We propose extending common performance measurement and visualization tools to identify network bottlenecks within MPI collectives. By creating additional trace points in the Peruse utility of Open MPI, we track low-level InfiniBand (IB) communication events and then visualize the communication profile in Boxfish for a more comprehensive analysis. The proposed tool-chain is non-intrusive and incurs less than 0.1% runtime overhead with the NAS Parallel FT benchmark.

### Our Goals

1. Expose MPI’s internal performance in a portable manner
2. Develop a lightweight profiler to capture low-level metrics
3. Enable the visual, hardware-centric analysis of performance metrics

### Design and Implementation

1. **Exposing Low-level Performance in Open MPI**
   - **Background**: High performance computing (HPC) systems are growing in physical size and complexity.
   - **Motivations**: MPI’s hardware abstraction hinders exposing performance from within the MPI library and from within the network layer.
   - **Experiment 1**: TSUBAME-KFC
     - Peruse defines an interface for exposing the internal performance of MPI libraries
     - User-supplied callback functions can be attached to internal MPI events
   - **Experiment 2**: NAS Parallel FT Benchmark
     - Uses the Peruse interface

2. **Non-intrusively Collect Low-level Metrics**
   - **Experiment 1**: TSUBAME-KFC
     - Developed a non-intrusive profiler named ibprof, which:
       - Uses the PERUSE_OPENIB_SEND event to aggregate messages sent from each local IB interface to each remote interface
       - Designs the network communication profile to Open Trace Format (OTF) files
       - Supports the profiling of all communication, specific collective(s), and specific code section(s)
   - **Experiment 2**: NAS Parallel FT Benchmark
     - Based on the Torus 3D module that is bundled with Boxfish
     - Can natively visualize any 2D network topology and can extend to support all topologies
     - Uses biocolored network links to accurately represent bidirectional traffic flow

### Analysis Process

- **ibdijagnet files (IB config. data)**
- **ibprof output files**
- **Post-processor**
  - Create connected network graph
  - Position nodes and links
  - Map performance data to network elements (nodes and links)

### Results

**Profiling Overhead**
- Experiment environment: TSUBAME-KFC
  - Used 32 nodes, two IB FDR switches
  - Open MPI 1.6.3 with our Peruse enhancements
- **Experiment 1**: MPI Alltoall microbenchmark
  - 30 profiled and 30 un-profiled trials
  - 20,000 collective calls (19,998 measured) per trial
- **Average communication overhead = 4.08%**
- **Experiment 2**: NAS Parallel FT Benchmark
  - 30 profiled and 30 un-profiled trials
  - **Average runtime overhead = 0.205%**

**Visualizing Network Communication**
- **Experiment**: TSUBAME-KFC
- **Visualizing the NPB FT benchmark (problem class: E) running on 512 nodes of TSUBAME2.5**