16th ANNUAL WORKSHOP 2020

Visualize and Analyze your Network Activities using OSU INAM

Hari Subramoni, Pouya Kousha, Kamal Raj Ganesh, Dhabaleswar K. (DK) Panda

The Ohio State University

E-mail: panda@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda

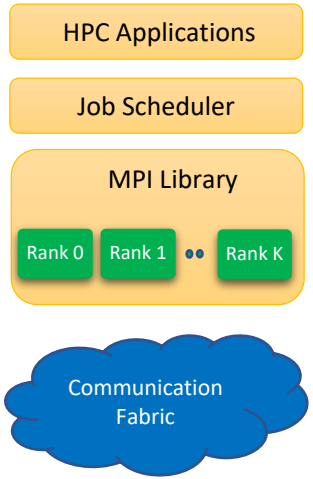
OUTLINE

- Introduction & Motivation
- Design of OSU INAM
- Impact of Profiling on Application Performance
- Usage Scenarios
- Conclusions & Future Work

PROFILING TOOLS PERSPECTIVE AND CHALLENGES

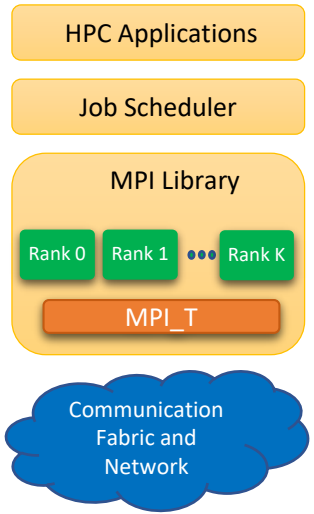
- There are 30+ profiling tools for HPC systems
- System level vs User level
  - User level novelty
- Different set of users have different needs
  - HPC administrators
  - HPC Software developers
  - Domain scientists
- Different HPC layers to profile
  - How to correlate them and pinpoint the problem source?

Unified and holistic view for all users



PROFILING TOOLS PERSPECTIVE AND CHALLENGES (CONT.)

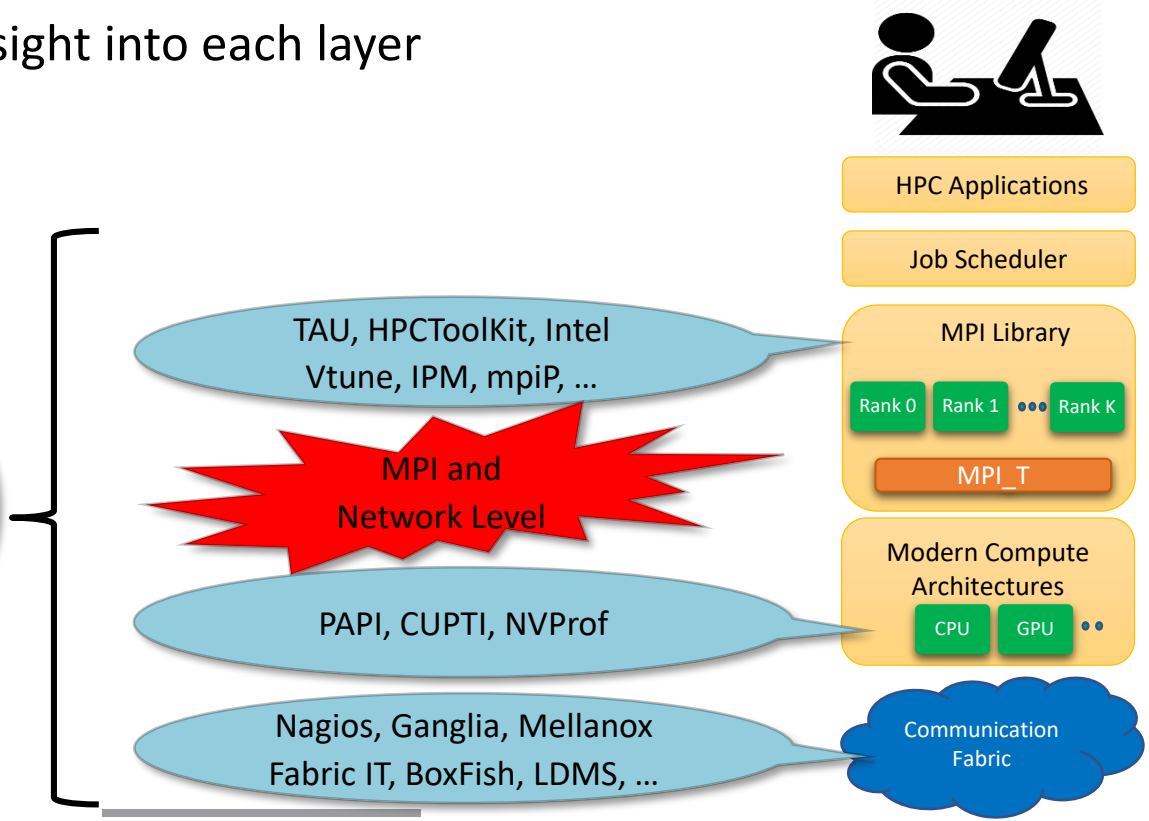
- Understanding the interaction between applications, MPI libraries, and the communication fabric is challenging
  - Find root causes for performance degradation
  - Identify which layer is causing the possible issue
  - Understand the internal interaction and interplay of MPI library components and network level



BROAD CHALLENGE

- There are tools to give insight into each layer
- There is a gap though!

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



OVERVIEW OF THE MVAPICH2 PROJECT

- **High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)**
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 3,050 organizations in 89 countries**
  - **More than 665,000 (> 0.6 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June '19 ranking)
    - 3rd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 8th, 391,680 cores (ABCI) in Japan
    - 16th, 556,104 cores (Oakforest-PACS) in Japan
    - 19th, 367,024 cores (Stampede2) at TACC
    - 31st, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
  - http://mvapich.cse.ohio-state.edu
- **Empowering Top500 systems for over a decade**

Partner in the 5th ranked TACC Frontera System

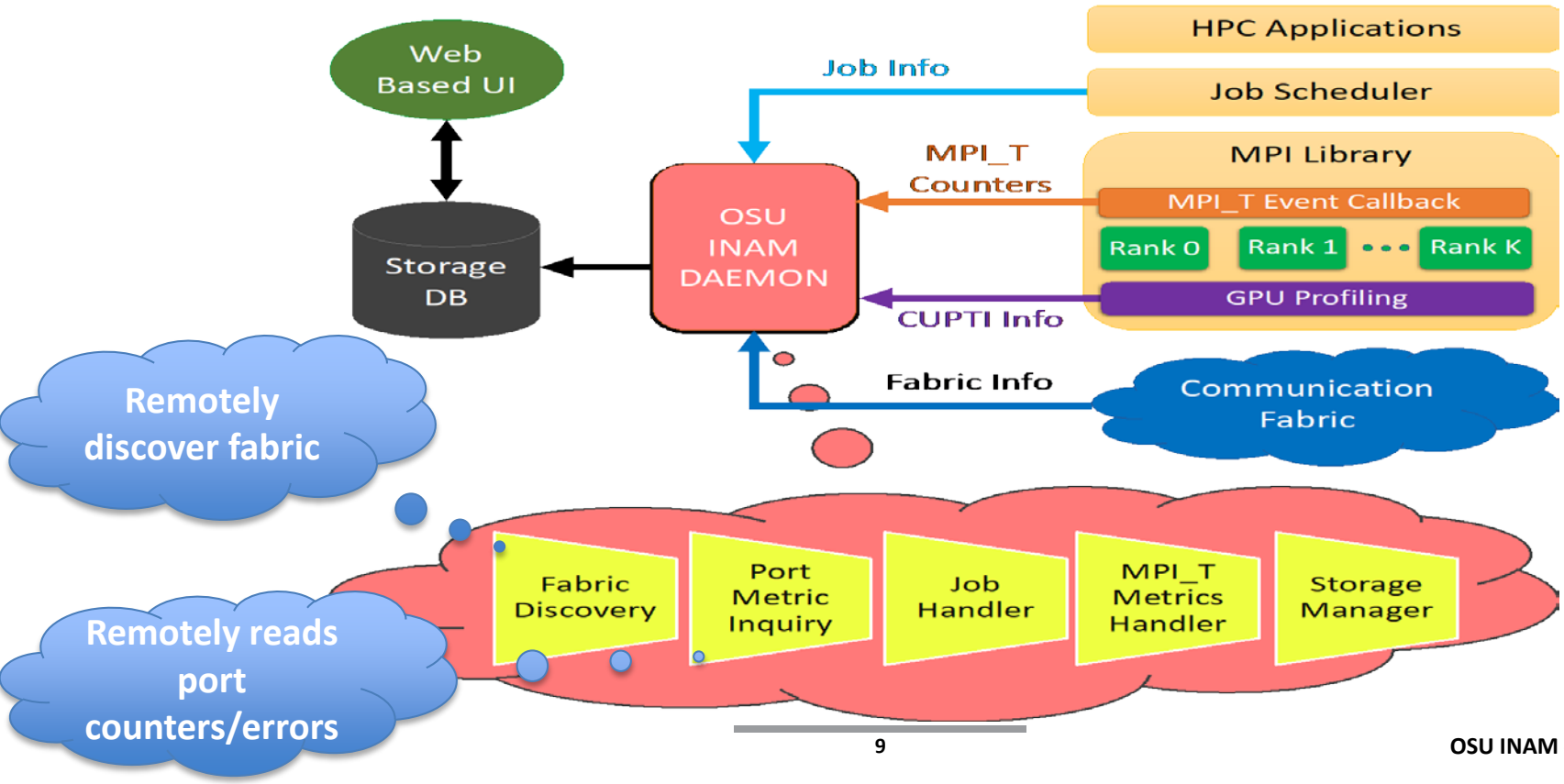
OVERVIEW OF OSU INAM

- **A network monitoring and analysis tool that is capable of analyzing traffic on the InfiniBand network with inputs from the MPI runtime**
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
- **Monitors IB clusters in real time by querying various subnet management entities and gathering input from the MPI runtimes**
- **Capability to analyze and profile node-level, job-level and process-level activities for MPI communication**
  - Point-to-Point, Collectives and RMA
- **Ability to filter data based on type of counters using "drop down" list**
- **Remotely monitor various metrics of MPI processes at user specified granularity**
- **"Job Page" to display jobs in ascending/descending order of various performance metrics in conjunction with MVAPICH2-X**
- **Visualize the data transfer happening in a "live" or "historical" fashion for entire network, job or set of nodes**
- **Sub-second port query and fabric discovery in less than 10 mins for ~2,000 nodes**
- **OSU INAM 0.9.5 released on 12/18/2019**
  - Support for PBS job scheduler
  - Support to display node-/job-level CPU, Virtual Memory, and Communication Buffer utilization information for historical jobs
  - Support to handle multiple job schedulers on the same fabric
  - Support for adding user-defined labels for switches to allow better readability and usability
  - Support authentication for accessing the OSU INAM webpage
  - Optimized webpage rendering and database fetch/purge capabilities
  - Support to view connection information at port level granularity for each switch
  - Support to search switches with name and lid in historical switches page
  - Support to view information about Non-MPI jobs in live node page

OUTLINE

- Introduction & Motivation
- Design of OSU INAM
  - Profiler Interface for MPI+CUDA Communication
  - Profiler Interface for MPI Data collection
- Impact of Profiling on Application Performance
- Usage Scenarios
- Conclusions & Future Work

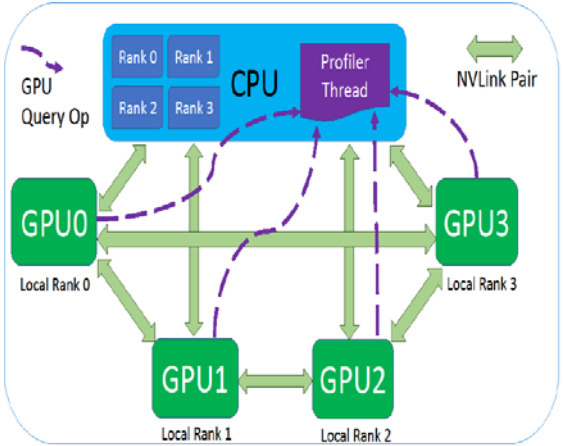
OSU INAM FRAMEWORK



PROFILER INTERFACE FOR MPI+CUDA COMMUNICATION

Low overhead GPU profiler module consist of intra-node topology and metrics inquiry

- Each node aggregates and sends the metrics to the OSU INAM daemon at user-defined interval
- **Startup Phase**
  - Each rank discovers the topology and updates shared region with rank and its device info.
  - Local rank zero will setup and start a profiler thread on CPU to profile all GPUs on node
    - Happens when the list of GPUs that will be used are known
- **Query Phase**
  - Profiler thread periodically profiles all selected GPUs
- **Exit Phase**
  - Once the ranks stop using device, profiler thread will perform one last read and send data then exit.



Information Flow for Intra-node GPU Profiling

PROFILER INTERFACE FOR MPI DATA COLLECTION

Extending MPI_T Performance Variable (PVAR)

- CPU utilization of each process; Memory utilization of each process; Inter-node and intra-node communication buffer utilization; Intra-node, Inter-node and total bytes sent/received and, Total bytes sent for RMA operations
- For each collective and point-to-point operation every rank
  - Stores the total bytes sent to and received from every other rank
  - An array of start and end time-stamps
  - Selected algorithm for the communication
  - The number of times a particular algorithm/function was called
- PVAR information is only sent if the aggregated bytes sent for a particular MPI operation exceeds a user-specified threshold

DATABASE SCHEMA & CORRELATION OF MPI_T AND GPU METRICS

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

Proposed Database Schema

Selected fields in the Database are used to correlate MPI_T and GPU metrics

The GPU metrics will be correlated to MPI_T information at Web UI

OUTLINE

- Introduction & Motivation
- Design of OSU INAM
- Impact of Profiling on Application Performance
- Usage Scenarios
- Conclusions & Future Work

OVERHEAD ANALYSIS

- Overhead of GPU Profiling – reading metrics 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 PVAR Collection – per MPI rank

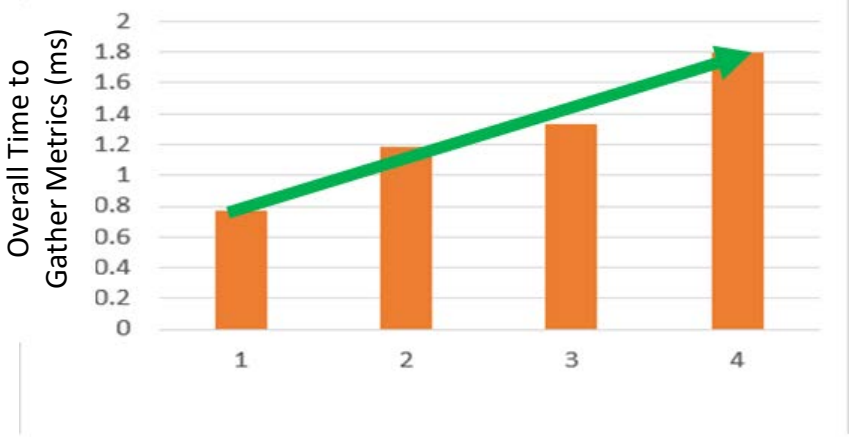
OVERHEAD OF COLLECTING PVAR DATA AT NANOSECOND GRANULARITY

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

Very Low Overhead for Query Phase

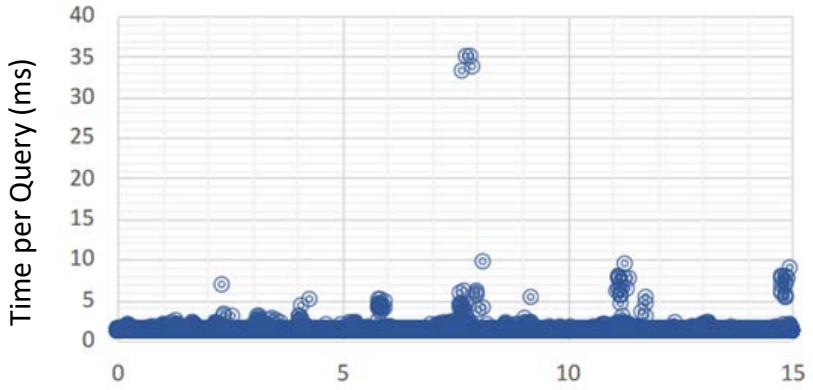
Designing a Profiling and Visualization Tool for Scalable and In-Depth Analysis of High-Performance GPU Clusters, P. Kousha , B. Ramesh , K. Kandadi Suresh , C. Chu , A. Jain , N. Sarkauskas , and DK Panda HiPC'19, Dec 2019

PROFILING VARIATION AND SCALABILITY



Overall Time for Gathering GPU Metrics

- Scales linearly
- Time proportional to number of GPUs
- Metrics are gathered per node



Time/Query for Gathering GPU Metrics

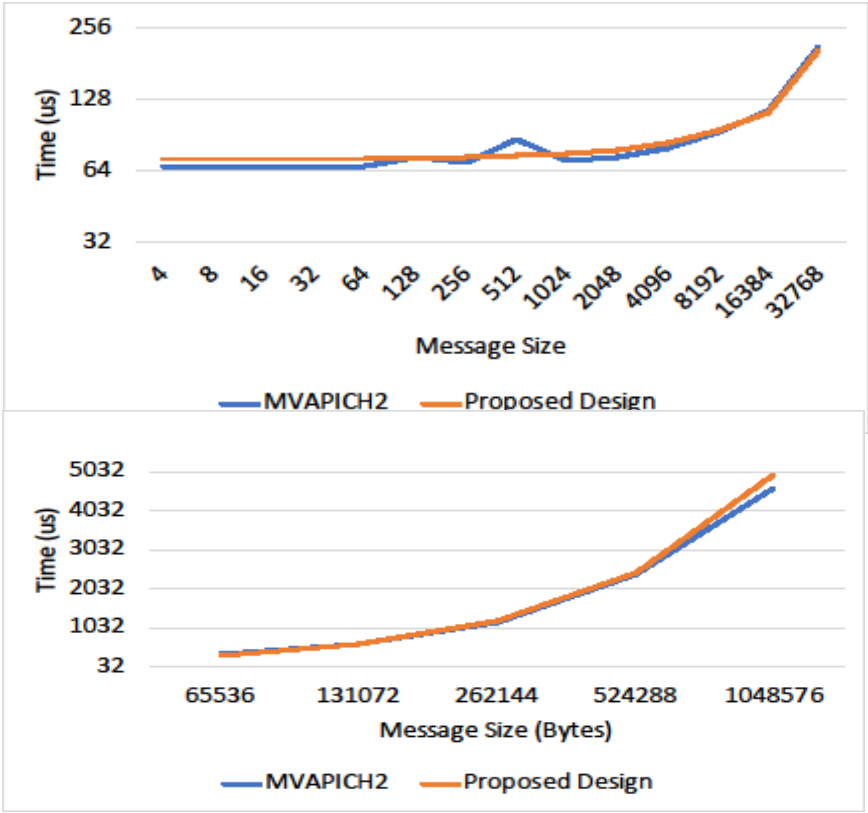
- Average time per query is 2ms
- Timing is stable across thousands of queries

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 , and DK Panda HiPC'19, Dec 2019

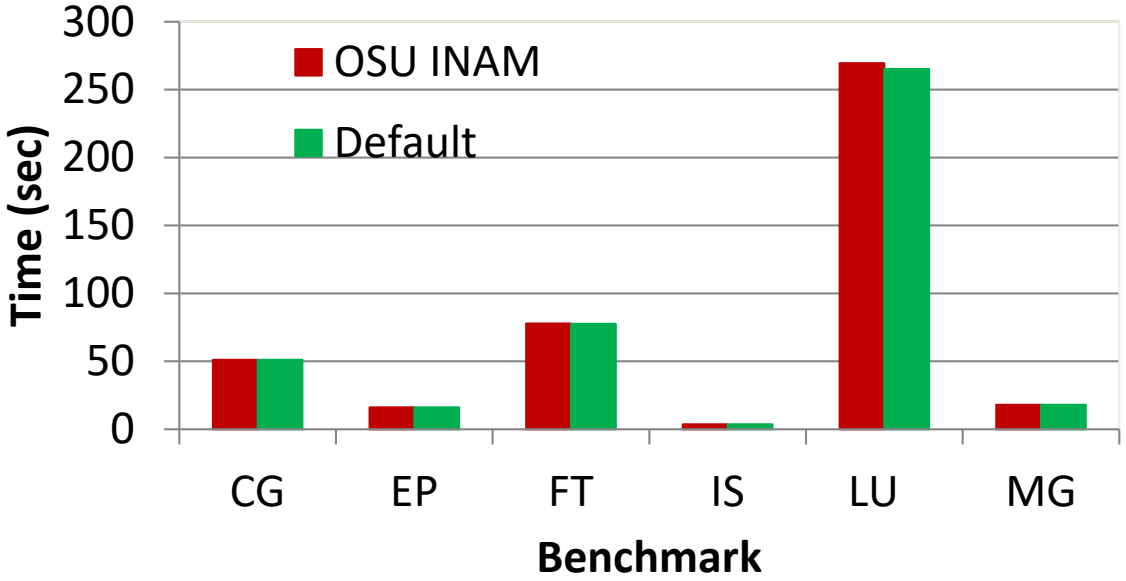
OVERHEAD OF GPU PROFILING ON END-TO-END PERFORMANCE

- Conducted OMB MPI_Allreduce for different message sizes

- ~5% degradation for message sizes between 4 – 4KB
  - Degradation reduces as message size increases



IMPACT OF PROFILING ON PERFORMANCE OF NAS PARALLEL BENCHMARKS

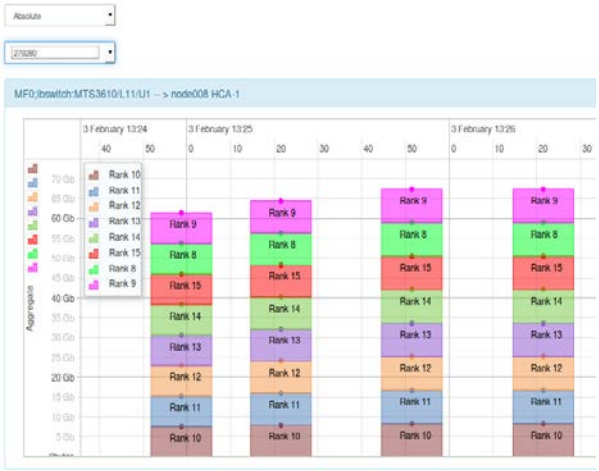


- Performance of NAS parallel benchmarks at 512 processes
- Little to no impact on the performance due to the addition of the data collection and reporting

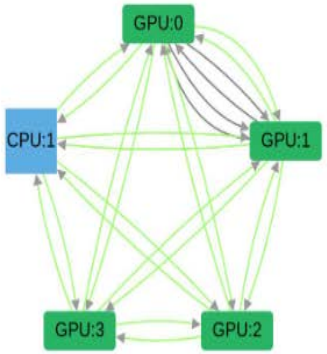
OUTLINE

- Introduction & Motivation
- Design of OSU INAM
- Impact of Profiling on Application Performance
- Usage Scenarios
- Conclusions & Future Work

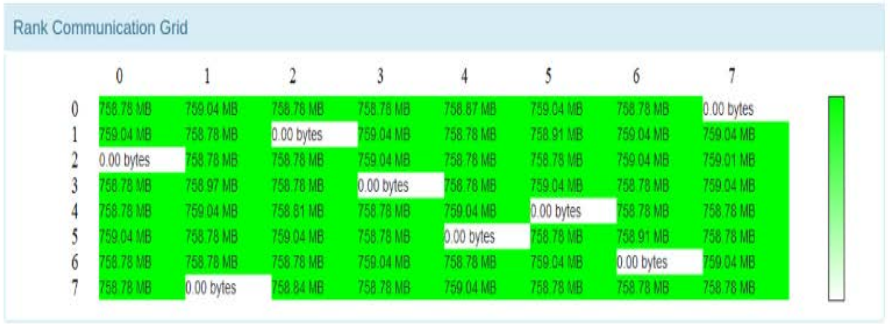
USAGE CASES



Live MPI level communication for each rank on a link



Intra-node Topology showing physical links between CPUs and GPUs



Rank level communication grid for an MPI_Allreduce Operation

Each element (i,j) in the grid represents amount data transferred from rank i to rank j

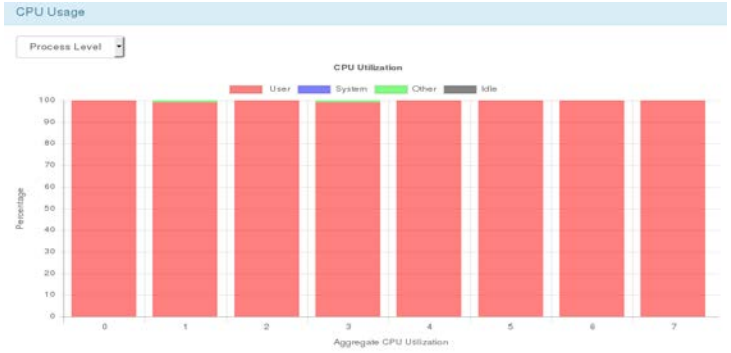
USAGE CASES (COND.)

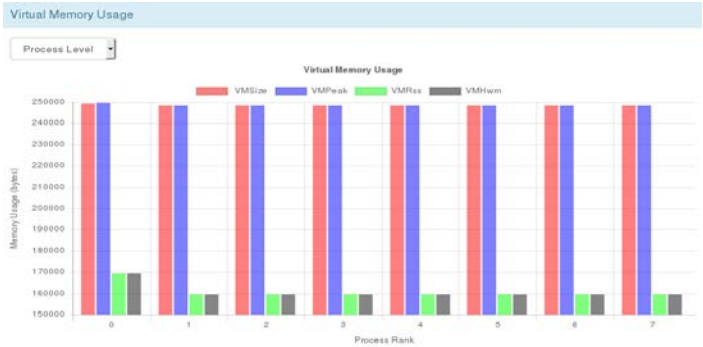
Monitoring Jobs Based on Various Metrics

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

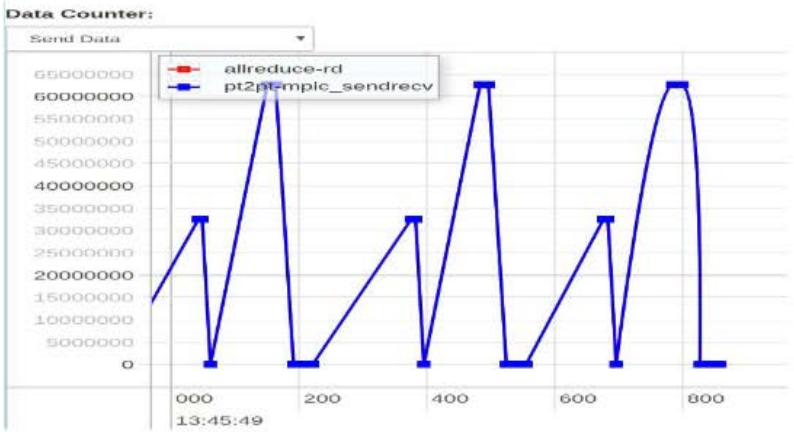
Showing 1 to 5 of 5 rows

Profiling and Reporting Performance Metrics at Different Granularities



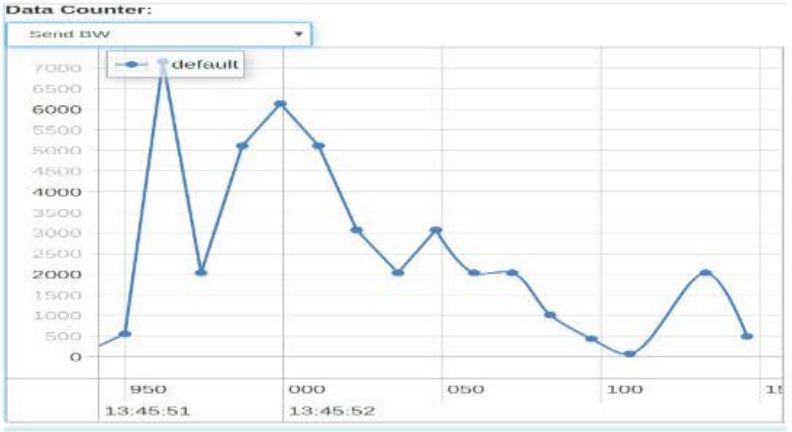


USAGE CASES (COND.)



Proposed Chart Displaying MPI_T PVAR

X-axis: Current time
Y-axis: Number of bytes sent over the network to the OSU INAM daemon

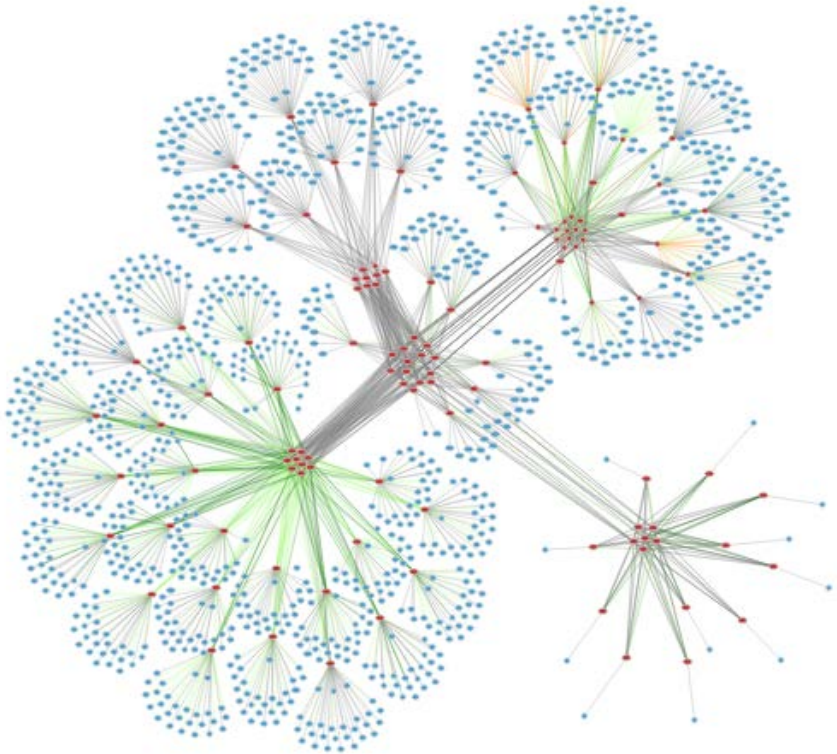


Physical and Logical NVLink Metrics

X-axis: Time
Y-axis: Bandwidth utilization for the link

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 , and DK Panda HiPC'19, Dec 2019

USAGE CASES (COND.)



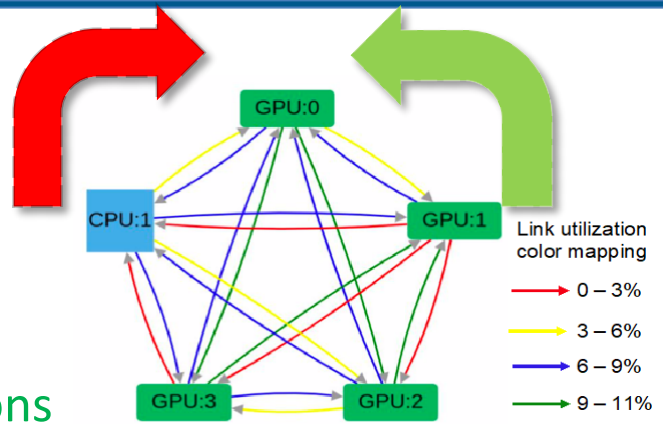
Network View with expanded and hidden modes showing Ohio Supercomputer Center (OSC) with 3 heterogeneous clusters all connected to the same InfiniBand Fabric (114 switches and 1,428 compute nodes connected through 3,402 links)

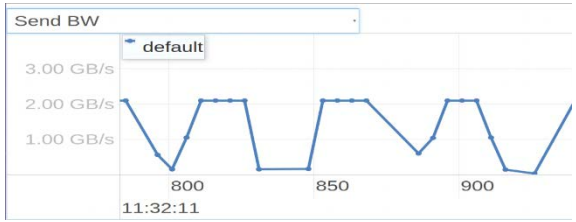
NETWORK AND LIVE JOBS VIEW GENERATION TIMING ON OSC WITH 1K JOBS

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

USAGE SCENARIOS: ALLREDUCE RING-BASED ALGORITHM

- The links in the clockwise direction have relatively lower link utilization compared to the links in the counter-clockwise direction
- Ineffective use of bi-directional bandwidth
- For developer: the ring should use both directions





NVLink utilization GPU0 to GPU1

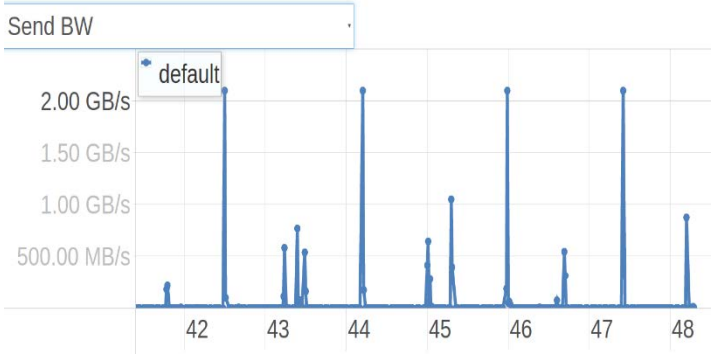
NVLink utilization GPU0 to GPU1

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 , and DK Panda HiPC'19, Dec 2019

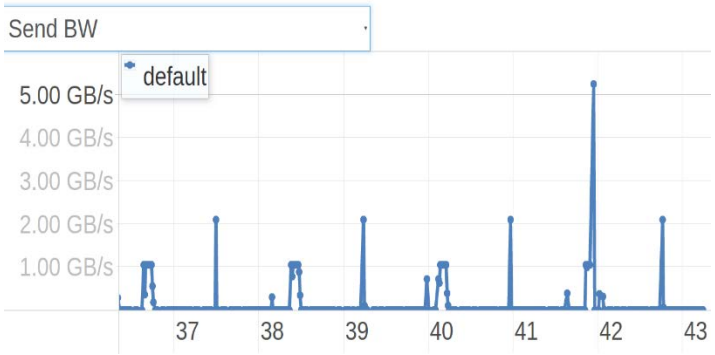
USAGE SCENARIO: TENSORFLOW

Running TensorFlow v1.12 with MVAPICH2 using Horovod with Resnet50 Model

- What is the impact of batch size on GPU communication?
  - Usually users are interested in images/sec
  - Useful to understand lower layer communication efficiency
- The smaller batch size result in lower link utilization and is less communication efficient



NVLink Metrics chart with a batch size of 2



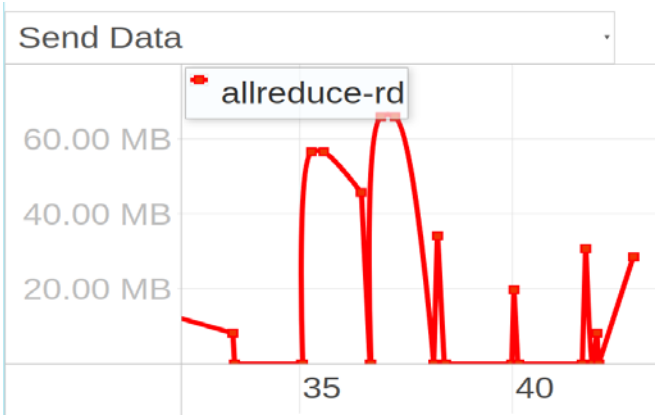
NVLink Metrics chart with a batch size of 32

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 , and DK Panda HiPC'19, Dec 2019

USAGE SCENARIO: TENSORFLOW (CONT.)

- What is the impact of batch size on MPI level?
  - The peak data (message size) transferred between ranks are the same, but showing different patterns between ranks
  - Horovod uses certain message sizes in the Allreduce operations and it depends on the Deep Learning model, batch size, GPU architecture, and other Deep learning parameters

PVAR Metrics chart with a batch size of 2



PVAR Metrics chart with a batch size of 32

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 , and DK Panda HiPC'19, Dec 2019

OUTLINE

- Introduction & Motivation
- Design of OSU INAM
- Impact of Profiling on Application Performance
- Features of OSU INAM & Demo
- Conclusions & Future Work

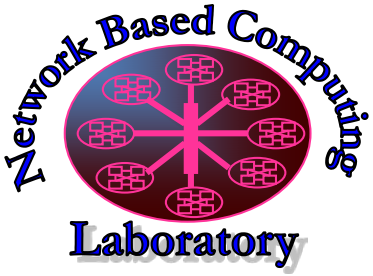
CONCLUSIONS & FUTURE WORK

- Designed OSU INAM capable of analyzing the communication traffic on the InfiniBand network with inputs from the MPI runtime
- Latest version (v0.9.5) available for free download from
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
- OSU INAM has been downloaded more than 500 times directly from the OSU site
- Provides the following major features
  - Analyze and profile network-level activities with many parameters (data and errors) at user specified granularity
  - Capability to analyze and profile node-level, job-level and process-level activities for MPI communication (Point-to-Point, Collectives and RMA)
  - Remotely monitor CPU utilization of MPI processes at user specified granularity
  - Visualize the data transfer happening in a "live" or historical fashion for Entire Network, Particular Job One or multiple Nodes, One or multiple Switches
- Future Work
  - Add support to profile and analyze GPU-based communication
  - Capability to profile various Deep Learning frameworks

THANK YOU!

subramon@cse.ohio-state.edu, kousha.2@osu.edu, sankarapandiandayalaganeshr.1@osu.edu,

panda@cse.ohio-state.edu



Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/

The High-Performance MPI/PGAS Project

http://mvapich.cse.ohio-state.edu/

The High-Performance Deep Learning Project

http://hidl.cse.ohio-state.edu/