NSF CSSI PI Meeting, Seattle, WA, Feb. 13-14, 2020

SI2-SSI (2020): FAMII: High-Performance and Scalable Fabric Analysis, Monitoring and Introspection Infrastructure for HPC and Big Data

PI: Dhabaleswar K. (DK) Panda, Co-PIs: Karen Tomko, Hari Subramoni, Heechang Na
Institutions: The Ohio State University

Award #: OAC-1664137

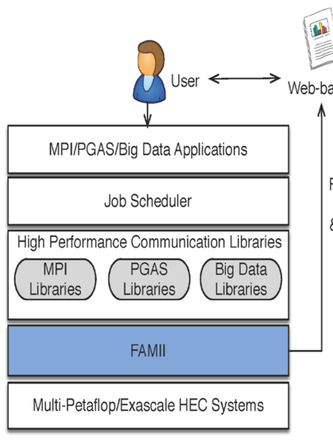
Research Challenges

Broad Challenge

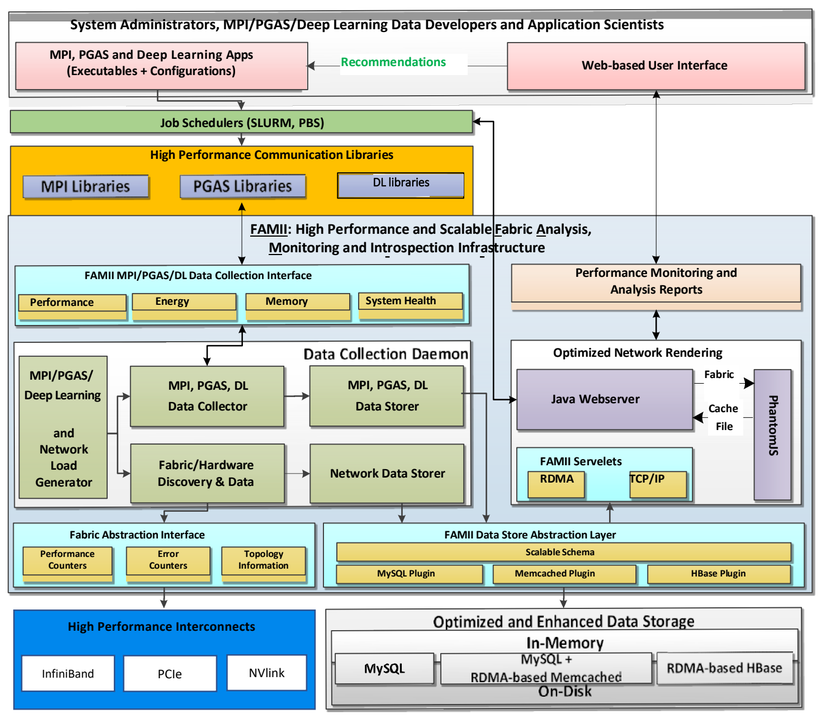
Can a high performance and scalable tool be designed which is capable of analyzing and correlating the communication on the fabric with behavior of HPC/Big Data/Deep Learning applications through tight integration with the communication runtime and the job scheduler?

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 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



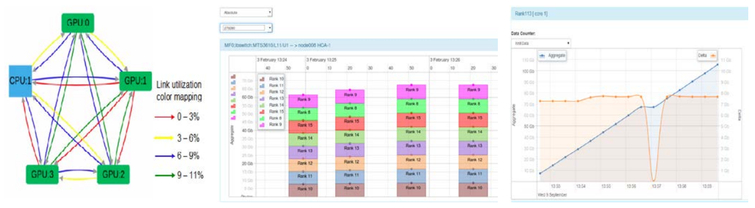
Proposed Framework



The Proposed Performance Monitoring, Analysis, and Introspection Framework

Usage Scenarios

- Identifying and Analysing Sources of Interconnect and Link Congestion

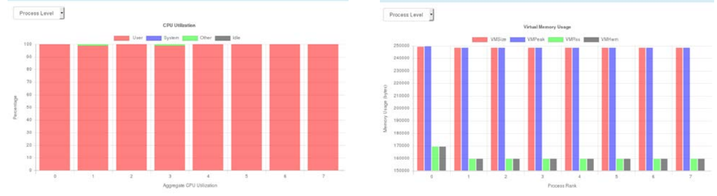


- Monitoring Jobs Based on Various Metrics

Job ID	CPU User Usage	Virtual Memory Size	Total Communication	Total Inter Node	Total Intra Node	Total Collective	RMA Sent
270747	99	8.19 Mb	92.35 Gb	36.69 Gb	55.66 Gb	64.46 Gb	0.00 bytes
270748	99	15.12 Mb	149.98 Gb	58.23 Gb	91.76 Gb	102.78 Gb	0.00 bytes
270749	99	30.39 Mb	151.23 Gb	58.35 Gb	92.88 Gb	100.34 Gb	0.00 bytes
270759	99	17.99 Mb	58.71 Gb	37.29 Gb	21.43 Gb	303.73 Kb	0.00 bytes
270765	99	9.42 Mb	32.52 Gb	23.19 Gb	9.33 Gb	0.00 bytes	0.00 bytes

Showing 1 to 5 of 5 rows

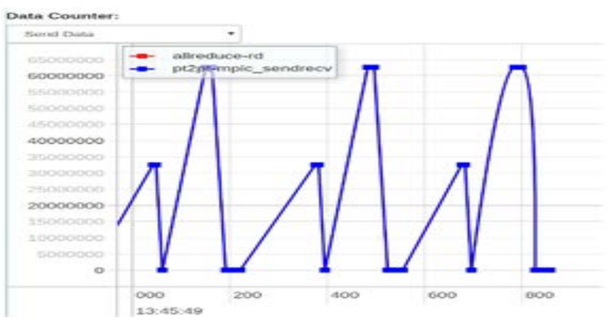
- Profiling and Reporting Performance Metrics at Different Granularities



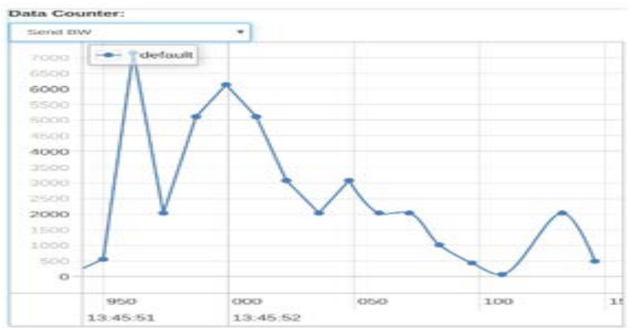
High-Performance, Low Overhead, and Scalable GPU Profiling Module

Phases in Intra-node GPU Metric Collection

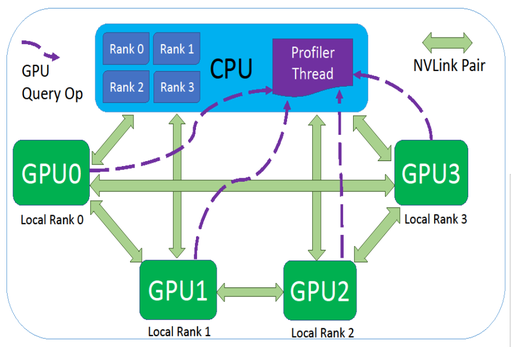
- **Startup**: Each rank discovers the topology and updates shared region. Then, one rank per node setups and starts a profiler thread on CPU to profile all GPUs on the node once using GPUs.
- **Query**: The profiler thread profile all enrolled GPUs based on user defined interval and send data to OSU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



Screenshot of PVAR Chart. The X-axis represents current time and Y-axis represents the number of bytes sent over the network to the OSU INAM daemon



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 OSU INAM Web UI

Scalability of the Design

- Each node aggregates and sends the GPU and PVAR metrics to the OSU INAM daemon
- 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

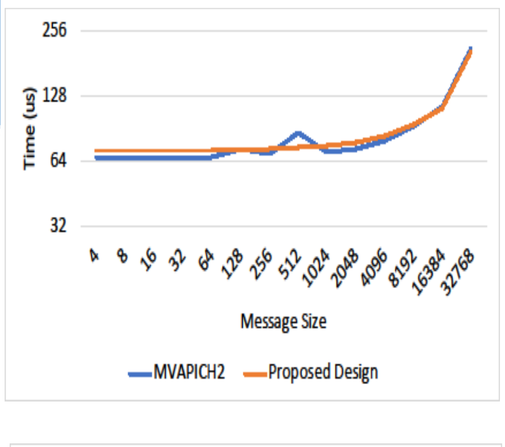
Metrics	Average	Min	Max	STDEV.p
Startup phase	1.632 s	1.561 s	1.672 s	0.035 s
CUDA context create	1.624 s	1.548 s	1.663 s	0.035 s
Query phase	2.33 ms	1.63 ms	208.03 ms	4.43 ms
Exit phase	88 us	85 us	93 us	28 us

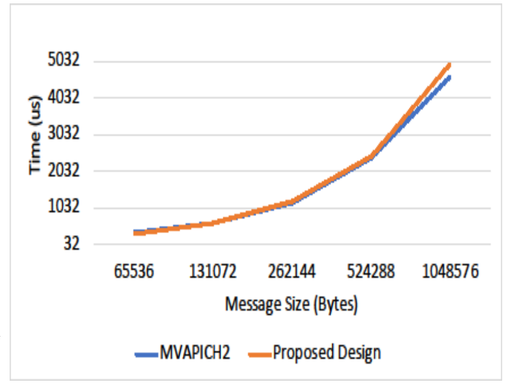
OVERHEAD OF COLLECTING PVAR DATA AT NANOSECOND GRANULARITY

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 by profiling on end-to-end performance

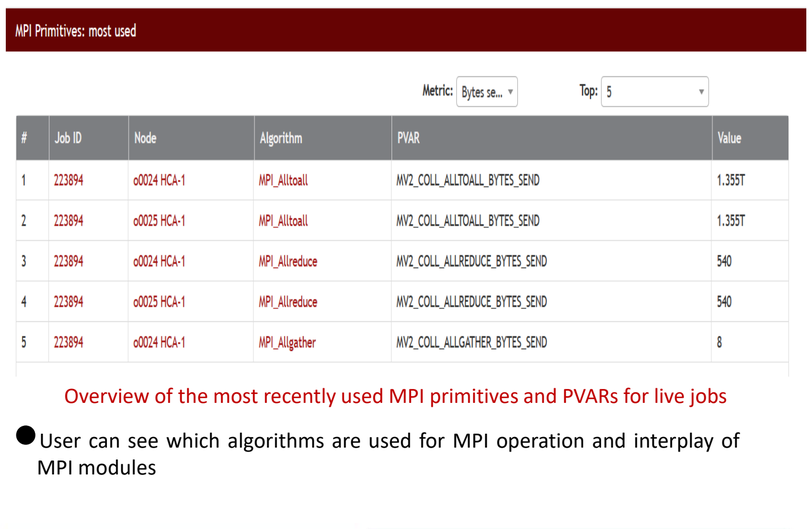




Overhead of proposed designs on End-To-End Performance
MPI_Allreduce for different message sizes

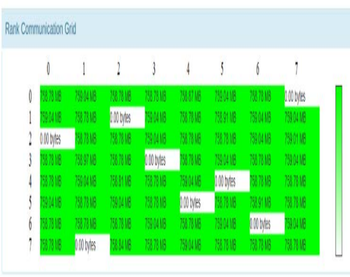
MPI_T Introspection

- MPI_T information will be sent from MPI library to OSU INAM Daemon
- OSU INAM receives and shows live view of MPI_T data for different levels like node/job and the cluster

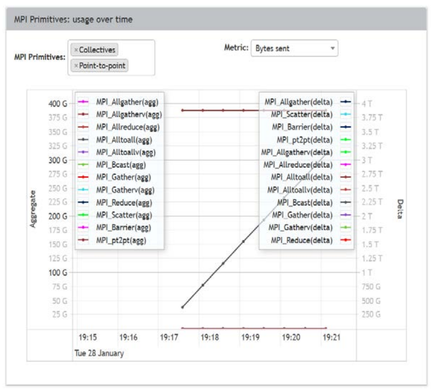


Overview of the most recently used MPI primitives and PVARs for live jobs

- User can see which algorithms are used for MPI operation and interplay of MPI modules

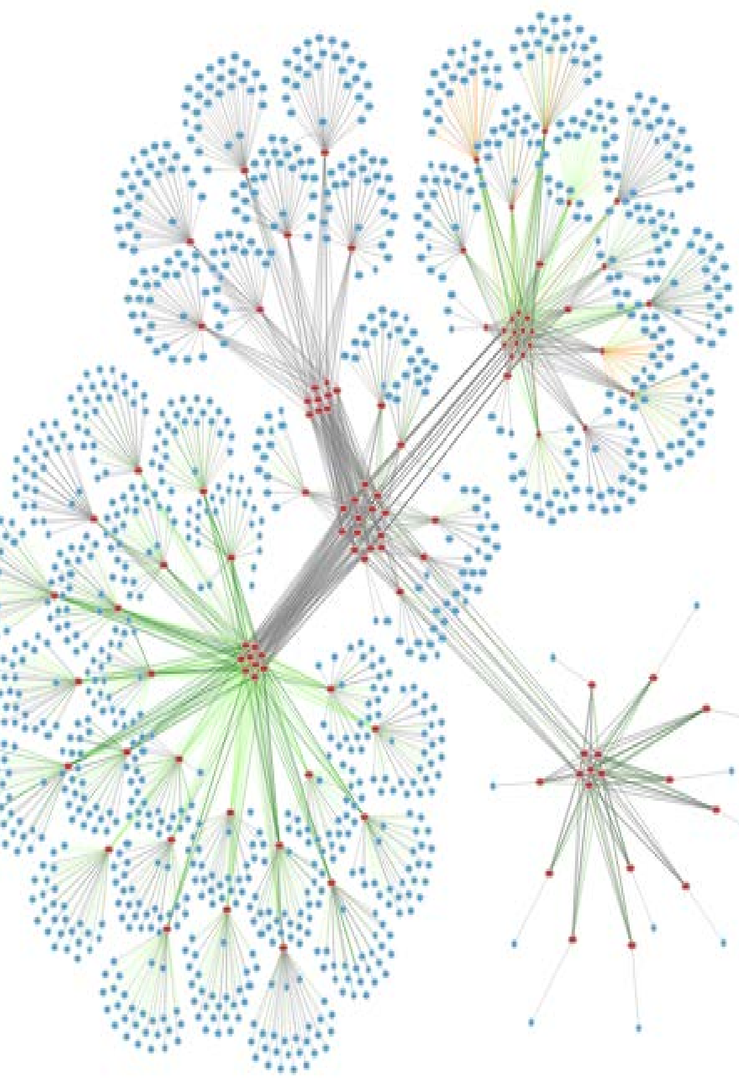


Rank level communication grid, each element (i,j) in the grid represents the amount of data transferred from rank i to rank j. This matrix depicts MPI_Allreduce operation



MPI primitive usage over the time for job/node/cluster level

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)

Challenges and Solutions

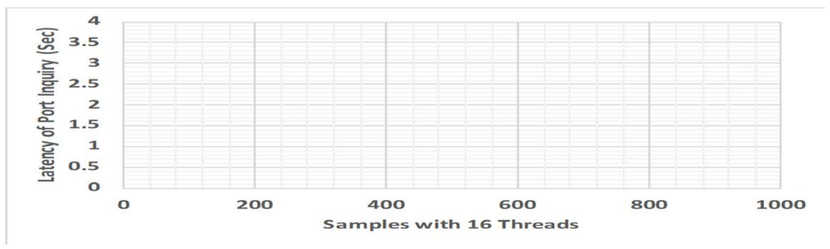
- Designing a low overhead and scalable tool to discover the fabric topology and gather fabric metrics is a burdensome task since it involves 1) profiling of the data and aggregation, 2) storing and 3) rendering the data.
- Interval to read the hardware counters should be low to ensure they do not overflow
- Uses different levels of threading, bulk insertions and deletions for storing, and parallel components for Fabric Discovery and Port Metric Inquiry

Performance Evaluation

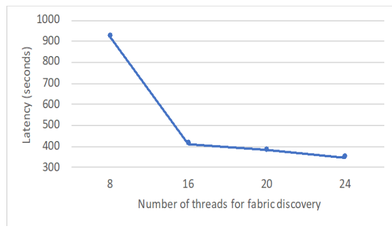
- Enhanced performance for fabric discovery using optimized OpenMP-based multi-threaded designs with **14x** speedup
- Ability to gather InfiniBand performance counters at sub-second granularity for very large (>2,000 nodes) clusters

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

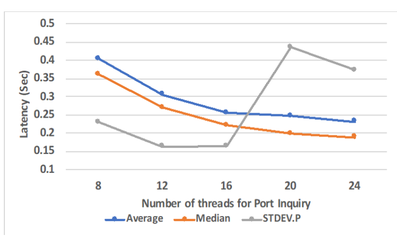
View	Average	Min	Max	STDEV.p
Network View	196.15 ms	187 ms	206.09 ms	5.75 ms
Live Jobs View	18.17 ms	16 ms	20 ms	1 ms



Histogram of querying port metrics for all nodes for OSC



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



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

Research Publications

- 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, D. Panda, IEEE HiPC, Dec 2019
- A. Ruhela, H. Subramoni, S. Chakraborty, M. Bayatpour, P. Kousha, and DK Panda, Efficient Design for MPI Asynchronous Progress without Dedicated Resources , Parallel Computing - Systems & Applications, Volume 85, July 2019.

Software Release, Community Engagement and Metrics

- A v0.9.5 release of OSU INAM has been made on Jan'20
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
  - More than 600 downloads with support for PBS and SLURM
- This release has been installed at OSC and OSU to monitor clusters
- Tutorials at SC '19, ISC'19, HiPEAC '20, MUG'19
- Community Engagement with: NOAA, U. of Utah, CAE Services @ Germany, Pratt & Whitney, Ghent University @ Germany, and Cyfronet @ Poland

Future Work

- As part of future work we aim to:
  - Extend data collection daemon to further intra-node metrics, intra-node communication matrix, and power metrics
  - Support to profile multiple MPI libraries through MPI_T interface
  - Extend support for introspection of PGAS and DL applications

Supported by: OAC-1664137

THE OHIO STATE UNIVERSITY · MVAPICH · Ohio Supercomputer Center — An OH·TECH Consortium Member