Visual Analysis Platform for Operation Data of Super Computer Clusters

Pouya Kousha
Department of Computer Science and Engineering
The Ohio State University
kousha.2@osu.edu

Alex Shearer
Department of Computer Science and Engineering
The Ohio State University
shearer.145@osu.edu

Dantong Xue
Department of Computer Science and Engineering
The Ohio State University
xue.425@osu.edu

***Abstract*—Designing a monitoring and profiling visualization tool for network analysis is a challenging task. Newer set of challenges come out as High Performance Computing (HPC) systems become more complex and users need better visualization capabilities for detailed network communication traffic. A breakdown of traffic is required to understand the application behavior in terms of communication while not loosing general view of network traffic. Such visualization capability is currently missing or not studied well. We use the data gathered by OSU INAM - a network monitoring tool that gathers, stores profiling data for Infiniband network. By having the data we investigate visualization techniques to demonstrate the data in a more efficient, easy to interpret and holistic manner. Such visualization designs will help the users and the management team to understand the underlying structure of the network and resource utilization for the nodes in the system.**

***Index Terms*—cluster, visualization, Profiling**

I. Introduction

Recent advances in High Performance Computing (HPC) have provided the fast processing engine for HPC applications. HPC systems must be highly optimized to meet the increasing needs of modern HPC applications. As a solution to deliver the required computing power for the applications, large-scale HPC cluster with Mellanox InfiniBand and High Speed Ethernet are being deployed and used by various HPC users like domain scientists, HPC software developers, and HPC administrators.

With the recent advances in interconnect technology and increasing cost of communication, providing efficient data movement between nodes on the communication fabric is essential to guarantee optimized end-to-end solutions. In such an ecosystem, understanding the interaction between nodes and switches in terms of edge and root switches has become even more challenging. For HPC users, tuning one job on the the node allocation is just the start. When running on a larger system that includes hundreds of high-performance jobs, it becomes challenging to observe the behaviour of jobs with regard to other jobs and the interplay of jobs in terms of network communication on each other. Therefore, detailed and easy-to-interpret insight of network level and the communication is needed for all types of HPC users to identify and alleviate performance bottlenecks.

Thanks to Dr. Jian Chen for her useful comments

OSU InfiniBand Network Analysis and Monitoring tool - *OSU INAM* [1] [2] is a system that monitors the high-performance computing units in a supercomputer cluster. It records information about the utilization of the links, the connections of the node, the errors during operation, etc. Current issue with the system is that it only gives numerical numbers without graphical interpretation of those numbers. The users and managers of the system cannot make analysis easily with current system. There is a need for interactive and straightforward visualization tool to better utilize and arrange the computing resources. The idea of this project is to give a clear visualization how the HPC infrastructure perform and how individual nodes in that network communicate with each other.

A. Project Scope and Considerations

The scope of this project is to use RI2 network profiling data gathered by OSU INAM to do network topology and network link utilization visualizations. RI2 is an in house cluster from The Ohio State University. The data shall not to be used in other research purposes. The port data counters for the RI2 network and the topology of a larger cluster will not be provided due to Intellectual Property issues. The visualizations developed as part of this project cannot be used for other publications, tutorials and workshop research posters and research papers. The target is to merely practice visualization techniques learned in CSE 5544 course and deploy them on a real world problem. If you need to run it with link usage data please contact Pouya Kousha.

II. Background

In this section we provide the necessary background for the paper. We define two types of switches:

- Edge switch: where the switch has links connected to compute nodes. For example, in FigureIV-C1 the four corner switches (blue circles) are edge switches as they are connected to compute nodes (white circles)
- Root switch: where the switch is not connected to the compute nodes and plays an infrastucture role in connecting the other edge switches together. For example, in Figure IV-C1 the two middle switches are root switch.

A. *Interconnect Technology*

High Speed Ethernet, InfiniBand and Omni-Path are the major interconnects used in modern HPC community. InfiniBand is high speed and high bandwidth switch interconnect providing low latency and used in more than 25% of top500 [3] list including #1 *Summit*, #2 *Sierra*, and #5 *Frontera*. The latest InfiniBand HDR adapters offer bidirectional bandwidth of 200Gbps. The data collected for each link are gathered from libibverbs provided by Mellanox [4]. The data shows total number of packets sent/received, total number of bytes sent/received, a classification of number of uni-cast and multi-cast traffic in bytes for sent/received. Typical users cannot get these information since it requires root access.

B. *Understanding The Data*

The data is obtained from storage units of OSU INAM [1]. A SQL dump is performed with useful tables exported as csv files. The data is provided by Pouya Kousha (one of our teammates) under the approval of his supervisor in accordance with Section I-A. For the network topology three types of tables are provided: Links, nodes. The nodes table consist of type of the nodes (compute node or switch), Global Unique IDentifier (GUID), Local IDentifier (LID), node name, and number of ports. The links table include information on source and destination (GUID, port) pair, type of the link depending on being switch¡-¿switch link or node¡-¿switch, link width, and link speed. For the network traffic we have port data counters table. In this table we show the network traffic for every link in the network. Since the information from both endpoints of link are the same, the data only gathers the counters from switch endpoints and leaving the nodes uncollected to optimize the data collection speed. Refer to Section II-A for understanding of collected metrics. The table includes the switch GUID, the port number, transmitted data (xmit_data), received data (rcv_data), number of sent/received packets, uni-cast sent/received data, multi-cast sent/received data, and time the entry was added on. The data for RI2 consist of 3557 entire sweep of the entire switch network of 60 nodes and 189 links. The data larger cluster consist of 1738 nodes with 3579 links with 109 switches. For both of the clusters the switches support up to 36 ports.

III. Challenges and Motivation

In this section we discuss the main challenges and motivation for choosing this project and enlist the potential benefits by having such visualization capabilities for HPC system users including system administrators, domain scientists, and HPC software developers.

Usually when HPC users get node allocations, they do not know where are their nodes located. The locality of node allocation can impact the performance of jobs as there are hundreds if not thousands of job running on in-production clusters. Therefore, the node allocation where all the nodes are connected to a switch endures lower performance jitter than a case where the nodes are spread across the cluster and connected by root and edge switches. Since there are other traffic from other jobs/node allocations are running on root switches then they impact and interplay with each other. Having an understanding to the node allocation can alleviate explaining performance jitters across different node allocations.

A. *Problem Statement*

To better clarify the problem and challenges we ask the following questions and seek to address them by our designs throughout the paper.

1) Can we visualize the details of network topology like connectivity in an interactive manner?
2) How can the network topology be visualized along with link usage to provide a full picture of network and the link usage?
3) How can we present the complicated network traffic of links like uni-cast and multi-cast data along with network topology and link usage to have a holistic visualization?
4) Considering the rendering of data points for link usage is time consuming, Is there an interactive visualization technique to show all of the data without compressing the data points?

B. *Potential Benefits and Contributions*

In this project we seek to address the problem statements mentioned in Section III-A by designing and implementing innovative visualization plot. The contributions of the paper are as follow:

- By having the network topology, users can understand their node allocation topology across the network to have a better expectations of network performance jitter and locality of their nodes
- By providing a simple and fast rendering of traffic on the links, users can visualize the communication pattern happening on links. This is critical to understand the patterns to study the communication behavior of applications
- By having all the data merged into one chart, users can relate, compare and understand the interplay of communication happening at different switches and nodes at the same time. This information increase the productivity of domain scientists who are not familiar with gathering and visualizing these information
- System administrators can identify the congested links very quickly to take actions.

IV. Design

In this Section we discuss our designs and implementation to address the problem statements mentioned in Section III-A. We suggest three designs and discuss how they satisfy the needs for holistic visualization for HPC network traffic. Based on the data we have, we want to provide a visualization platform that can meet the needs of the system operator.

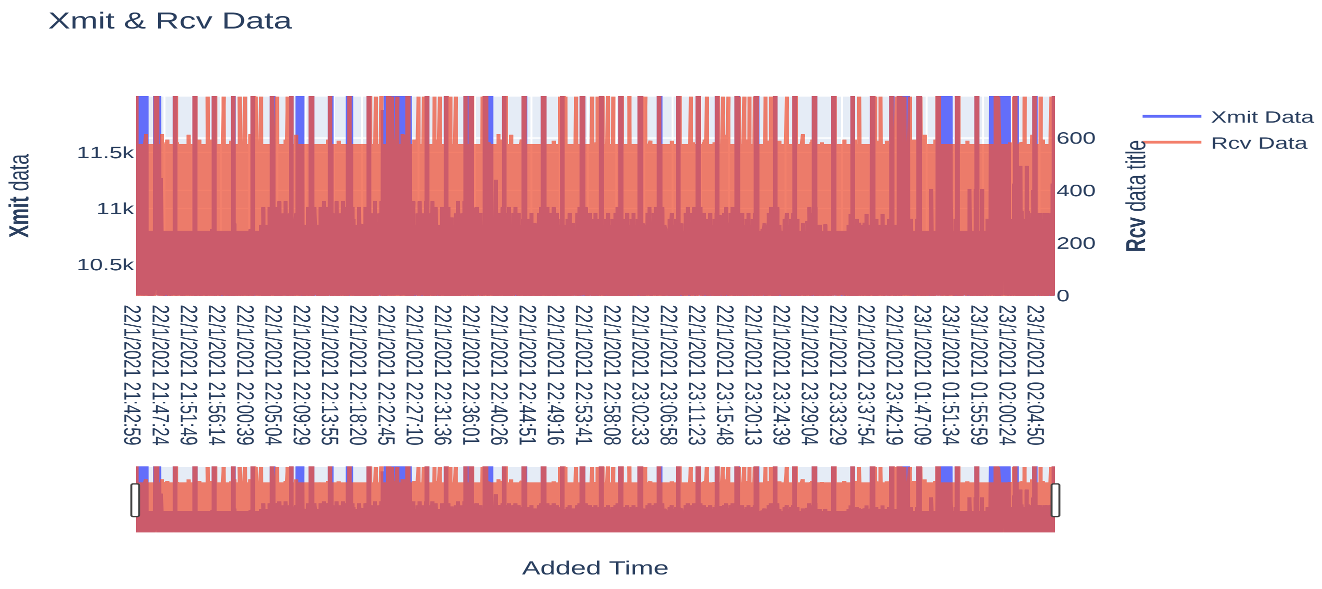

Fig. 1. Detailed design and implementation of time series view of network traffic for a specific link for 3557 data points- The chart is clickable and users can zoom in to view the detailed version.

A. Implementation

This visualization would be implemented using Python 3 with support of visualization packages including *Plotly* and *Matplotlib*. We would communicate using Microsoft Teams and emails. Code was shared and worked on collaboratively over *Google Collab* before it was exported to a standalone web page. We chose the *Plotly* graphical library because of it's flexibility in rendering custom figures, which became extremely useful in drawing the radar and network hybrid graph. Another factor in choosing *Plotly* was our shared experience of Python programming, as well as tools like *Google Collab* that allowed us to work collaboratively on the Python notebook. Our entire stack is $cufflinks > plotly > plotly.js > D3.js$.

B. Link Usage Over Time

To address Problem #4 discussed in Section III-A, in the first graph we want to guarantee providing the capability to visualize the network traffic like xmit_data and Recv_data over the time. There are more than 3,500 time series to show the data points. For the color we choose orange and blue; we avoid selecting red or green to make sure that the user does not think of good or bad data. tFor this problem we need to know which link is user interested and use the GUID and port number as argument to retrieve and visualize the data. Figure 1 shows the congested scatter plot version connected by lines for xmit/recv data. By hoovering over the data the chart in Figure 2 shows the number of packets sent and received for Xmit and Recv trends across timestamps. Figure 2 depicts a zoomed sliding window version of network traffic.

C. Network Topology Visualization

By this design we seek to address Problem #1 and #2 discussed in Section III-A. We use graph view to show the network topology. The white nodes represent the compute nodes while the red/blue circles depending on type of link

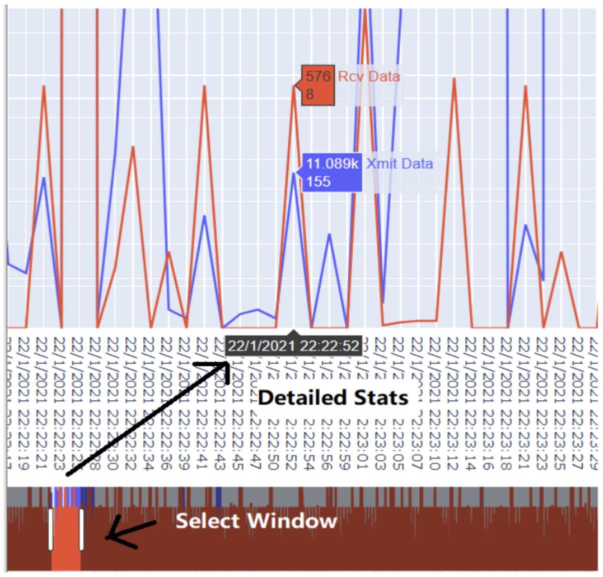

Fig. 2. A zoomed view of link usage over time for specific port an GUID from Figure 1- The chart is clickable and users can select .

view show the switch. We provided two versions to show the topology of the networks. In particular, we started with a simplified version of the network having single link between pairs of nodes and moved to multi link network to show the entirety of the network in case the user is interested to view the root switches links. This gives users flexibility of choosing their desired visualization and not get overwhelmed with extra links data. We used the colors to indicate the best is at 50% link usage and moving near full link capacity is not desired since it cause congestion.

*1) **Single Link Network Topology**:* Below is the design with single link between nodes. On top of providing backbone structure of the network, the colors of the links explicitly serve as an indicator to show the utilization rate of that specific link.We also integrated detailed information about links and nodes in this design. The users can hover the mouse on different elements of the graph to see messages about IDs of links and nodes, the source and destination of specific links, and the detailed utilization number of each link. This design of detailed artifacts supporting the overall topology view provides better user experience with high level of interaction.

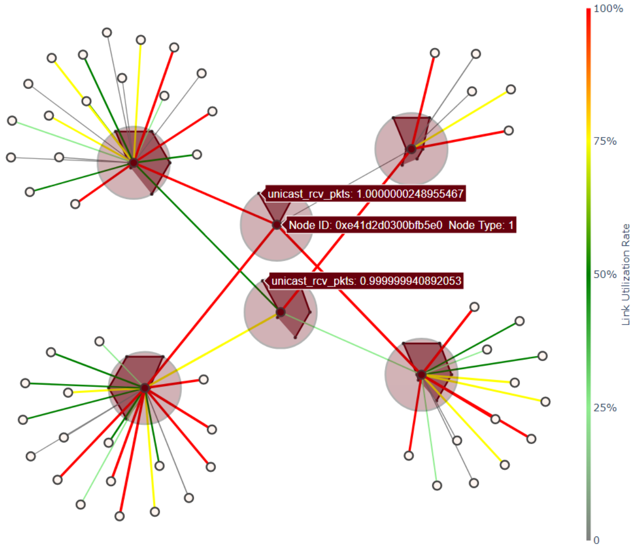
unicast_rcv_pkts: 1.0000000248955467
Node ID: 0xe41d2d0300bfb5e0 Node Type: 1
unicast_rcv_pkts: 0.999999940892053
100%
75%
50%
25%
0
Link Utilization Rate

Fig. 3. Single link network topology view of RI2 cluster with colored links based on their usage on the right side of the figure. Green indicated good usage and red indicates link congestion.

2) ***Multi Link Network Topology:*** We designed this visualization to allow monitoring the link usage where we have multiple links between two switches. This is used for understanding the link usage between root switches. We provided a graph intentionally address this need to visualize the link usage based on multi-link topology. We preserved the coloring strategy and the hovering detailing. We also decided to curve the links to help differentiate different links. Our implementation tool *Plotly* does not have built-in curve fitting support as it is one of the limitations of *Plotly*. We implemented our own curve interpolation algorithm. Let two ends of the edge be $A(x_1, y_1), B(x_2, y_2)$, suppose we have n links between two nodes, for all $k = 1, .., n$, we draw the spline passes through $[A, (x_1 + \frac{k(x_2 - x_1}{n}, y_2 + \frac{k(y_1 - y_2)}{n}), B]$. Our clients can easily identify the root nodes connecting to the switches while still be able to comprehend link usage statistics on each distinguishing link.

D. RadarView Chart

By this design we seek to address Problem #3 discussed in Section III-A to present the complicated network traffic of links like uni-cast and multi-cast data along with network topology and link usage to present a holistic visualization. For the network switches, there are additional attributes that we want to display for breakdown of network traffic in terms of (multi-cast,uni-cast, total traffic) data bytes sent/recv. This gives us 6 different axis to insert into a circular parallel axis. Figure IV-D depicts such visualization capability and example.

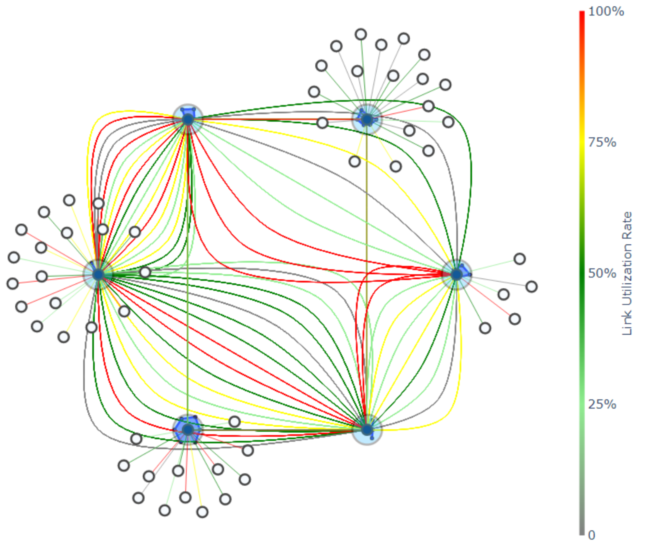

Fig. 4. topology view of RI2 cluster with colored links based on their usage on the right side of the figure. Green indicated good usage and red indicates link congestion.

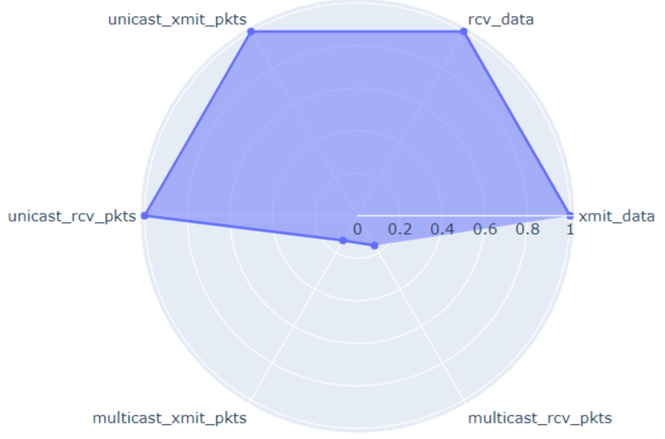

Fig. 5. Explanation and mapping of network metrics to parallel axis in a circular view used for overlaying over the switches.

These attributes include the total transmitted and received data, the the percentage of packets transmitted and received that are uni-cast versus multi-cast. To include this additional traffic information all within the same visualization, we opted to overlay radar charts on top of each switch node on the graph. By including this traffic information this way, viewers can easily understand the traffic going through one of the switches just at a glance, and compare switches to each other by comparing the shape of the radars. To get the exact number or percentage of a specific attribute, the user can hover over part of the radar chart to get the exact figure. This capability enables users to compare the different metrics of network , in terms of multi-cast vs uni-cast data, and compare them to total

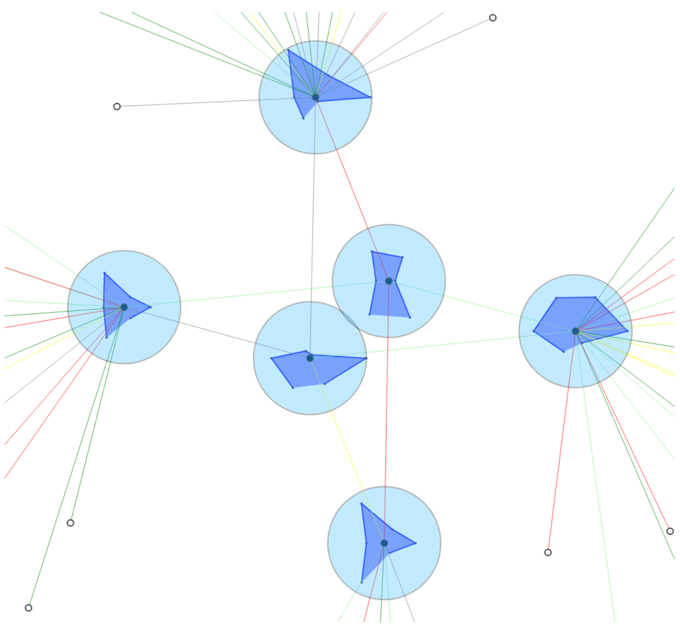

Fig. 6. Easy to interpret visualization and comparison of network switches based on their traffic breakdown. RadarView of switches are shown in blue color so that it does not interfere with other colors. Insight #1: We observe that the bottom switch has 2x received bytes compared to transmitted data and the percentage of uni-cast and multi-cast received data is almost equal. Insight #2: we observe that the most right switch has more balanced sent and received bytes.

bytes sent/recv on the link without having to go to the time-series graph. Moreover, RadarView chart enables easy visual comparison of switches.

V. Scalability

In this project, we also explore the scalability aspects of our network visualization. In particular, we move to a larger dataset of nodes and links, with 30x nodes and 20x links, consisting of 1738 nodes with 3579 links and 109 switches. We noticed that the challenge is to preserve a readable structure of the network while providing detailed information for specific links and nodes.

In single link setting, the network can perform reasonably. The scaling is automatic as the code will fetch the number of nodes and links from data read in. The radar plot provides valuable information about the switches even in massive network setting. After zooming in, we can still get very clear information about nodes and edges in our network, especially the information relevant to a particular switch node. We believe our clients can easily get the information they need after they zoom in to a particular node.

VI. Discussion of Attempts and Limitations

One of the downsides of using the *Plotly* library ended up being we were not able to implement the interactivity between figures that we would have originally wanted. This was originally believed to be an issue with using *Google Colab* to host the shared code during development, but after

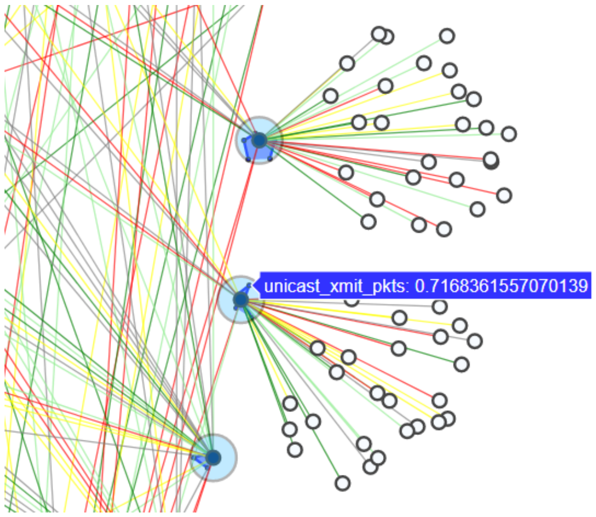

Fig. 7. Single link topology network for a large cluster Zoomed in

further research the issue came from the library itself. While *Plotly* allows for user interactions on a single graph, due to the library's internal implementation, it is impossible to have user interaction when creating subplots. Due to this unforeseen technical limitation, the created visualizations were not able to interact with each other. Multiple methods were attempted to overcome this, including changing running environment settings, as well as creating a *Plotly Dash* application, but none were able to fix the issue of interaction between figures. With *Google Colab* and *Plotly*, we also cannot achieve streaming data to provide live visualization framework. The streaming feature is a premium feature for Dash Enterprise. With this foundation, we can easily move our current designs to online real-time platform if there is a future need.

Another limitation that we were facing was that the data does not have categorical variables to use different coloring schemes for showing the nodes or time series charts. For Figure 1 and 2, we tried using the bubble chart for the size of data points, however since the range of traffic (from 1 Byte to 100 GB per second) is very different and and does not follow a Gaussian pattern, the data points in Figure 1 and 2 were visualizing too small or too large. Considering we did not have categorical variables, coloring the data points was not effective either.

VII. Conclusion

The visualization we created addresses our initial problem statements as it successfully shows both the network topology as well as all of the relevant traffic information in a single visualization. This can visualization serve as an interactive management and monitoring platform that can help the users and supervisors of the HPC facilities to better understand how their systems perform. Furthermore, this project can be expanded upon the additional features of streaming live data from the network as well as allowing further user interactivity to allow even better network monitoring. But even without these features the constructed visual works to visualize the complex structure within these super computer clusters.

References

[1] "OSU INAM," 2020, Accessed: April 16, 2021. [Online]. Available: http://mvapich.cse.ohio-state.edu/tools/osu-inam/

[2] P. Kousha, S. D. Kamal Raj, H. Subramoni, D. Panda, H. Na, T. Dockendorf, K. Tomko , "Accelerated real-time network monitoring and profiling at scale using osu inam," July 2020.

[3] "TOP 500 Supercomputer Sites," http://www.top500.org.

[4] "Mellanox CS7500 Switch Series," Accessed: April 16, 2021. [Online]. Available: http://www.mellanox.com/page/products$_dyn?product_family = 191mtag = cs7500$