OPENFABRICS ALLIANCE

15th ANNUAL WORKSHOP 2019

Visualize and Analyze your Network Activities using OSU INAM

Hari Subramoni

The Ohio State University

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

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

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

OUTLINE

MOTIVATION

- IB clusters and the MPI-based applications complex
- Challenging to identify interaction between and impact of underlying IB network on performance of HPC application
- Such knowledge critical to maximize efficiency and performance of HPC applications
- Rely on a plethora of MPI level and IB level tools to analyze and understand an HPC system to answer questions like
  - *Why is my application running slower than usual now?*

LIMITATIONS OF EXISTING IB FABRIC MONITORING TOOLS

- Several tools exists to analyze and inspect the IB fabric
  - e.g.: Nagios, Ganglia, Mellanox Fabric IT, INAM, BoxFish
- Lack of interaction with & knowledge about MPI library
  - Cannot classify traffic based on MPI primitives
    - e.g.: Point-to-point, Collective, RMA
  - Cannot correlate of network level and MPI level behavior
- Lack of interaction with the job scheduler
  - Cannot classify network traffic as belonging to a particular job
  - Cannot pin point source of conflict at finer granularity

LIMITATIONS OF EXISTING MPI PROFILING TOOLS

- Several tools exists that allow to profile MPI library
  - TAU, HPCToolkit, Intel Vtune, IPM, mpiP
- Lack of interaction with & knowledge about IB fabric
  - Cannot correlate network level and MPI level behavior
- Unable to provide deep insights into MPI library
  - Recently proposed MPI_T interface enables deep introspection
  - e.g.: MPIAdvisor – No knowledge about the underlying IB fabric

BROAD CHALLENGE

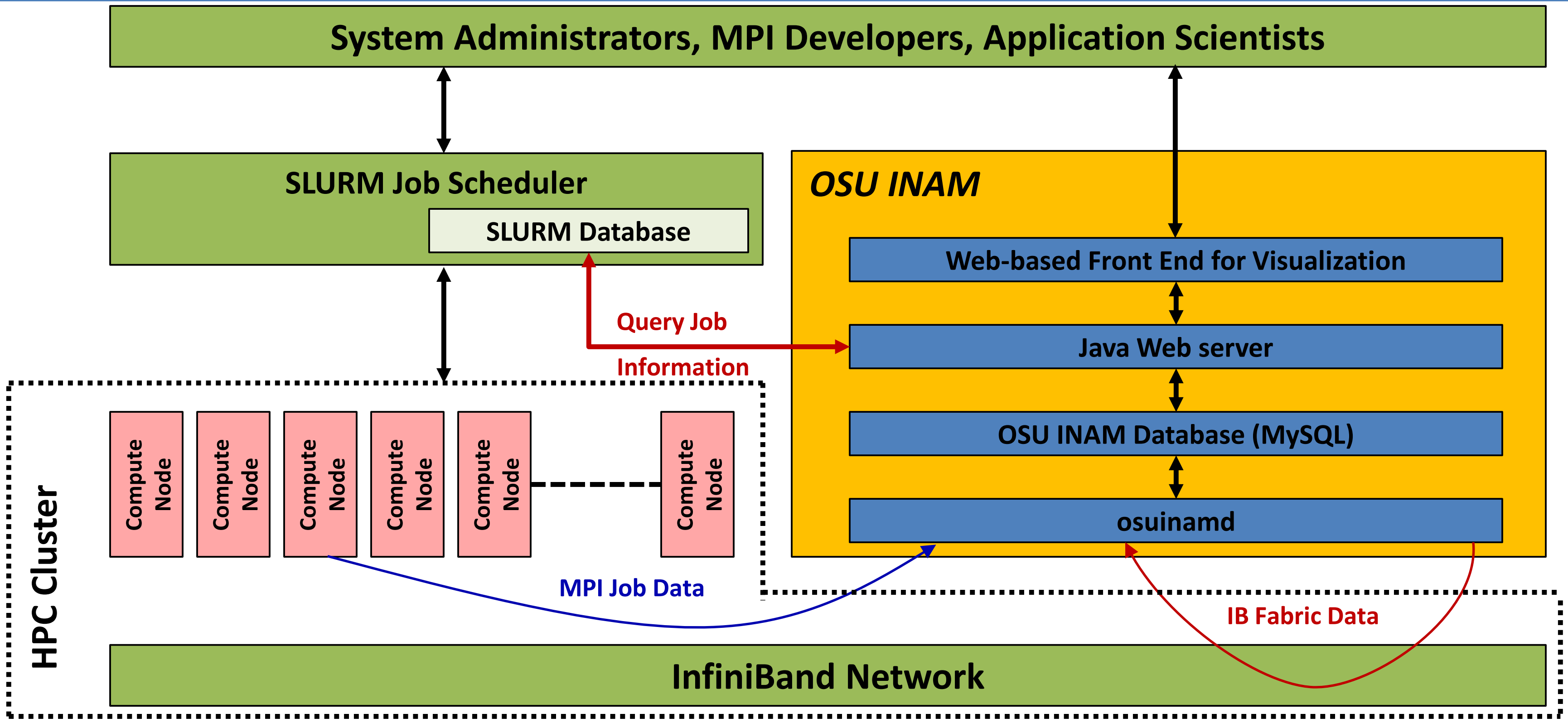
How can we design a tool that enables in-depth understanding of the communication traffic on the InfiniBand network through tight integration with the MPI runtime?

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
- OSU INAM v0.9.4 released on 11/10/2018
  - Enhanced performance for fabric discovery using optimized OpenMP-based multi-threaded designs
  - Ability to gather InfiniBand performance counters at sub-second granularity for very large (>2,000 nodes) clusters
  - Redesign database layout to reduce database size
  - Enhanced fault tolerance for database operations
  - OpenMP-based multi-threaded designs to handle database purge, read, and insert operations simultaneously
  - Improved database purging time by using bulk deletes
  - Tune database timeouts to handle very long database operation
  - Improved debugging support by introducing several debugging levels

OUTLINE

OSU INAM FRAMEWORK



MPI DATA COLLECTION THREAD

- **Collect data specific to each MPI process and pushes it to OSU INAM Database**
  - Allows analysis and visualization job/node/process level granularities
- **Thread is a listener – accepts data from remote MPI processes**
  - Avoid bottlenecks that arise where thread actively polls each MPI process
- **OSU INAM communication requirements**
  -
- **IB based communication to achieve high performance and low latency**
  - Uses interrupt driven mode in IB
    - Reduce CPU utilization by eliminating the need to continually poll
- **Design choices for IB transport protocol**
  - IB supports several transport protocols – RC, XRC, DC, UD
  - UD / DC transport protocols have significant benefits for scalability and memory footprint
  - UD protocol as the IB transport protocol for the MPI data collection thread

CO-DESIGN OF MPI AND OSU INAM

- **Enhance MPI_T based profiling in MVAPICH2-X**
  - 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
- **MVAPICH2-X collects information via MPI_T and transmits updates to the MPI data collection thread via UD Queue Pairs (QP) at user specified intervals**
  - Default value: 30 seconds
- **Each packet sent has some meta data information used later to retrieve the data from the database**
- **MPI data collection thread dumps the UD QP and Local Identifier (LID) that it is listening on to a file**
- **This location of this file is passed through environment variables to MPI runtime by the system administrator**

FABRIC DISCOVERY THREAD & DATABASE THREAD

- **Fabric Discovery Thread**
  - Responsible for discovering the IB fabric and extracting data from selected components
  - Identify the various IB devices present in the network and their current status and stores in DB
  - Computes network path between each pair of hosts and stores in DB
  - Monitor the network for any changes at a user specified interval
  - Queries performance counters from selected components at user specified intervals
  - Queues up the message in FIFO to the database thread for eventual insertion into the database
- **Database Thread**
  - Responsible for receiving information from the MPI data collection and the FD threads
  - Create the tables in schema that the tool expects
    - Automatically update tables used by earlier versions of tool

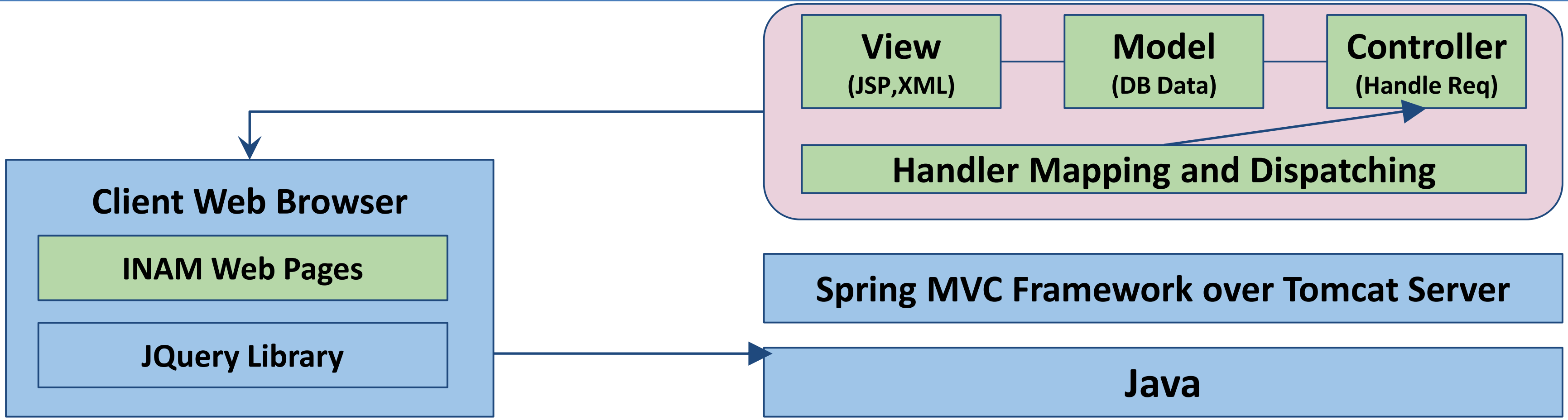
DESIGN OF OSU INAM DATABASE

- **Consists of multiple tables to enable various features of OSU INAM**
  - Tables to hold InfiniBand network infrastructure related data
    - “route”, “links”, “nodes”, “port data counters”, and “port errors”
    - Hold data for links, nodes, ports and routes
  - Tables to keep track of MPI process communication characteristics
    - “process info”, “process comm main”, and “process comm grid”
- **Allows OSU INAM to**
  - Analyze and profile node-level, job-level and process-level activities for MPI
  - Profile and report parameters/counters of MPI processes at the node-, job- and process-level
  - Visualize the communication map at process-level and node-level granularities
  - Analyzing and classifying InfiniBand network traffic flows in a physical link

DESIGN OF JAVA-BASED WEB SERVER

- Queries OSU INAM and SLURM databases to obtain MPI, Network and Job specific information
  - Users can modify frequency of query
- Validates and correlates results of different queries and presents data to the user in an unified fashion
- Based on the Spring MVC (Model, View and Controller) architecture
- Client side uses light-weight JQuery library to send HTTP requests through AJAX
- OSU INAM can send data to and retrieve responses from the server asynchronously
  - Dramatically improves user experience by hiding data processing / page rendering in the background

DESIGN OF WEB-BASED FRONT-END VISUALIZATION



1. HTTP request by users action sent to server side by Web browser / JQuery library with AJAX
2. Tomcat server receives the request, passes it to Spring framework
3. Spring framework dispatches request to the corresponding controller
4. Selected controller queries the model for some information in database
5. After processing, the Spring framework receives response to build the view through JSP, XML, etc
6. HTTP response will be sent back to the browser at the client side and the Web page will get updated

OUTLINE

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

EXPERIMENTAL SETUP

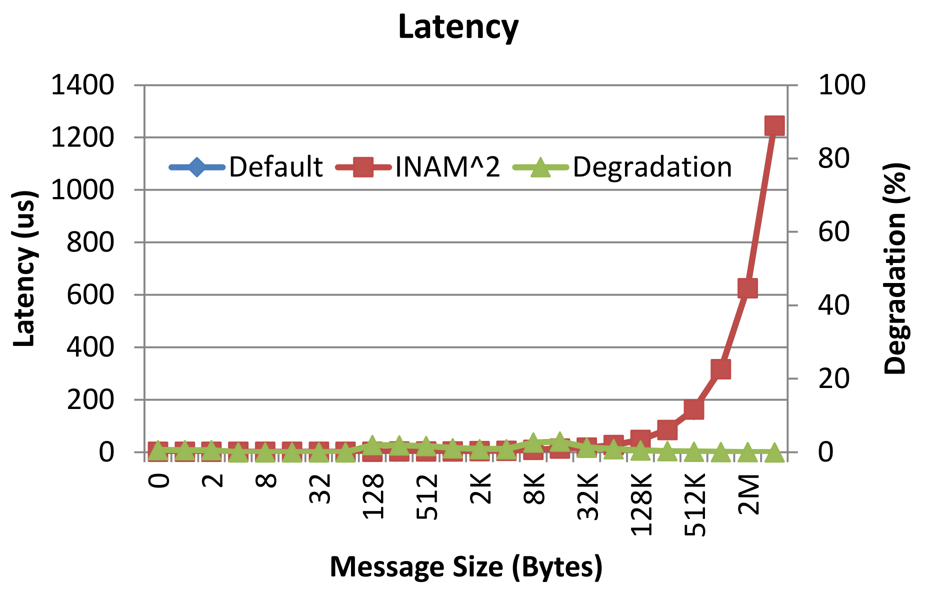
- Each node of our 184 node testbed has eight Intel Xeon cores running at 2.53 Ghz with 12 MB L3 cache; 12 GB of memory and Gen2 PCI-Express bus
- Equipped with MT26428 QDR ConnectX-2 HCAs
- Interconnected using Mellanox MTS3610 QDR switch, with 11 leafs, each having 16 ports.
- The operating system used is Red Hat Enterprise Linux Server release 6.5 (Santiago), with the 2.6.32-431.el6.x86 64 kernel version
- Mellanox OFED version 2.2-1.0.1 is used on all machines.

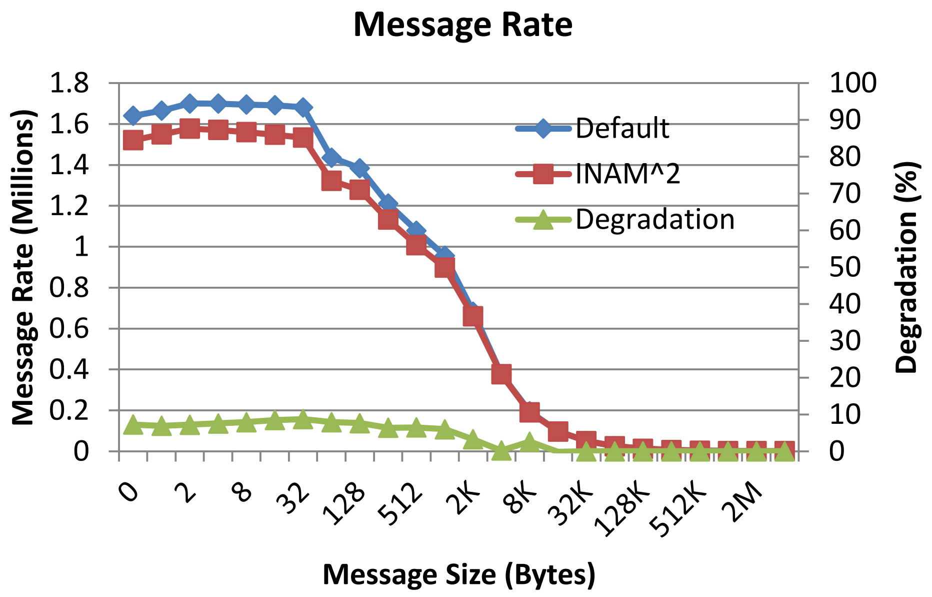
OVERVIEW OF THE MVAPICH2 PROJECT

- **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 2,950 organizations in 86 countries**
  - **More than 527,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '18 ranking)
    - 3rd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 14th, 556,104 cores (Oakforest-PACS) in Japan
    - 17th, 367,024 cores (Stampede2) at TACC
    - 27th, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - http://mvapich.cse.ohio-state.edu
- **Empowering Top500 systems for over a decade**

Partner in the upcoming TACC Frontera System

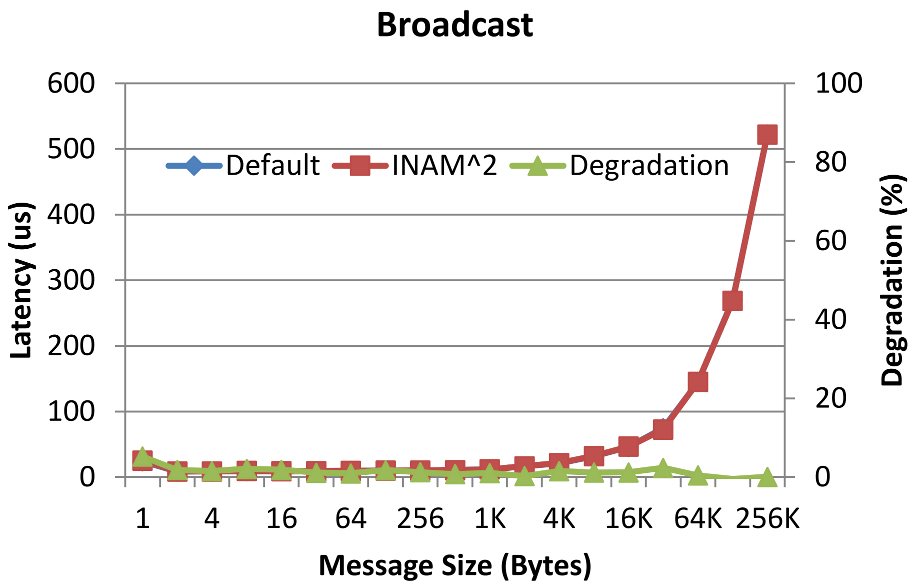
IMPACT OF PROFILING ON PERFORMANCE OF POINT-TO-POINT OPERATIONS

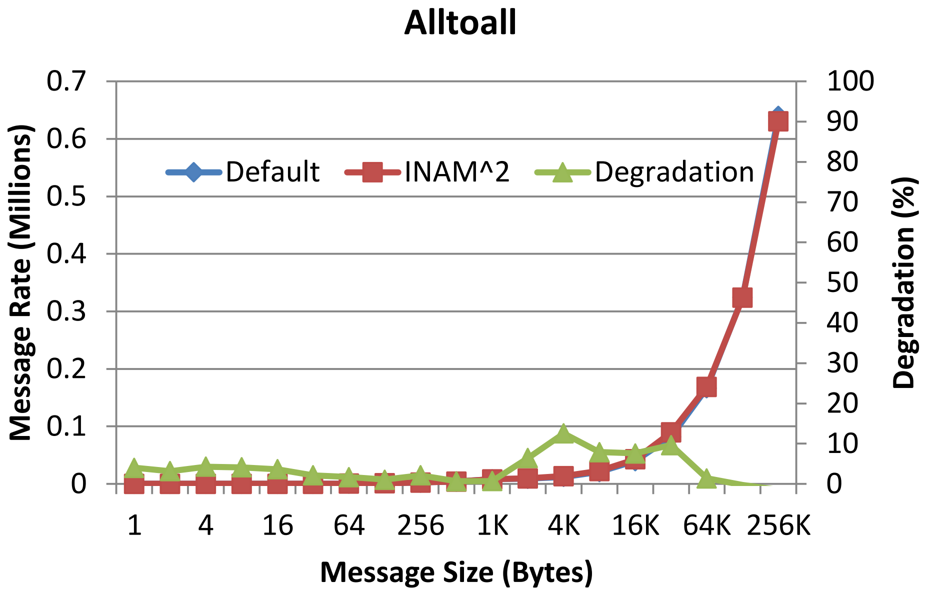




- Data collection adds very less than degradation when compared to the native performance

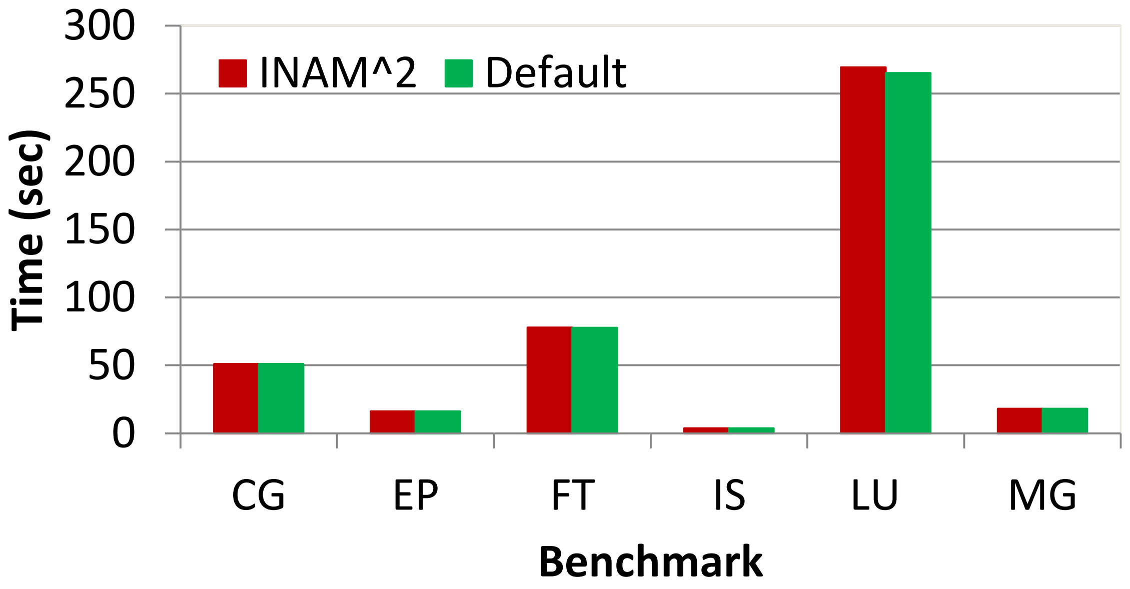
IMPACT OF PROFILING ON PERFORMANCE OF COLLECTIVES





- Performance of Broadcast and Alltoall at 512 processes
- Data collection adds very less degradation when compared to the native performance

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
- Features of OSU INAM & Demo
- Conclusions & Future Work

DISCUSSION ON FEATURES OF OSU INAM

- Analyzing and Understanding Inter-node Communication Buffer Allocation and Use
- Identifying and Analyzing Sources of Link Congestion
- Monitoring Jobs Based on Various Metrics
- Capability to Profile and Report Several Metrics of MPI Processes at Different Granularities

OUTLINE

CONCLUSIONS & FUTURE WORK

- Designed OSU INAM capable of analyzing the communication traffic on the InfiniBand network with inputs from the MPI runtime
- Latest version (v0.9.4) 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 PGAS programming languages

THANK YOU!

subramon@cse.ohio-state.edu, panda@cse.ohio-state.edu

Network Based Computing Laboratory

Network-Based Computing Laboratory

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

MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

The High-Performance MPI/PGAS Project

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

HiDL

High-Performance Deep Learning

The High-Performance Deep Learning Project

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