OSU INAM: Profiling and Visualization Tool for Scalable and In-Depth Analysis of High-Performance GPU-enabled HPC Clusters

Pouya Kousha, Kamal Raj, Hari Subramoni, and Dhabaleswar K. (DK) Panda
The Ohio State University
{kousha.2, sankarapandiandayalaganeshr.1, subramoni.1}@osu.edu, panda@cse.ohio-state.edu

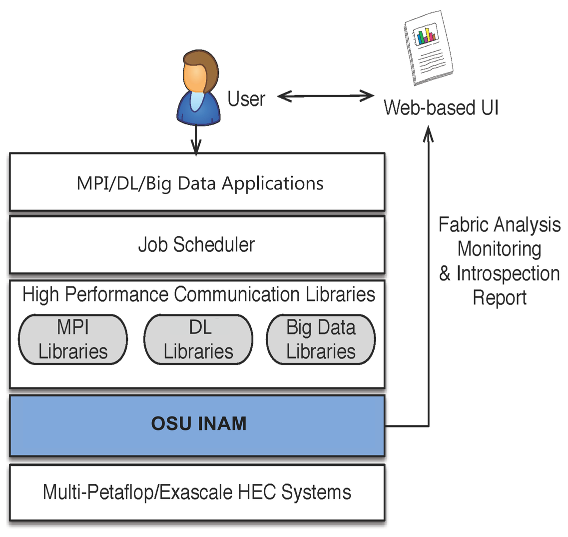
Research Challenges

Broad Challenge

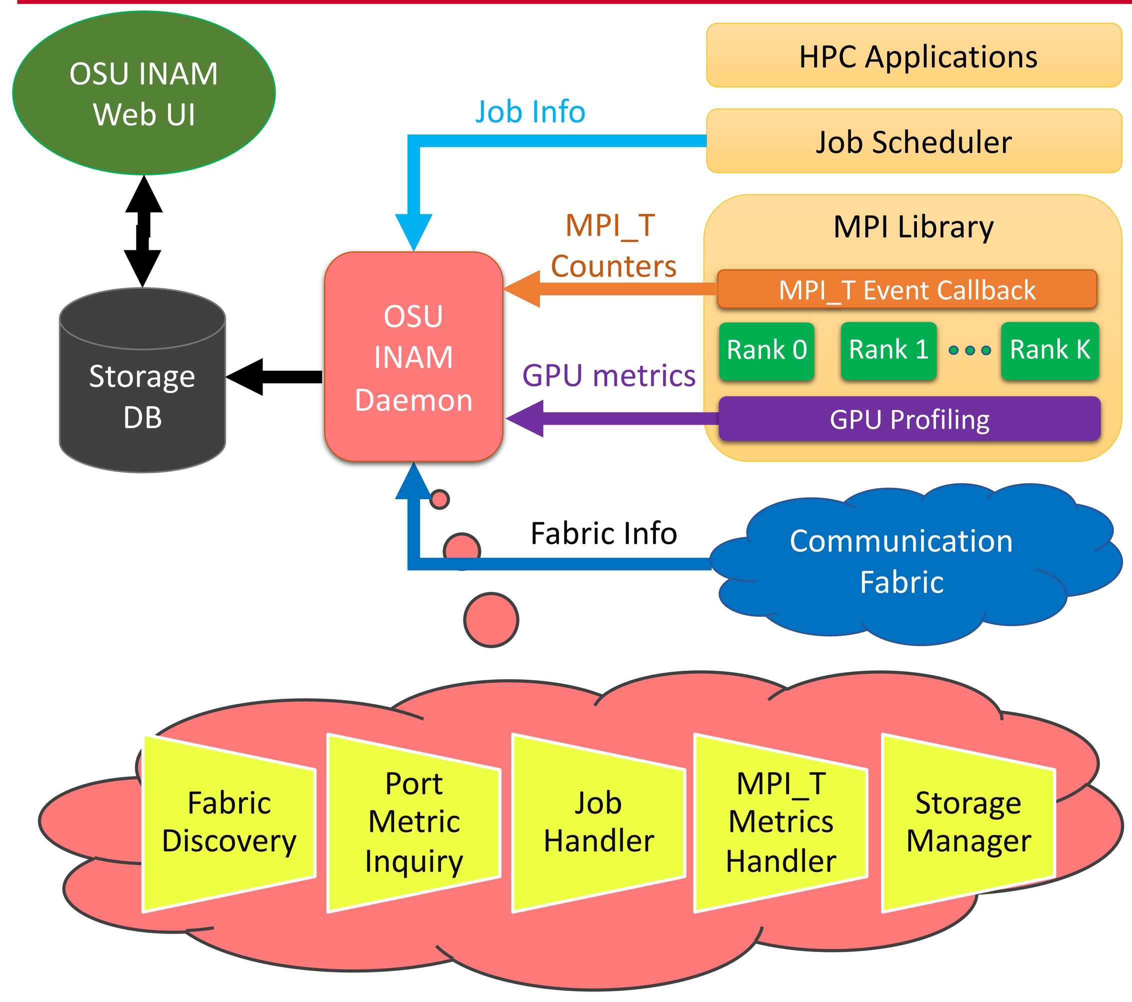
How can we design a tool that enables holistic real-time, scalable and in-depth understanding of communication traffic on the interconnect and GPU through tight integration with the MPI runtime and job scheduler?

Research Challenges and Contributions

1. Enhance gathering, storing, retrieving, and visualization of the metrics for large and complex HPC networks with low latency
2. Introduce MPI_T event-based metrics for point-to-point and collective MPI communication patterns
3. Design a profile-enabled communication library for to gather GPU and MPI performance counters using CUPTI and MPI_T interfaces, respectively
4. Present a real-time and low-overhead profiler tool to correlate MPI-level with network-level metrics

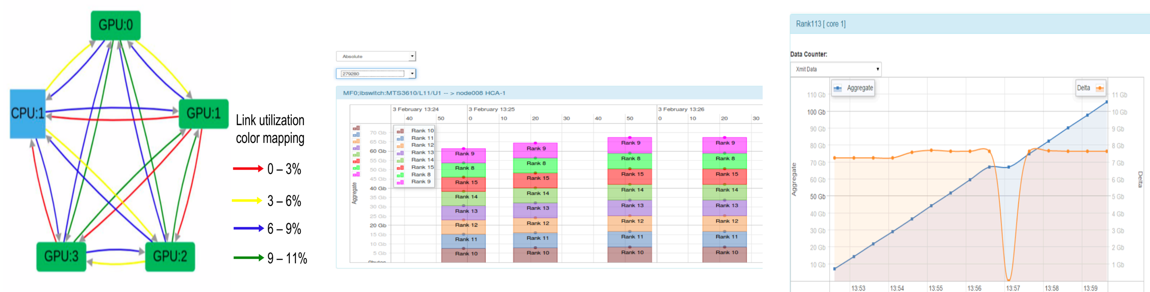


Proposed Framework

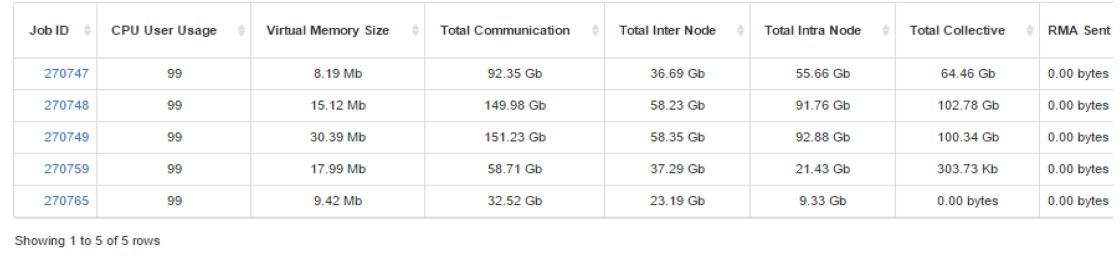


Usage Scenarios

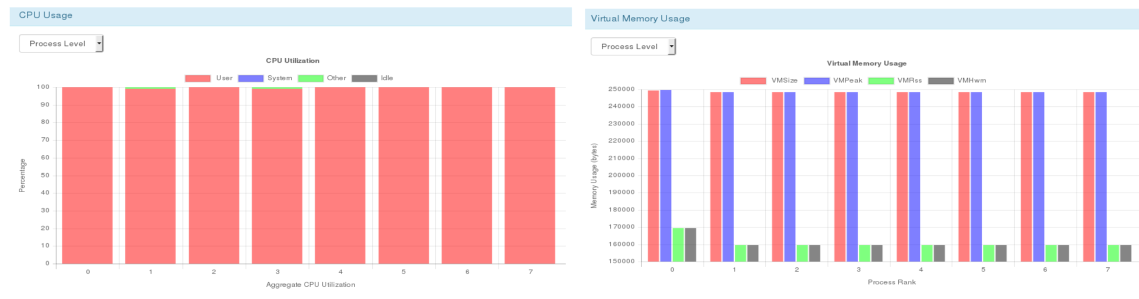
- Identifying and Analysing Sources of Interconnect and Link Congestion



- Monitoring Jobs Based on Various Metrics



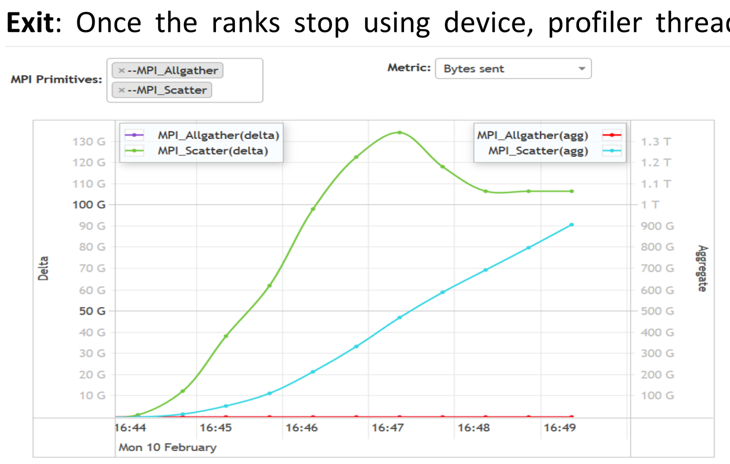
- Profiling and Reporting Performance Metrics at Different Granularities



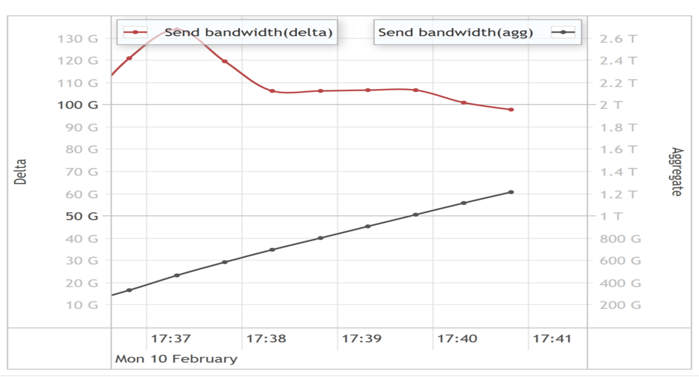
High-Performance, Low Overhead, and Scalable GPU Profiling Module

Phases in Intra-node GPU Metric Collection

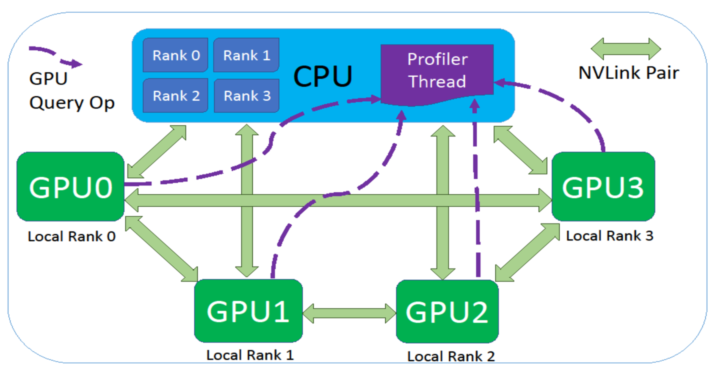
- **Startup**: Each rank discovers the topology and updates shared region. Then, one rank per node setups and starts a profiler thread on CPU to profile all GPUs on the node once using GPUs.
- **Query**: The profiler thread profile all enrolled GPUs based on user defined interval and send data to the tool server periodically
- **Exit**: Once the ranks stop using device, profiler thread



Screenshot of PVAR Chart. The X-axis represents current time and Y-axis represents the number of bytes sent over the network reported to the tool server



Screenshot of NVLink Metrics chart for TensorFlow. The X-axis represents time and Y-axis represents the link bandwidth utilization



Collection of GPU metrics

The GPU metrics will be correlated to MPI_T information at Web based UI of the tool.

Scalability of the Design

- Each node aggregates and sends the GPU and PVAR metrics to the server
- The metrics scale linearly based on the number of GPUs per node

TIMING OF THE GPU PROFILER THREAD PHASES FOR EACH NODE. EACH NODE HAS FOUR GPUS

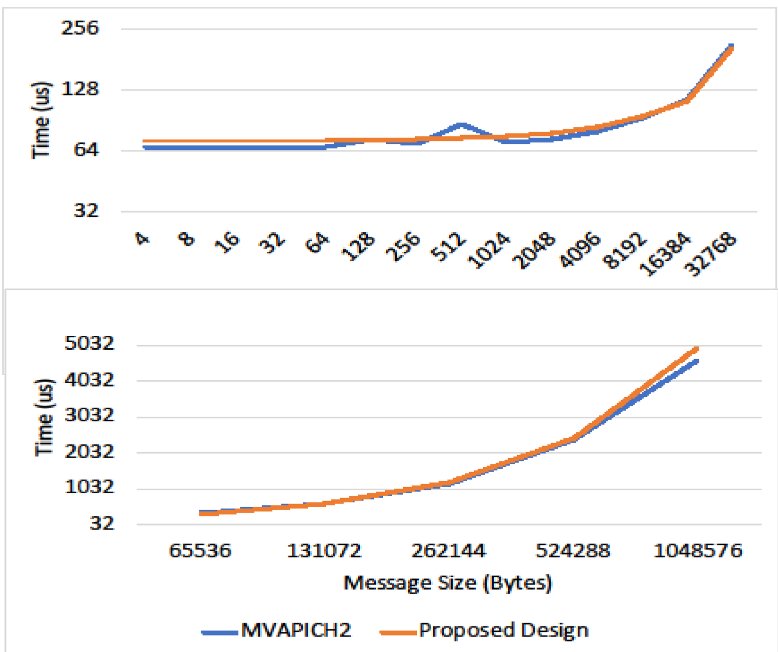
Metrics	Average	Min	Max	STDEV.p
Startup phase	1.632 s	1.561 s	1.672 s	0.035 s
CUDA context create	1.624 s	1.548 s	1.663 s	0.035 s
Query phase	2.33 ms	1.63 ms	208.03 ms	4.43 ms
Exit phase	88 us	85 us	93 us	28 us

OVERHEAD OF COLLECTING PVAR DATA AT NANOSECOND GRANULARITY

Metrics	Average	Min	Max	STDDEV.p
Collecting PVARs	517.63 ns	140 ns	16,204 ns	305.91 ns

Overhead of Proposed Designs on End-To-End Performance

- Very low (~5%) overhead caused on end-to-end performance



Overhead of proposed designs on End-To-End Performance MPI_Allreduce for different message sizes

Intra_node_topo
Id (primary key)
Node_name
Physical_link_count
Link_capacity
Source
Source_id
Destination
Destination_id

NVLink_metrics	
Id (primary key)	Source_local_ran k
Link_id	Source_global_rank
Node_name	Dest_local_rank
Source_name	Dest_global_rank
Source_port	Data_unit
Source_id	Data_recv
Dest_name	Data_sent
Dest_port	Data_recv_rate
Dest_id	Data_sent_rate
Added_on	

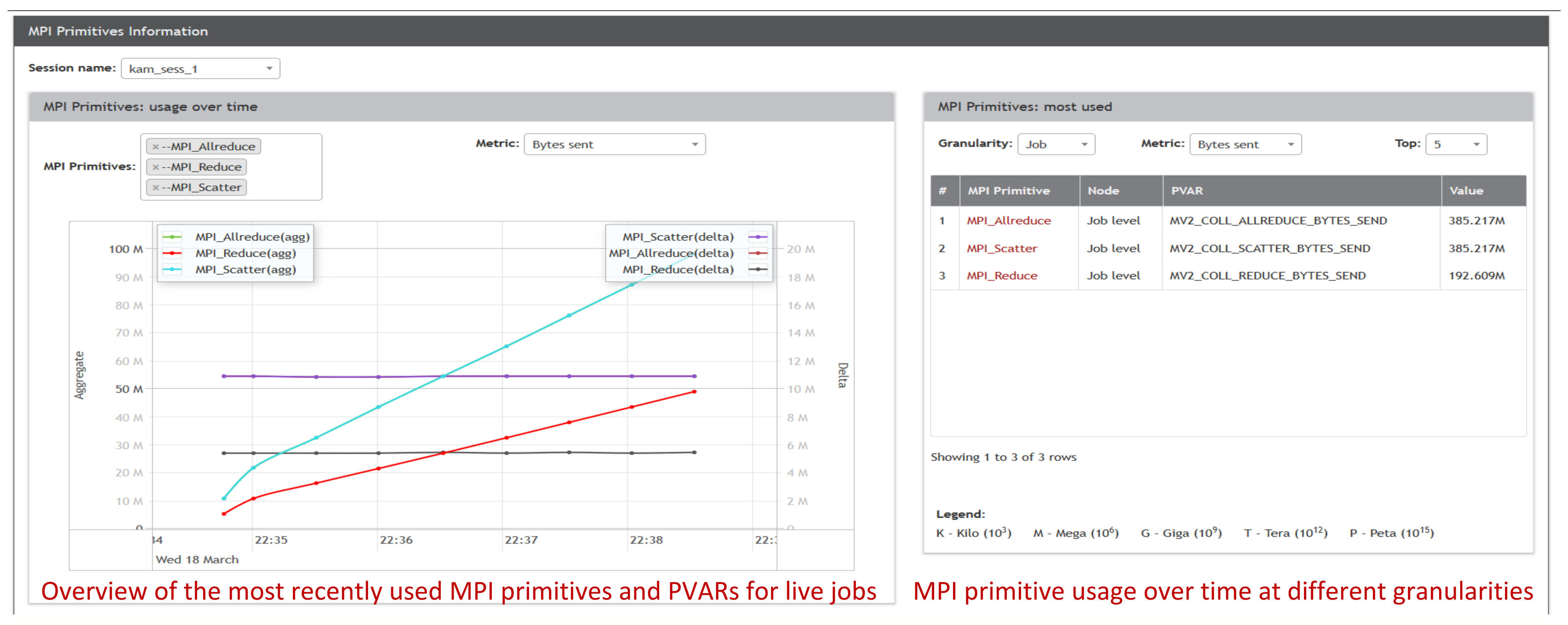
PVAR_table
Id (primary key)
jobid
Node_name
Start_time
End_time
Bytes_recv
Bytes_sent
PVAR_name
Algorithm
Source_rank
Dest_rank
Added_on

Table schemas for storing NVLink metrics, PVARs, and the Topology

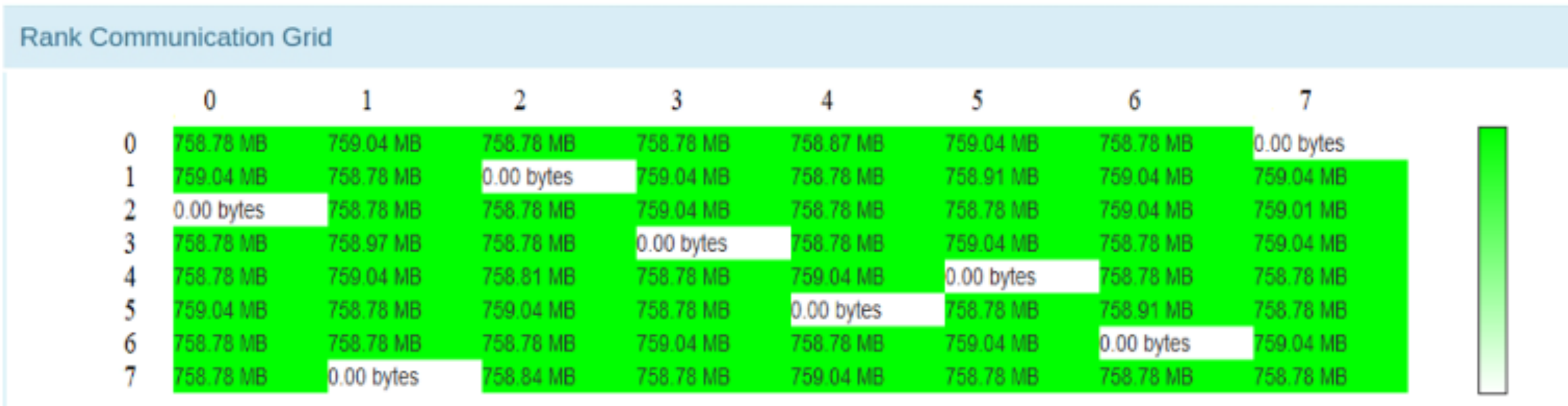
MPI_T Introspection

- MPI_T information will be sent from MPI library to OSU INAM Daemon
- OSU INAM web UI shows live view of MPI_T data for different levels like node/job and the entire cluster
- User can see which algorithms are used for MPI operation and interplay of MPI modules for live and historical jobs

Overview of the most recently used MPI primitives for inter and intra node data exchange and PVARs for live jobs

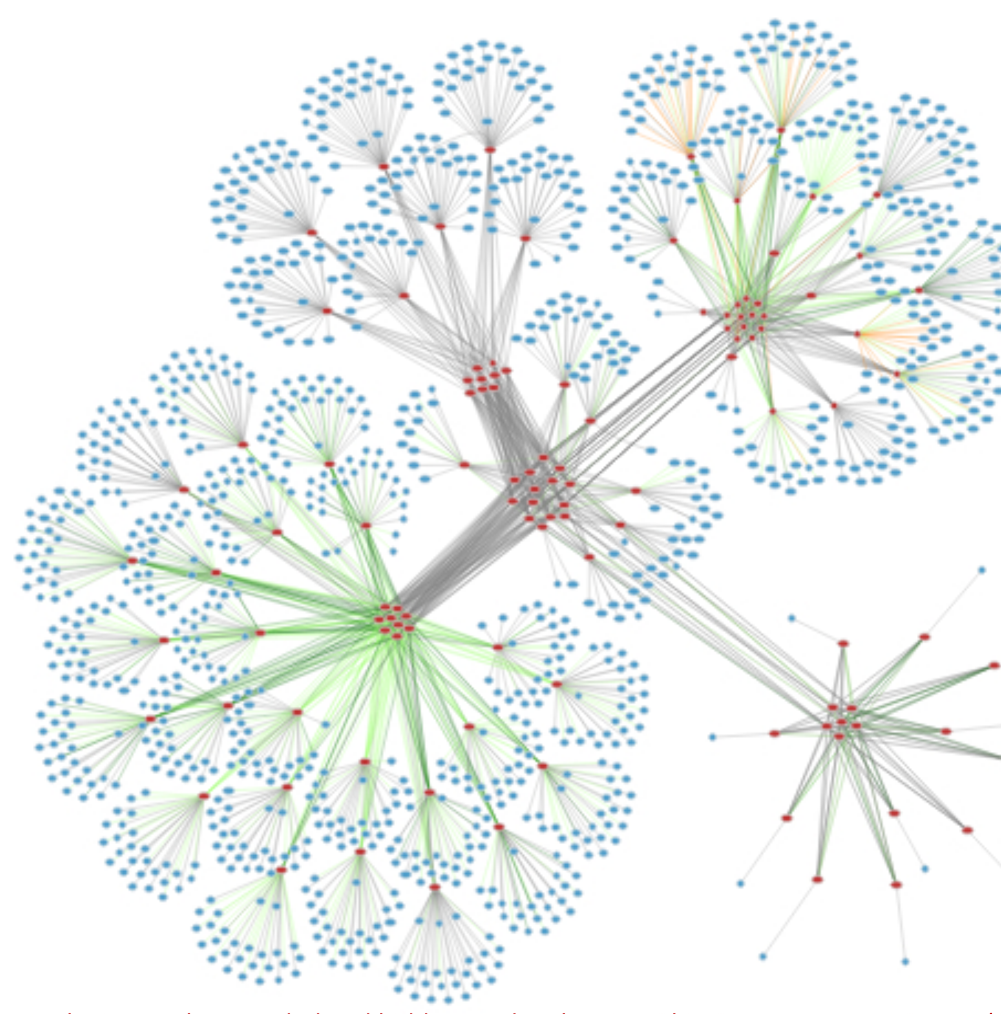


Overview of the most recently used MPI primitives and PVARs for live jobs   MPI primitive usage over time at different granularities

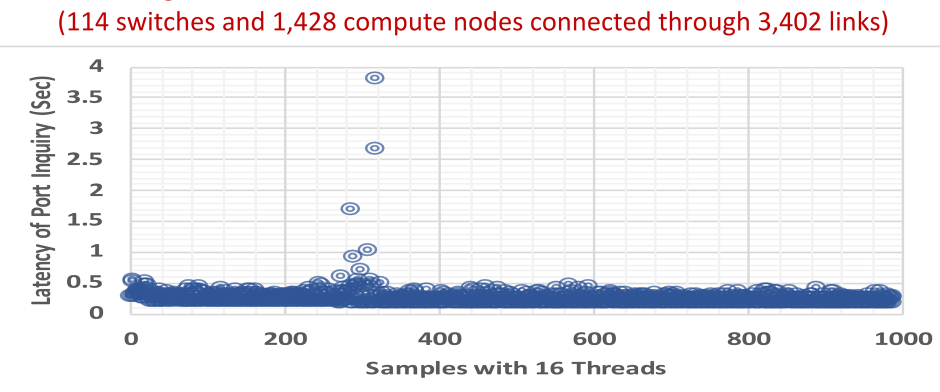


Rank level communication grid, each element (i,j) in the grid represents the amount of data transferred from rank i to rank j. This matrix depicts MPI_Allreduce operation

Enhanced Fabric Discovery and Port Metrics Inquiry



Network View with expanded and hidden modes showing Ohio Supercomputer Center (OSC) with 3 heterogeneous clusters all connected to the same InfiniBand Fabric (114 switches and 1,428 compute nodes connected through 3,402 links)



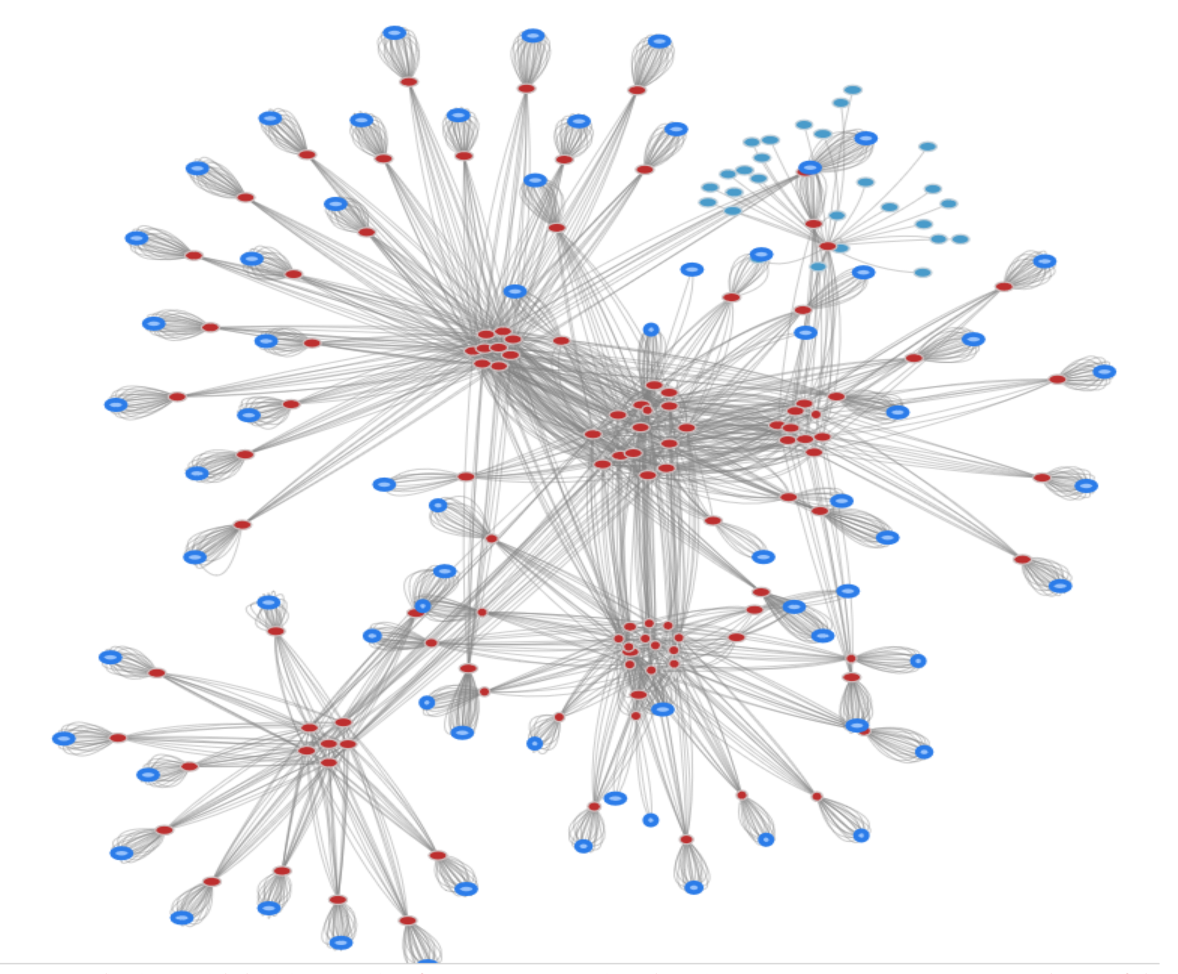
Histogram of remotely querying port metrics for all nodes for OSC

Challenges and Solutions

- As HPC systems are becoming larger and expect to have better capabilities like real-time profiling at fine granularity and scalability concerns a newer set of challenges arise
- Interval to read the hardware counters should be low to ensure fine granularity
- Uses different levels of threading, bulk insertions and deletions for storing, and parallel components for Fabric Discovery and Port Metric Inquiry

NETWORK AND LIVE JOBS VIEW GENERATION TIMING ON OSC WITH 1K JOBS

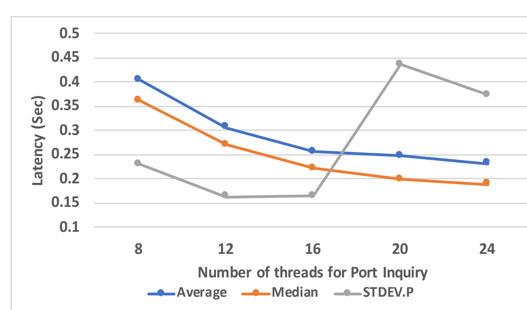
View	Average	Min	Max	STDEV.p
Network View	196.15 ms	187 ms	206.09 ms	5.75 ms
Live Jobs View	18.17 ms	16 ms	20 ms	1 ms



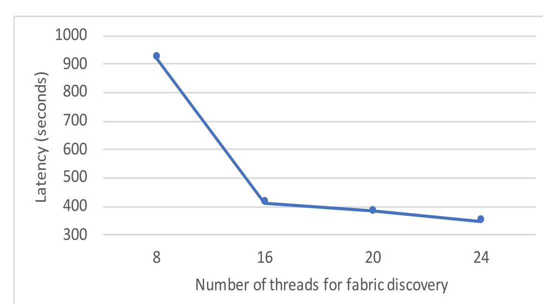
Network View with hidden modes for compute nodes showing Frontera supercomputer cluster fabric (494 switches and 8,811 compute nodes connected through 22,819 links)

Performance Evaluation

- Enhanced performance for fabric discovery using optimized OpenMP-based multi-threaded designs with **14x** speedup
- Ability to remotely gather InfiniBand performance counters at sub-second granularity for very large (>2,000 nodes) clusters



Impact of multi-threading on Port Inquiry module on OSC cluster



Impact of multi-threading on Fabric Discovery module on OSC cluster

Release & Research Dissemination

- The tool is publicly released as OSU INAM v0.9.6 for x86-64, ARM and OpenPower
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
  - More than 600 downloads with support for PBS and SLURM
- Community engagement with: OSC @ USA, NOAA @ USA, U. of Utah @ USA, CAE Services @ Germany, Pratt & Whitney, Ghent University @ Germany, Cyfronet @ Poland, and Georgia Tech Univ @ USA.

Future Work

- Extend data collection server to further collect intra-node, intra-node, I/O, and power metrics
- Support to profile multiple MPI libraries
- Extending support for HPC, Big Data and DL applications layer

THE OHIO STATE UNIVERSITY   MVAPICH

Reference:
1) Designing a Profiling and Visualization Tool for Scalable and In-Depth Analysis of High-Performance GPU Clusters, **P. Kousha**, B. Ramesh, K. Kandadi Suresh, C. Chu, A. Jain, N. Sarkauskas, H. Subramoni, D. Panda. IEEE HiPC, Dec 2019
2) Accelerated Real-time Network Monitoring and Profiling at Scale using OSU INAM, P. Kousha, S. D. Kamal Raj, H. Subramoni, D. Panda, H. Na, T. Dockendorf, K. Tomko. PEARC 2020, Jul 2020