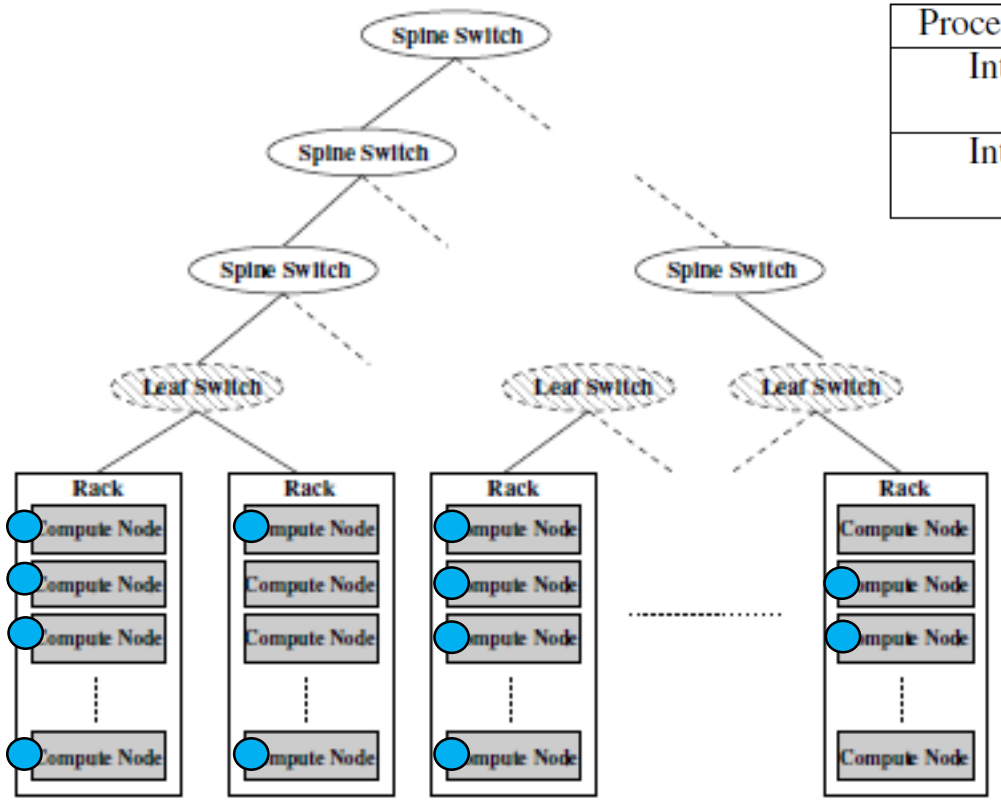
- Topology-aware Gather and Scatter
- Modified communication model
- 54% improvement on micro-benchmarks

Topology-aware Gather

- Designate a rack leader process
- Rack-leader processes independently perform intra-switch gather
- R rack leaders perform inter-switch gather
- Reduced L and B terms (due to reduction in inter-switch exchanges)



Create Sub-communicators



Process Location		Number of Hops	MPI Latency (*us*)
Intra-Rack	Intra-Chassis	0 Hops in Leaf Switch	1.57
	Inter-Chassis	1 Hop in Leaf Switch	2.04
Inter-Rack		3 Hops Across Spine Switch	2.45
		5 Hops Across Spine Switch	2.85

- intra-chassis communicators
- intra-switch communicators
- chassis-leader communicators
- switch-leader communicators

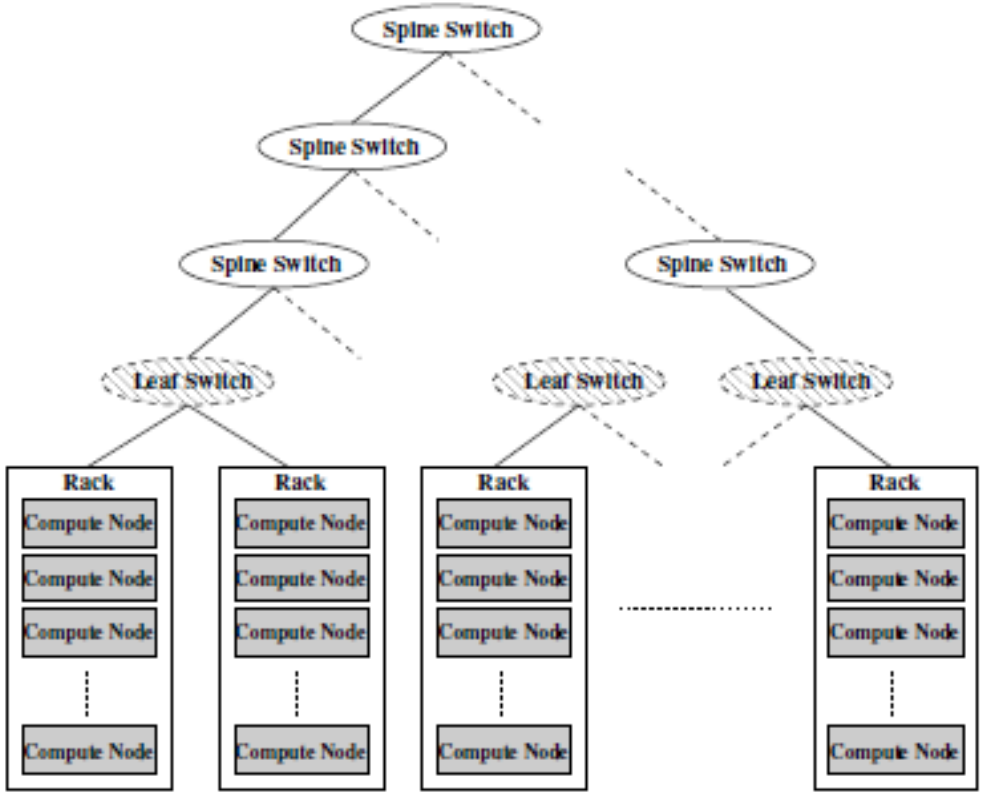
Discover Topology

- Infiniband tools
  - ibnetdiscover – outputs the switch connections / identifiers
  - One-time discovery (in general)

- MPI_Init
  - Create intra-chassis communicators – all nodes in the same chassis
  - Create intra-switch communicators – all nodes in the same leaf switch
  - Assign one chassis-leader and one switch-leader
  - Create switch-leader and chassis-leader communicators

Cost of Communication

t_s-intra-node < t_s-intra-switch < t_s-inter-switch
t_w-intra-node < t_w-intra-switch < t_w-inter-switch



Cost involved for communication within the same node
L: t_s-intra-node
B: t_w-intra-node

Cost of communication within the same leaf switch
L: t_s-intra-switch
B: t_w-intra-switch

Cost involved for an inter-switch communication
L: t_s-inter-switch
B: t_w-inter-switch

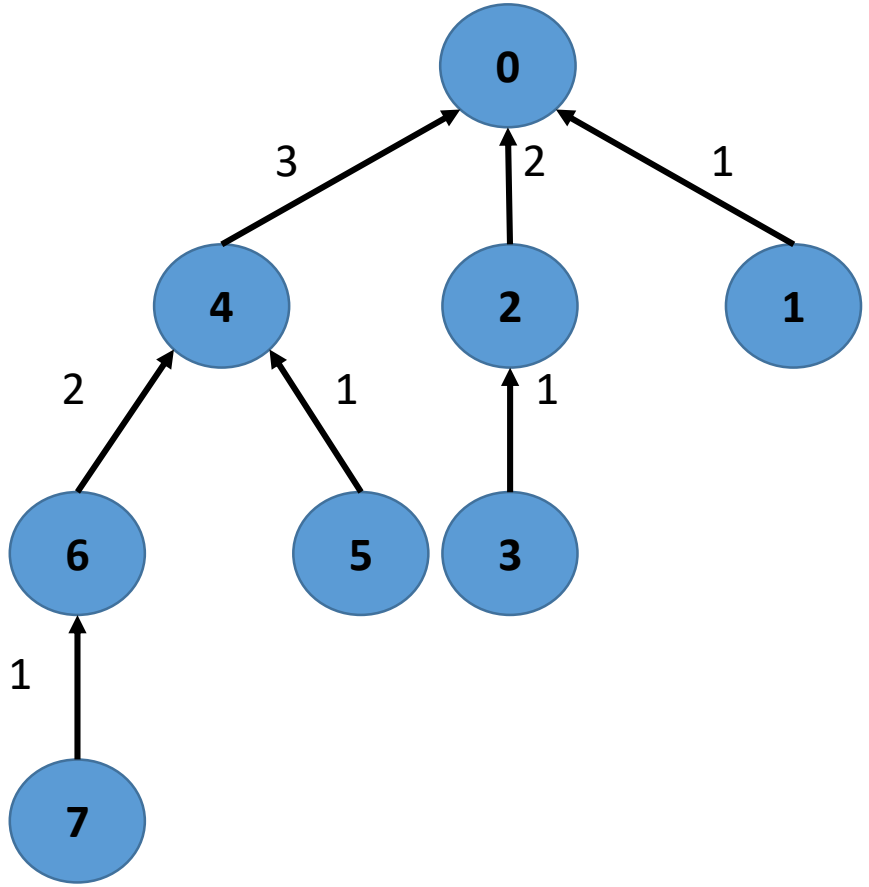
1. Actual cost depends on the #hops based on the actual placement of processes
2. Contention for intra-node/switch << inter-switch

Gather Communication Cost

Binomial tree

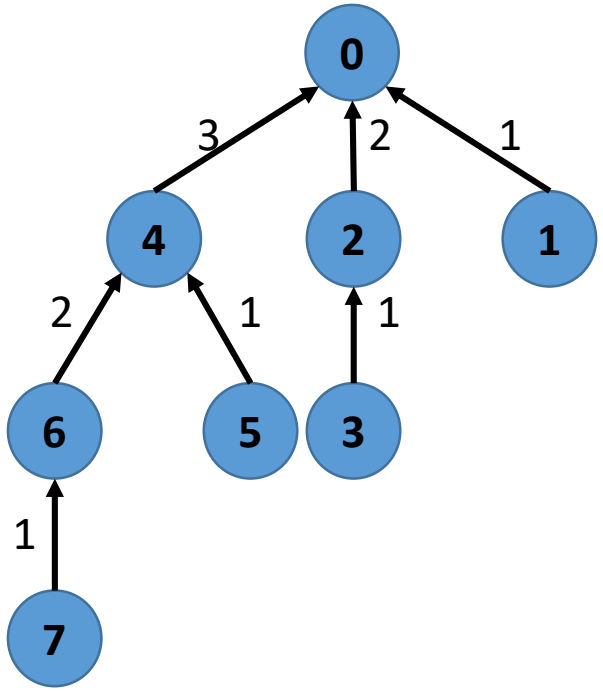
Time: (log p)*L + (p-1)/p*(n/B)

L = latency, B = bandwidth



Cost Model (Original Gather - Binomial)

Number of racks = R
Number of processes = P
Message size = N



Number of exchanges at i^{th} level: C_i
C_1 = Number of intra-node transfers
C_2 = Number of intra-switch transfers
C_3 = Number of inter-switch transfers

Switch-level contention: α

There's a typo in the paper. Read this as ts-intra-node

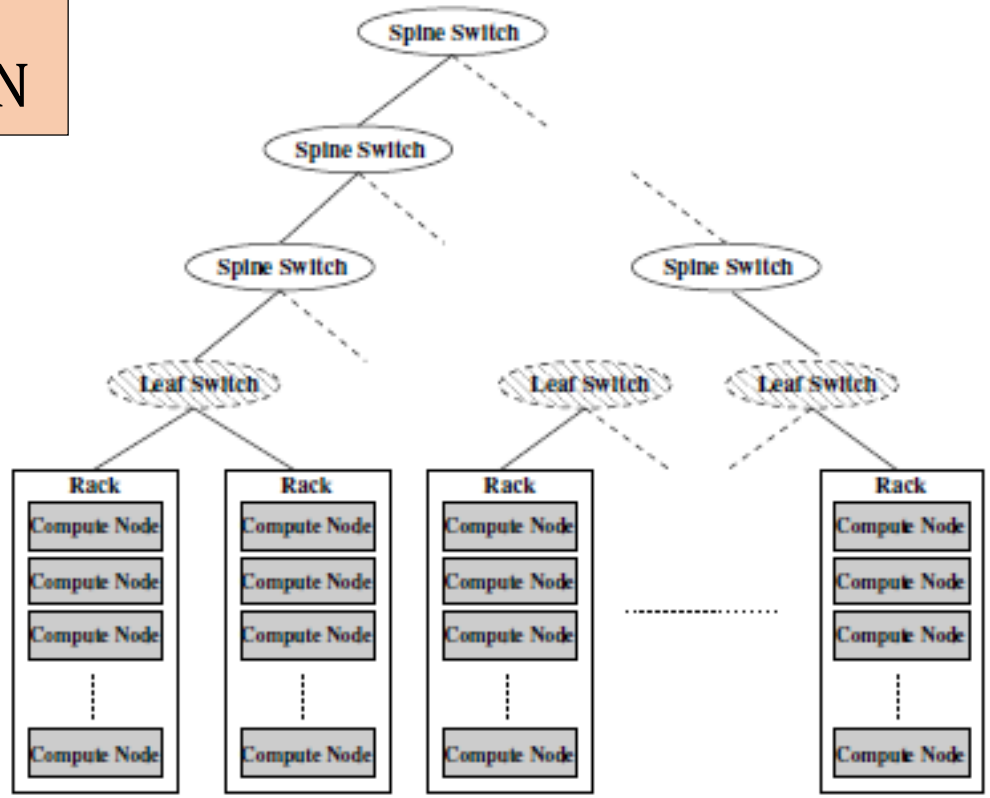
$$T_{binomial} = (t_s\text{-}inter\text{-}node * C_1 + t_s\text{-}intra\text{-}switch * C_2 + \alpha * t_s\text{-}inter\text{-}switch * C_3) + t_w\text{-}intra\text{-}node * (C_1) * (N * \gamma) + t_w\text{-}intra\text{-}switch * (C_2) * (N * \beta) + \alpha * t_w\text{-}inter\text{-}switch * (C_3) * (N * \delta)$$

Communication Cost for Gather (Binomial)

- The bandwidth term is obtained by adding costs at each level
- $[C_1 * \gamma + C_2 * \beta + C_3 * \delta] * N = (p-1)/p * N$

There's a typo in the paper. Read this as ts-intra-node

$$T_{binomial} = (t_s\text{-}inter\text{-}node * C_1 + t_s\text{-}intra\text{-}switch * C_2 + \alpha * t_s\text{-}inter\text{-}switch * C_3) + t_w\text{-}intra\text{-}node * (C_1) * (N * \gamma) + t_w\text{-}intra\text{-}switch * (C_2) * (N * \beta) + \alpha * t_w\text{-}inter\text{-}switch * (C_3) * (N * \delta)$$



Communication Cost Comparison

$$T_{binomial} > (\alpha * t_s\text{-}inter\text{-}switch * C_3) + (\alpha * N * t_w\text{-}inter\text{-}switch * C_3 * \delta)$$

$$T_{topo} > (\alpha * t_s\text{-}inter\text{-}switch * log(R)) + (\alpha * (1 - 1/R))/(M * t_w\text{-}inter\text{-}switch)$$

$$T_{binomial}/T_{topo} = N * \delta * C_3/(M * (1 - 1/R))$$

$$T_{binomial}/T_{topo} = f(R)$$

1. The rack-leader processes independently perform an intra-switch gather operation. This phase of the algorithm does not involve any inter-switch exchanges.
2. Once the rack-leaders have completed the first phase, the data is gathered at the root through an inter-switch gather operation performed over the *R* rack-leader processes.

Experimental Setup

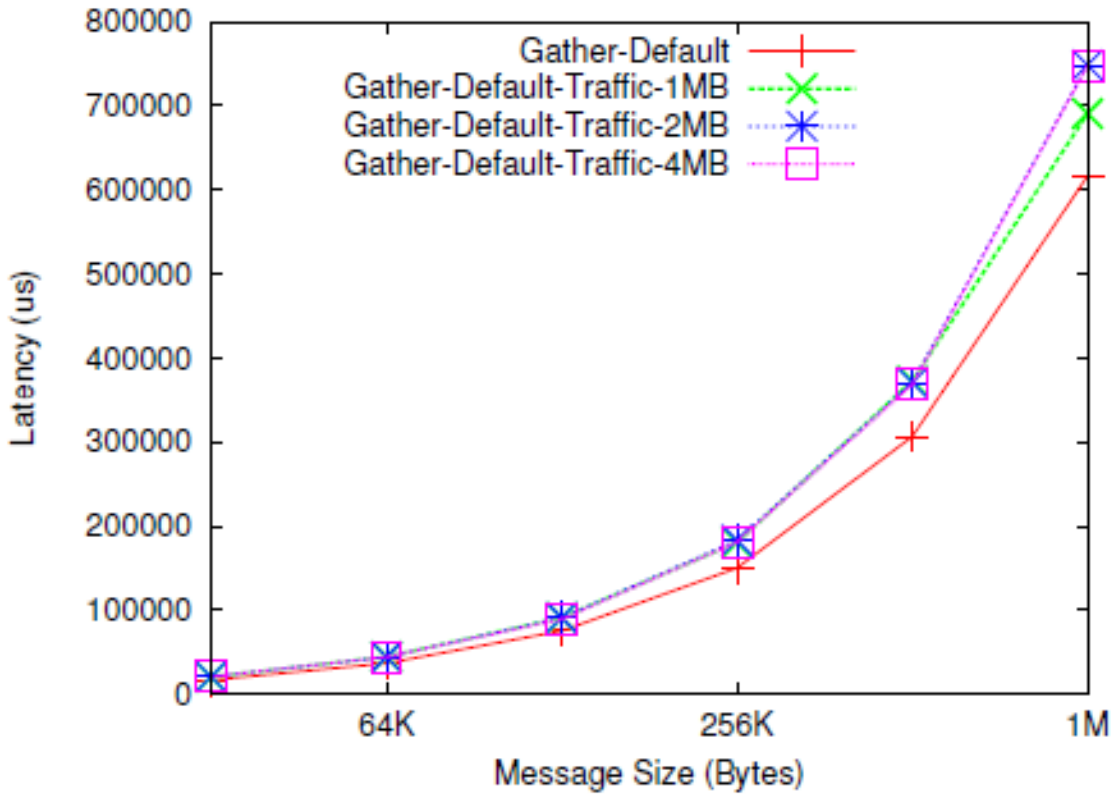
- Three InfiniBand DDR switches A, B and C to create a tree topology.
- Switch A is connected to 8 nodes based on the quad-core, quad-socket AMD Barcelona architecture (4 nodes /64 processes used)
- Switch B is connected to 32 nodes based on the quad-core, dual-socket Intel Clovertown architecture (29 nodes/ 232 processes used)
- Switches A and B are connected to Switch C with two InfiniBand DDR links each.

Benchmark

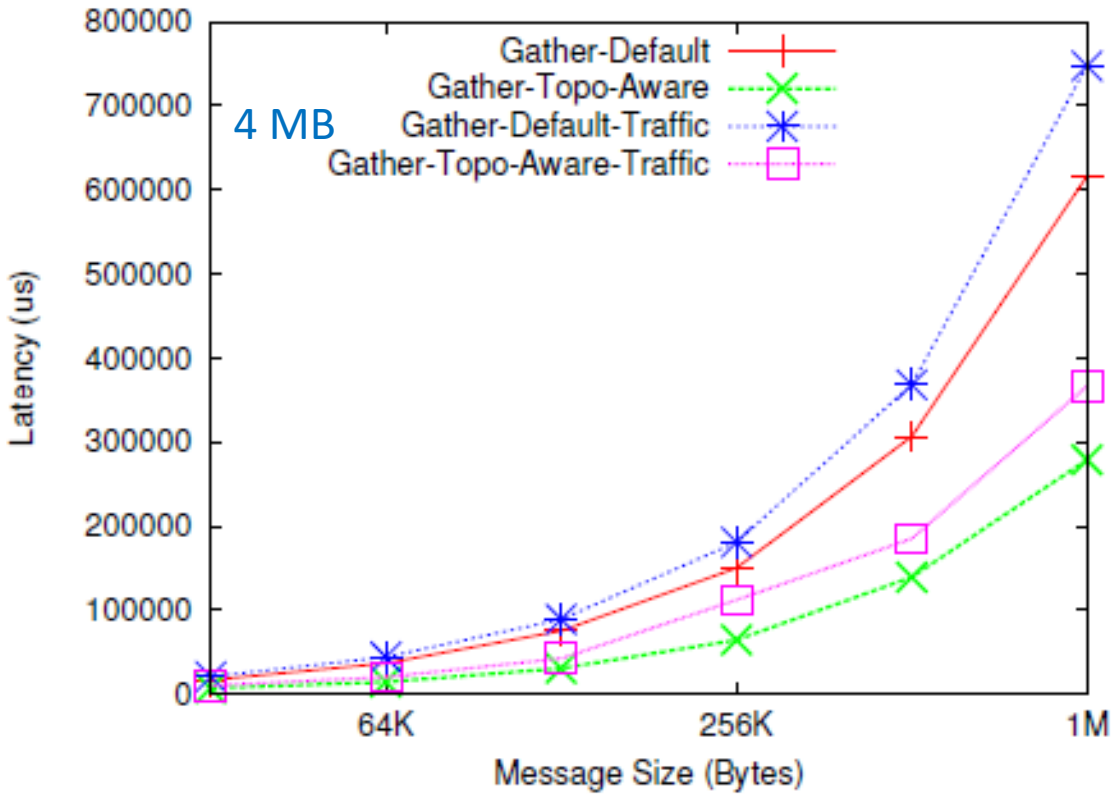
- A simple benchmark code iterates through various message sizes (0 – 1 MB) and invokes a collective call several times in a loop.
- Measured average time when system is quiet
- AlltoAll is used to create background traffic (K/2 in each switch)
  - Constant traffic K*M bytes over the switches

Gather Results (With and Without Traffic)

21% degradation

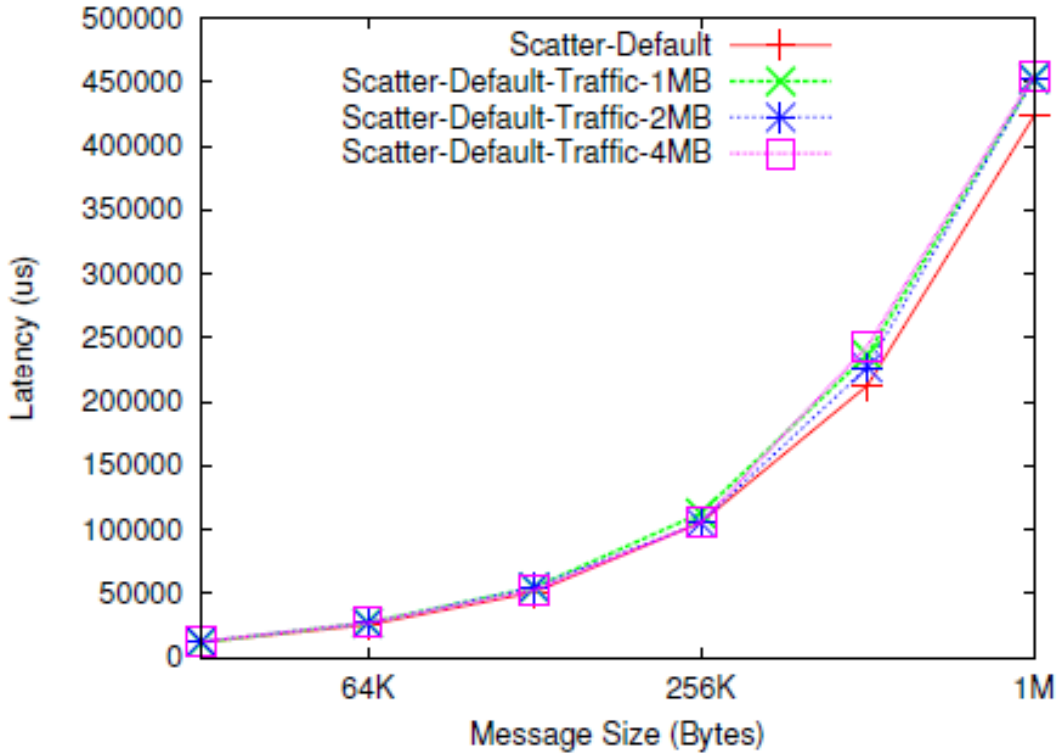


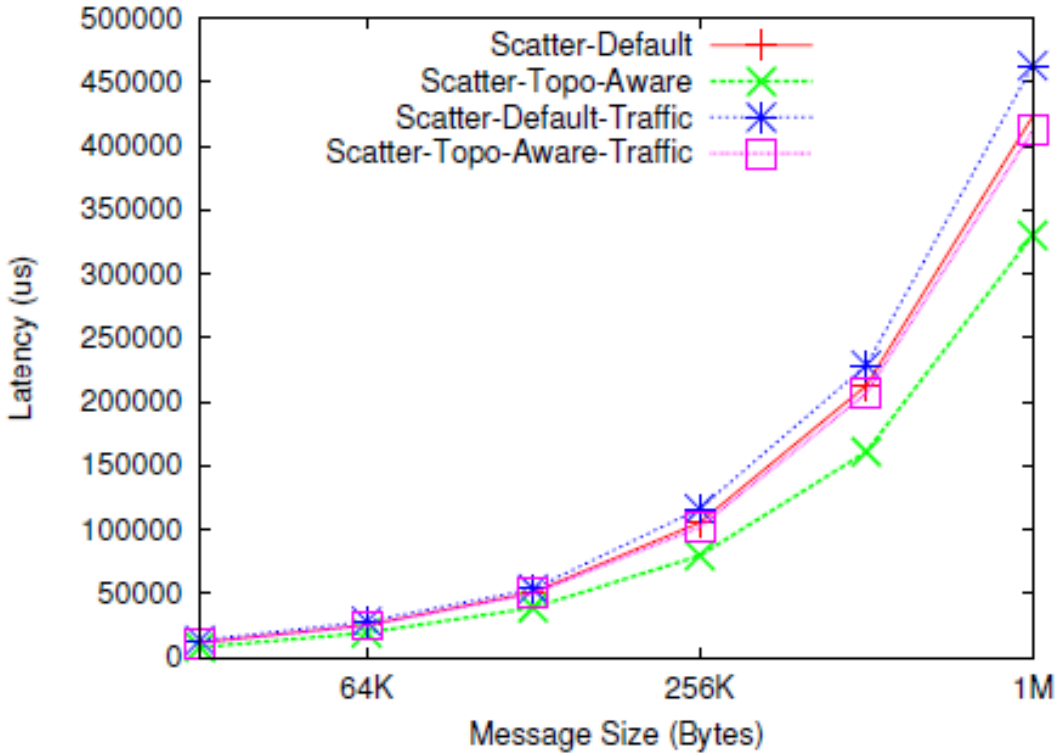
50% reduction



Scatter Results

Homework: T_Scatter < T_Gather?



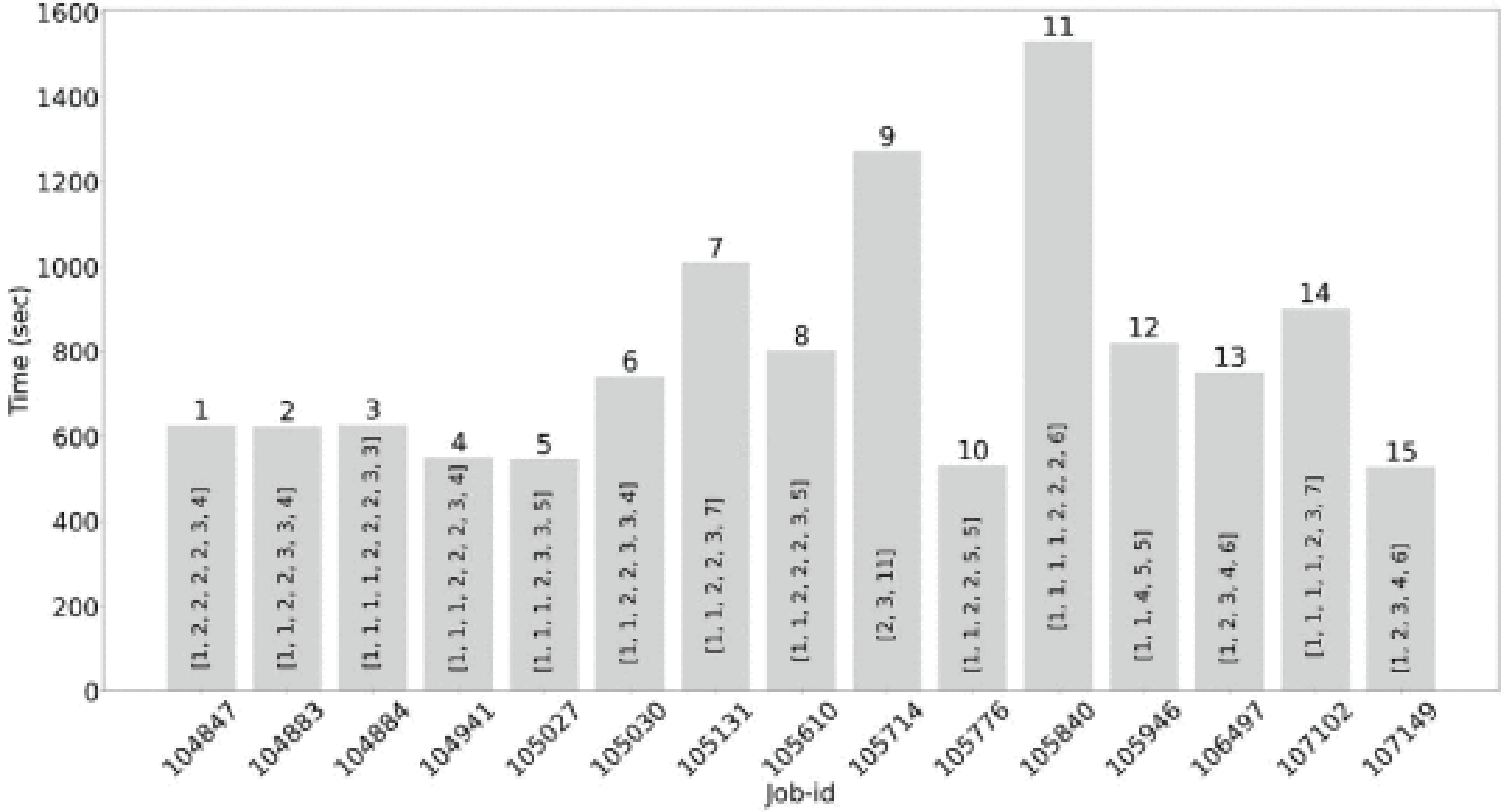


Conclusions

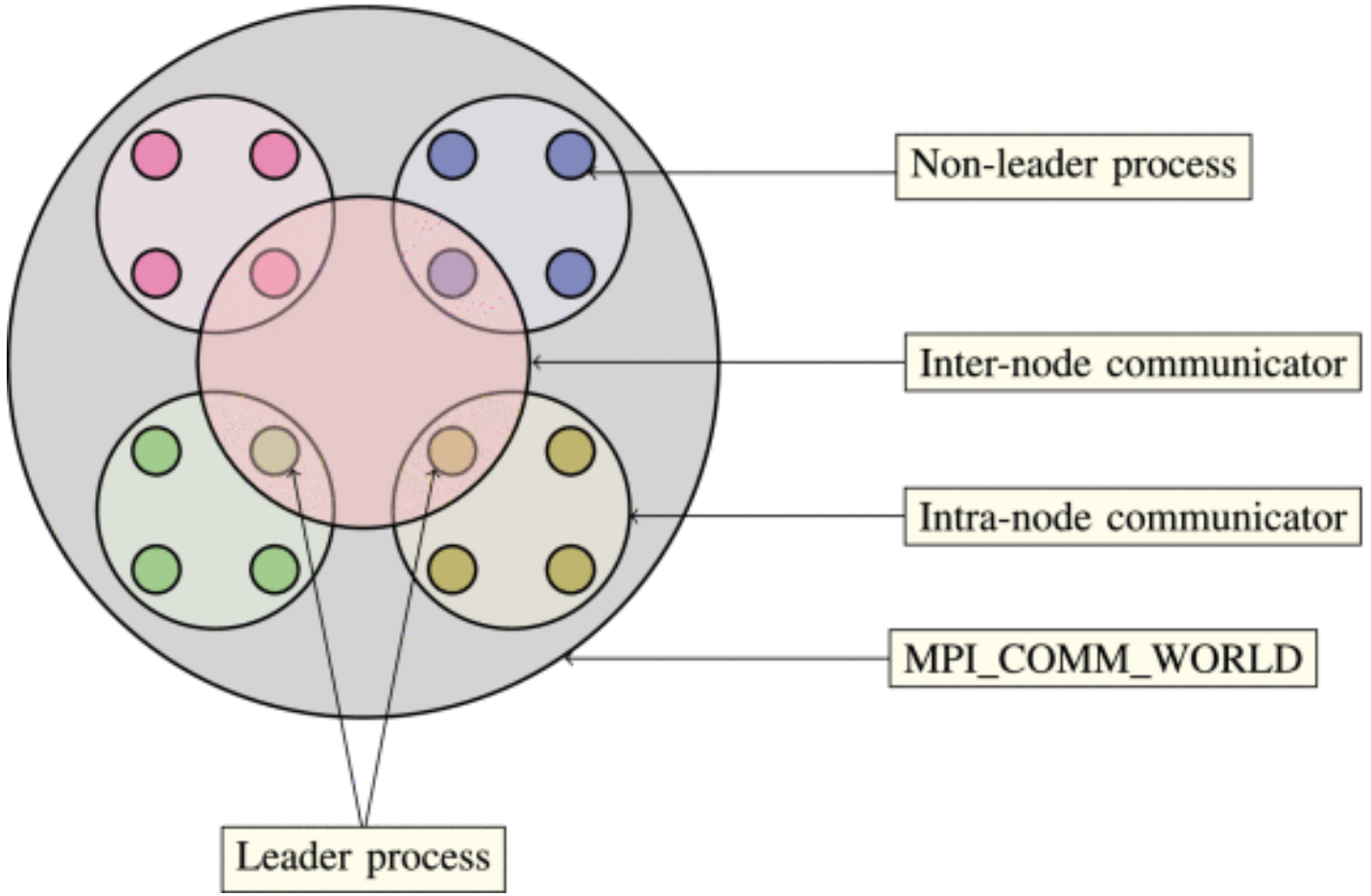
- Presence of background traffic may lead to performance degradation
- Efficient software stack may improve performance
- Topology-aware Gather and Scatter was shown to perform better

Hierarchical Communication Optimization for FFT, M Kumar, P Malakar, SC W HiPar 2022

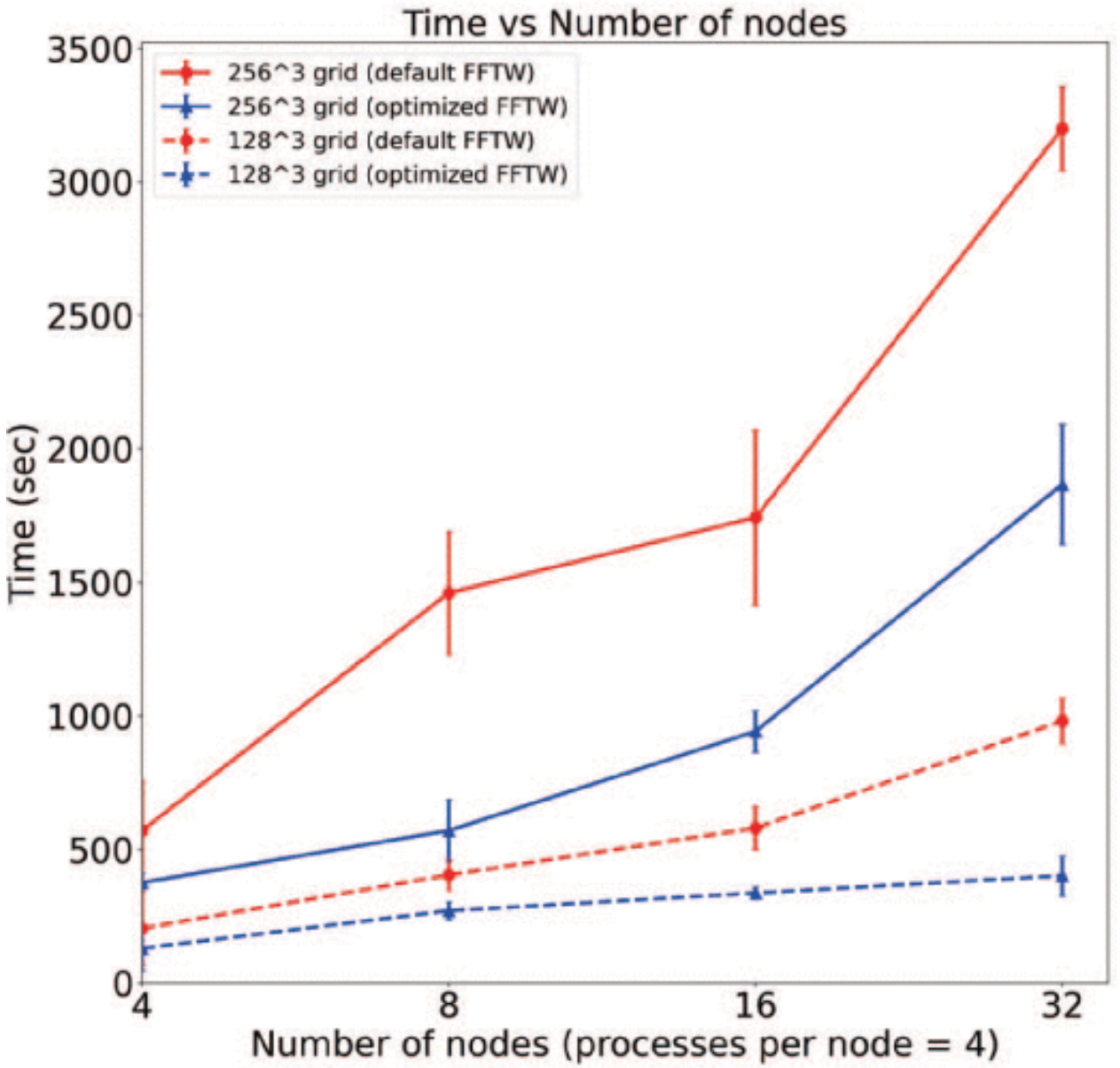
Variation in Job Runtimes



Hierarchical AlltoAll/Sendrecv



Results


Time vs Number of nodes
256^3 grid (default FFTW)
256^3 grid (optimized FFTW)
128^3 grid (default FFTW)
128^3 grid (optimized FFTW)
Time (sec)
0
500
1000
1500
2000
2500
3000
3500
4
8
16
32
Number of nodes (processes per node = 4)