OSAKA UNIVERSITY

Cybermedia Center
Osaka University

Towards Realizing a Dynamic and MPI Application-aware Interconnect with SDN

Keichi Takahashi
Cybermedia Center, Osaka University

The 26th Workshop on Sustained Simulation Performance (WSSP26)

Agenda

Introduction

- Why do we need an MPI Application-aware Interconnect?

SDN-accelerated MPI Primitives

- Is our idea feasible at all?

A Coordination Mechanism of Computation and Communication

- How do we reconfigure the interconnect in accordance with the execution of applications?

A Toolset for Analyzing Application-aware Dynamic Interconnects

- How will the proposed architecture perform with various types of applications and clusters?

Challenges in Future Interconnects

Over-provisioned designs might not scale well

- Interconnects can consume up to 50% of total power [1] and 1/3 of total budget of a cluster [2]
- Properties such as full bisection bandwidth and non-blocking may become increasingly difficult to achieve
- Need to improve the utilization of the interconnect

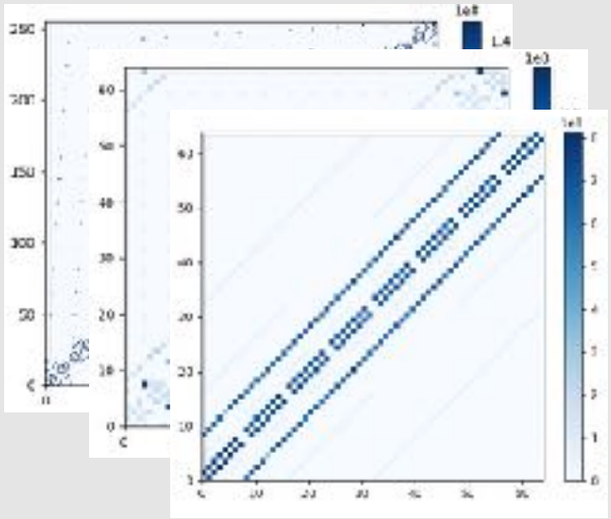
Our proposal is to adopt:

- Dynamic (adaptive) routing
- Application-awareness network control

[1] J. Kim et al."Flattened Butterfly : A Cost-Efficient Topology for High-Radix Networks," ISCA, vol. 35, no. 2, pp. 126–137, 2007.
[2] D. Abts et al., "Energy proportional datacenter networks," ACM SIGARCH Comput. Archit. News, vol. 38, no. 3, p. 338, 2010.

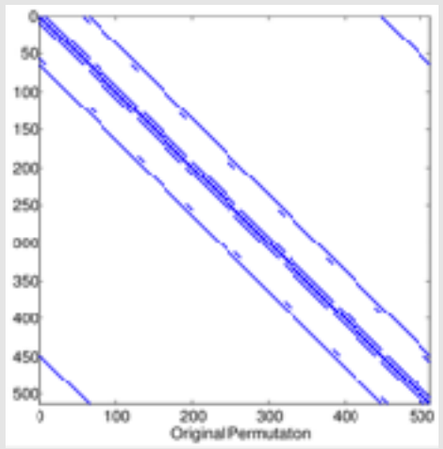
Source of Inefficiency in Current Interconnect

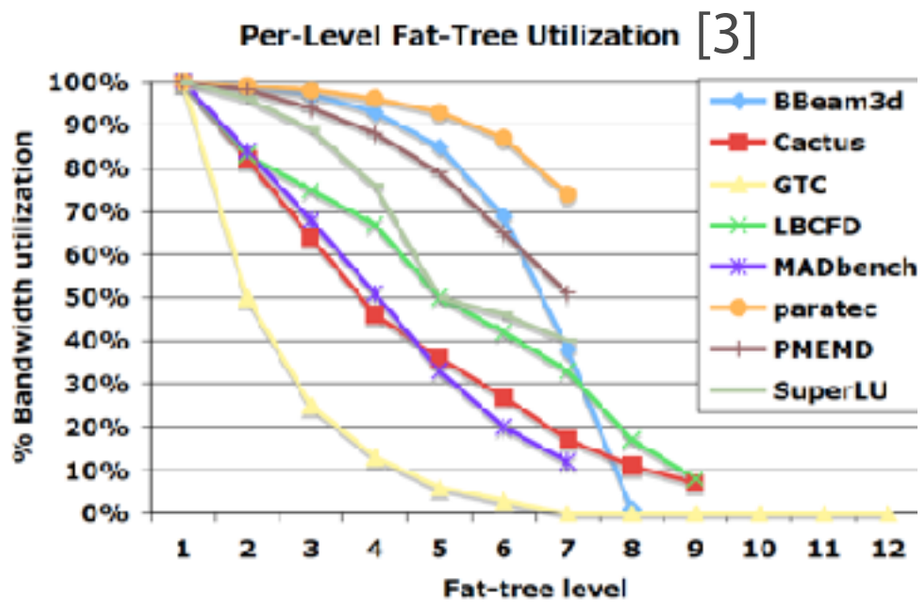
Communication Pattern of Applications



Mismatch

Topology of the Interconnect





[3]

Lower utilization
Higher congestion
Lower communication performance

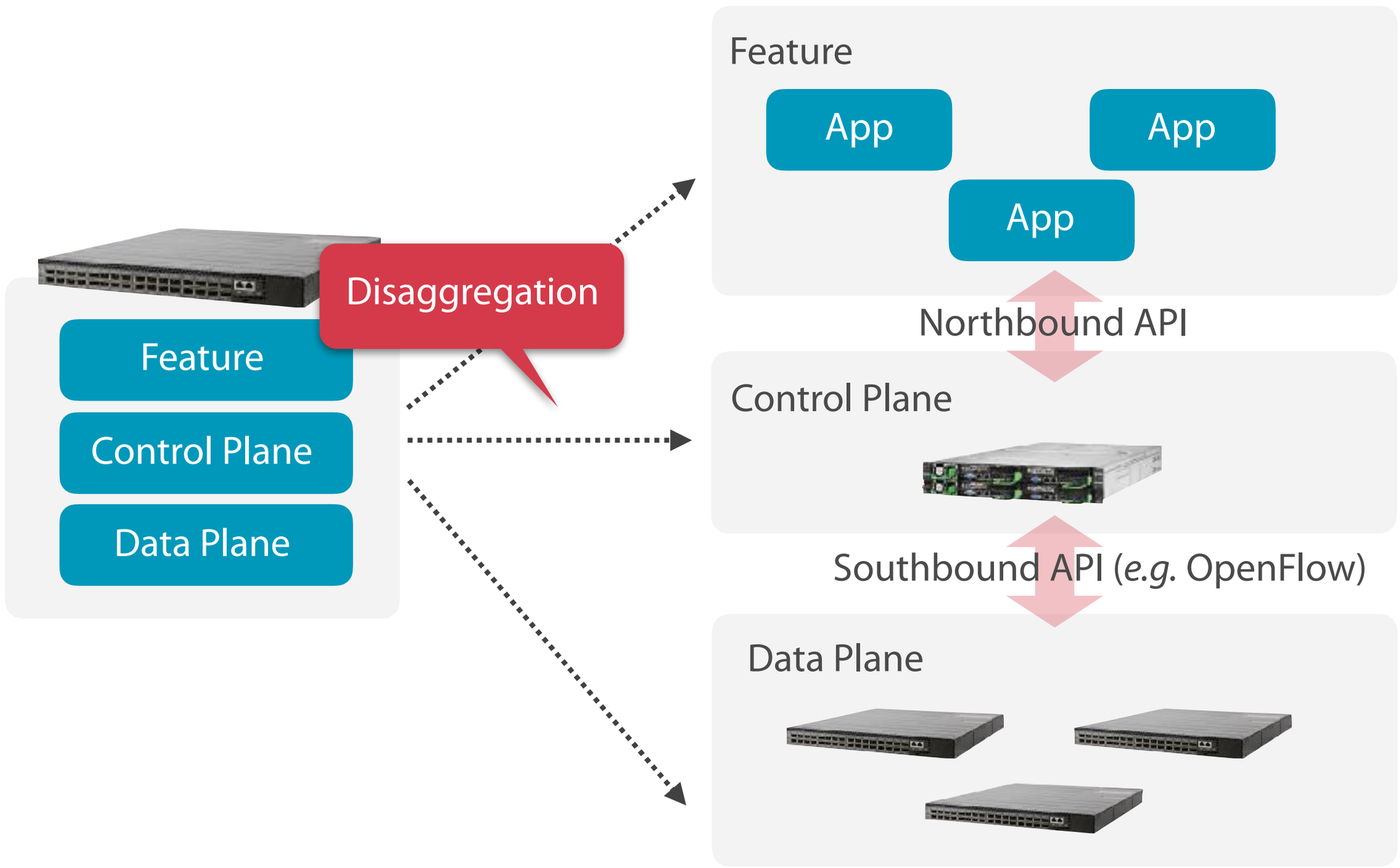
[3] S. Kamil et al. ,"Communication Requirements and Interconnect Optimization for High-End Scientific Applications," *IEEE Trans. Parallel Distrib. Syst.*, 2010.

SDN-enhanced MPI Framework

Our prototype framework that integrates SDN into MPI

- Dynamically controls the interconnect based on the communication pattern of MPI applications
- Uses Software-Defined Networking (SDN) as a key technology to realize dynamic interconnect control (*e.g.* dynamic routing)
- Successfully accelerated several MPI primitives (*e.g.* MPI_Bcast and MPI_Allreduce)

SDN | Software-Defined Networking



Conventional Networking

Software Defined Networking

OpenFlow | Standard implementation of SDN

Control Plane

OpenFlow Controller

Notify flow entry misses

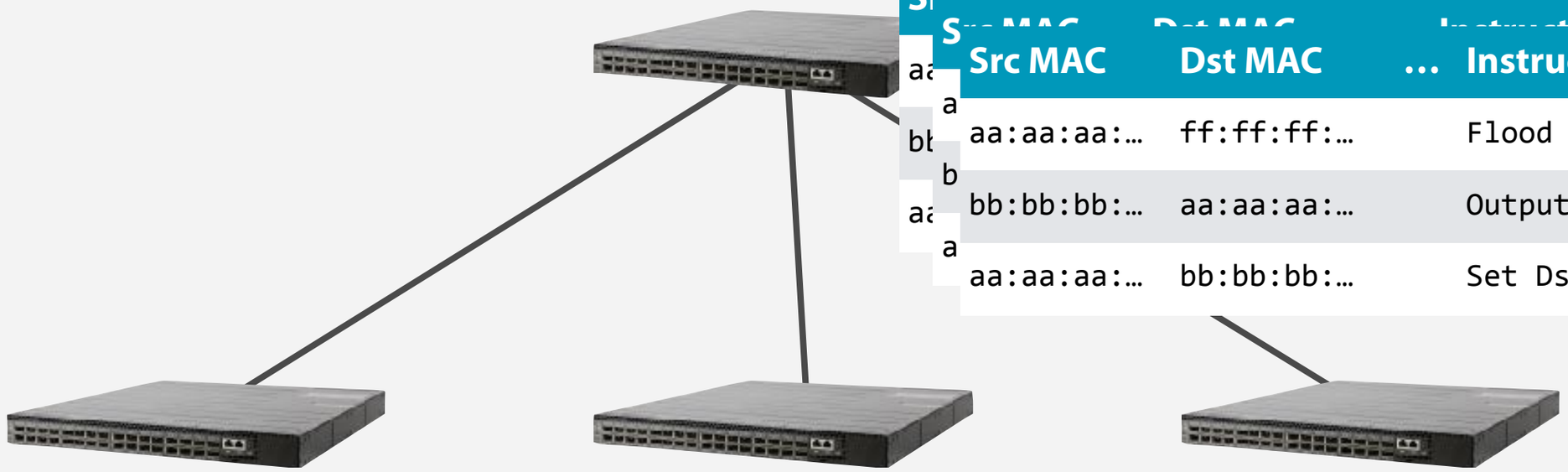
Add/Modify/Delete flow entries
Inject packets into data plane

OpenFlow Messages

Data Plane

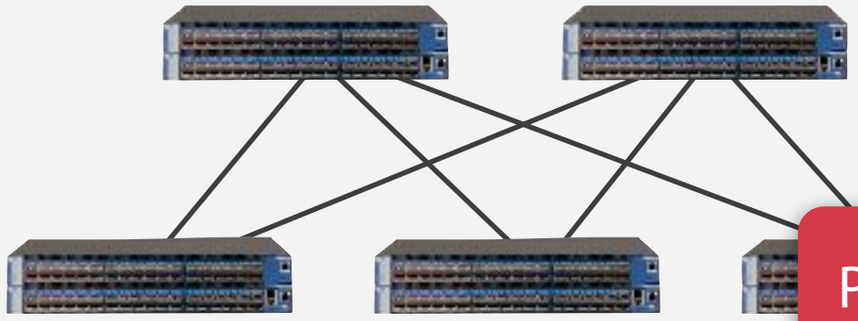
Flow Table (Collection of flow entries)

Src MAC	Dst MAC	…	Instructions
aa:aa:aa:…	ff:ff:ff:…		Flood
bb:bb:bb:…	aa:aa:aa:…		Output to Port X
aa:aa:aa:…	bb:bb:bb:…		Set Dst IP to Y,…



Basic Idea of SDN-enhanced MPI

Interconnect



Interconnect Control Seq.

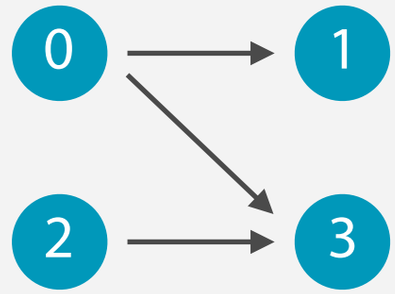
Proactive reconfiguration using OpenFlow

Resource (Path, Bandwidth, *etc.*)
Allocation

Computing Nodes

Extract via Tracer/Profiler
Static Analysis

Communication Pattern



Related Work

SDN-enhanced InfiniBand (Lee et al. *SC16*)

- Enhancement to InfiniBand that allows dynamic and per-flow level network control

Conditional OpenFlow (Benito et al. *HiPC 2015*)

- Enhanced OpenFlow that allows users to add flow entries that are activated when an Ethernet Pause (IEEE 802.3x) occurs
- Primary goal is to implement non-minimal adaptive routing on Ethernet

Quantized Congestion Notification Switch (Benito et al. *HiPINEB 2017*)

- Another enhancement to OpenFlow that uses received QCNs (802.1 Qau Quantized Congestion Notification) to probabilistically determine which path to select

Agenda

Introduction

- Why do we need an MPI Application-aware Interconnect?

SDN-accelerated MPI Primitives

- Can we accelerate MPI primitives based on our idea?

A Coordination Mechanism of Computation and Communication

- How do we reconfigure the interconnect in accordance with the execution of applications?

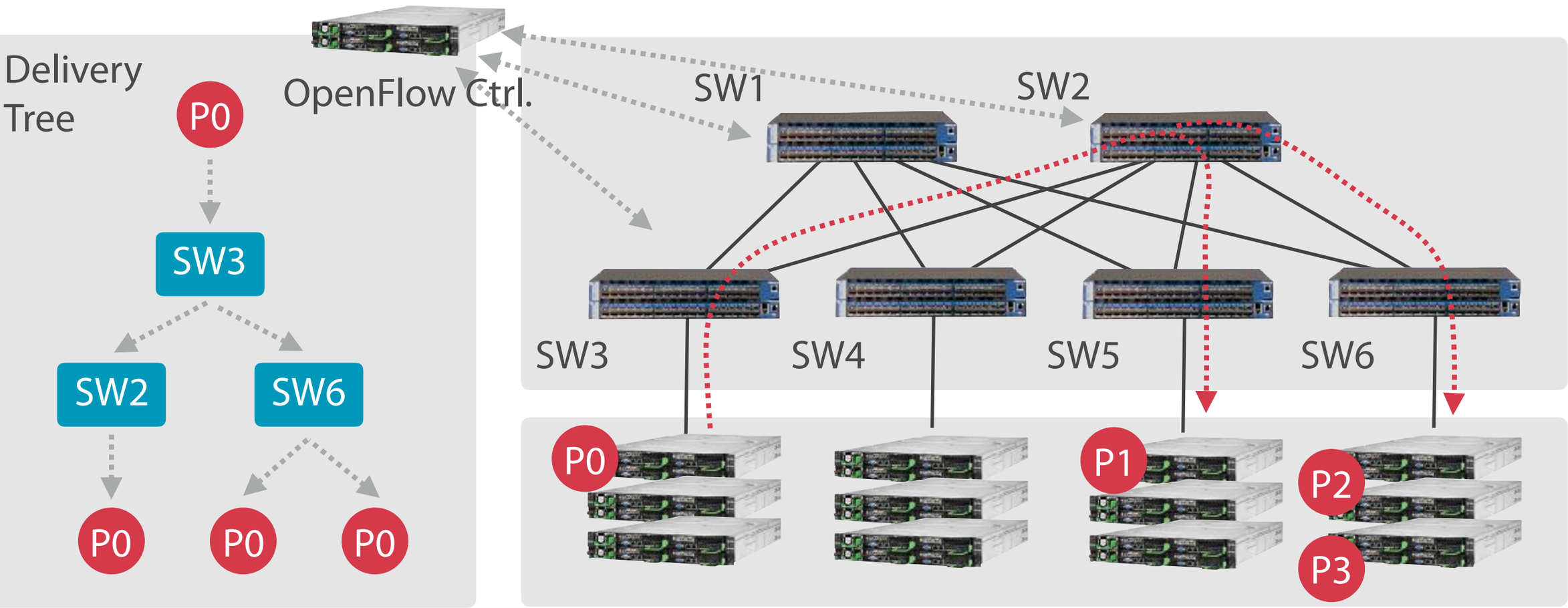
A Toolset for Analyzing Application-aware Dynamic Interconnects

- How will our idea on various types of applications and clusters?

SDN-accelerated MPI_Bcast

MPI broadcast leveraging hardware-multicast

- Multicast rules are dynamically installed using OpenFlow
- Considers background traffic from other jobs to construct optimal delivery tree



Agenda

Introduction

- Why do we need an MPI Application-aware Interconnect?

SDN-accelerated MPI Primitives

- Is our idea feasible at all?

A Coordination Mechanism of Computation and Communication

- How do we reconfigure the interconnect in accordance with the execution of applications?

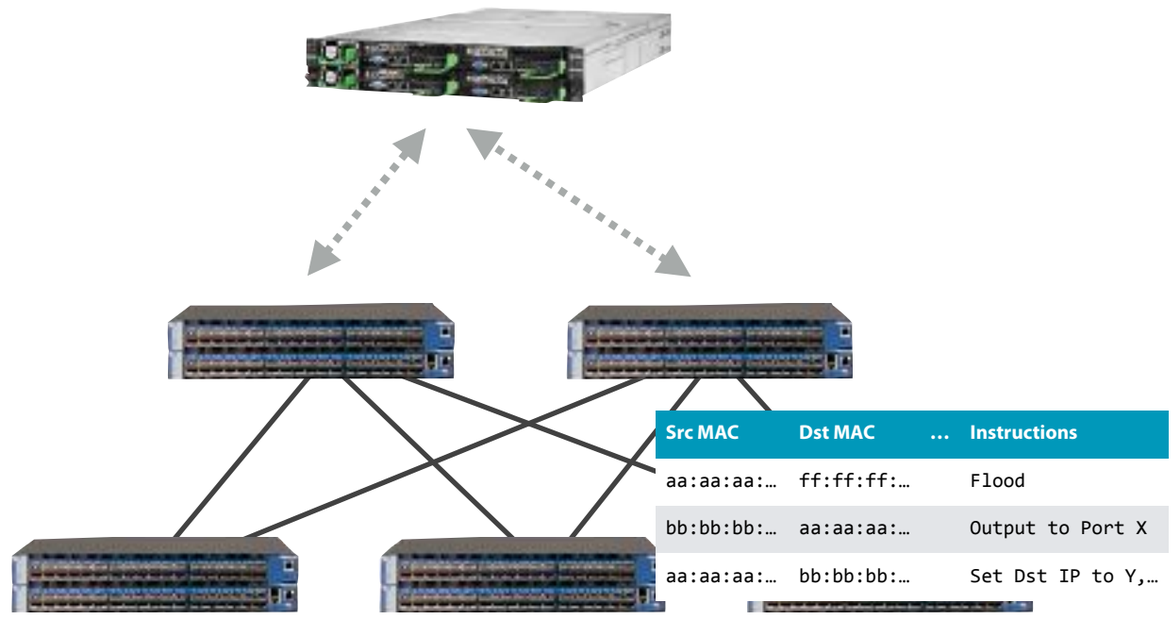
A Toolset for Analyzing Application-aware Dynamic Interconnects

- How will our idea on various types of applications and clusters?

Synchronizing Computation and Communication

Execution

```
#include <mpi.h>

int main() {
    MPI_Init(&argc, &argv);

    MPI_Bcast(buf, count, …);

    /* … */

    MPI_Allreduce(buf, count, …);

    MPI_Finalize();
}
```



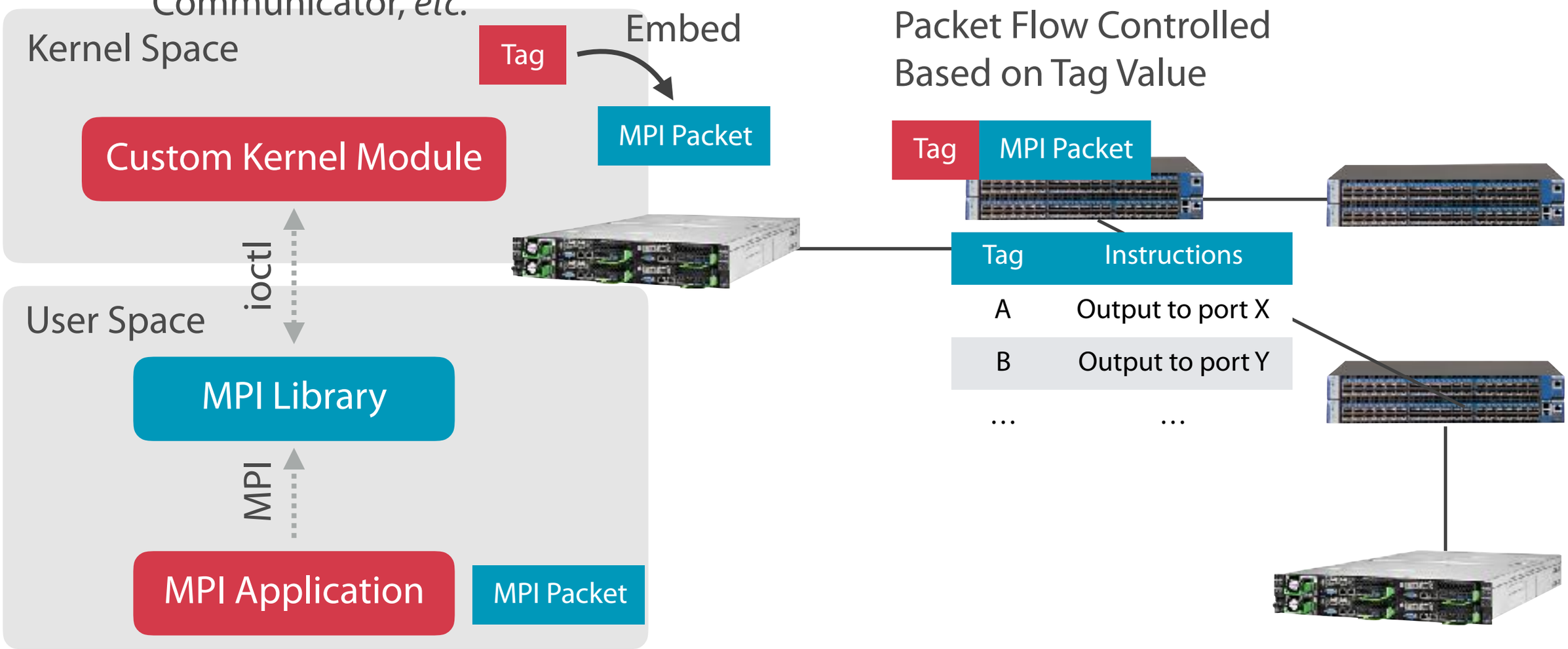
Time-varying Communication Pattern ⟷ **Reconfiguration of the Interconnect**

How do we synchronize these two?

UnisonFlow

Our Idea: Embed encoded MPI envelope into each packet

- Current implementation uses virtual MAC addresses to represent tags



[5] Keichi Takahashi, "Concept and Design of SDN-enhanced MPI Framework", *EWSDN 2015*

Agenda

Introduction

- Why do we need an MPI Application-aware Interconnect?

SDN-accelerated MPI Primitives

- Is our idea feasible at all?

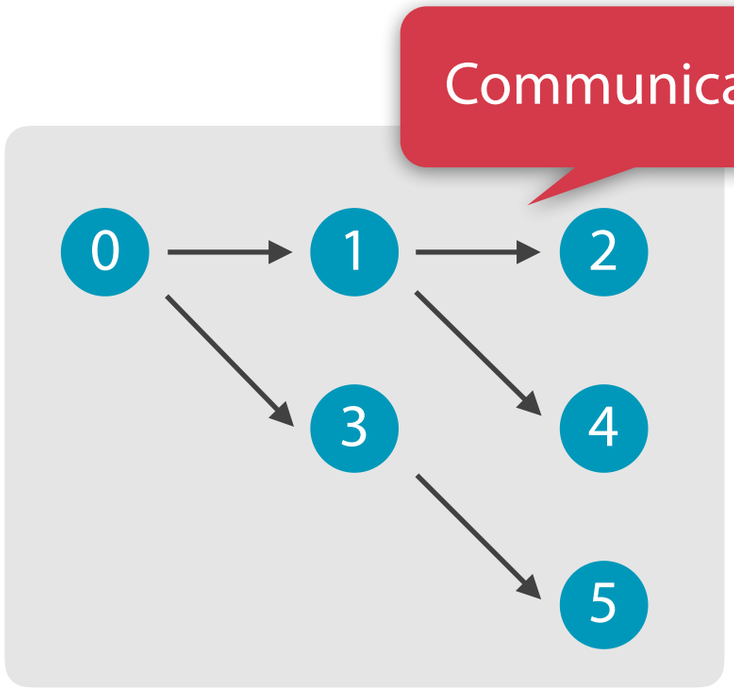
A Coordination Mechanism of Computation and Communication

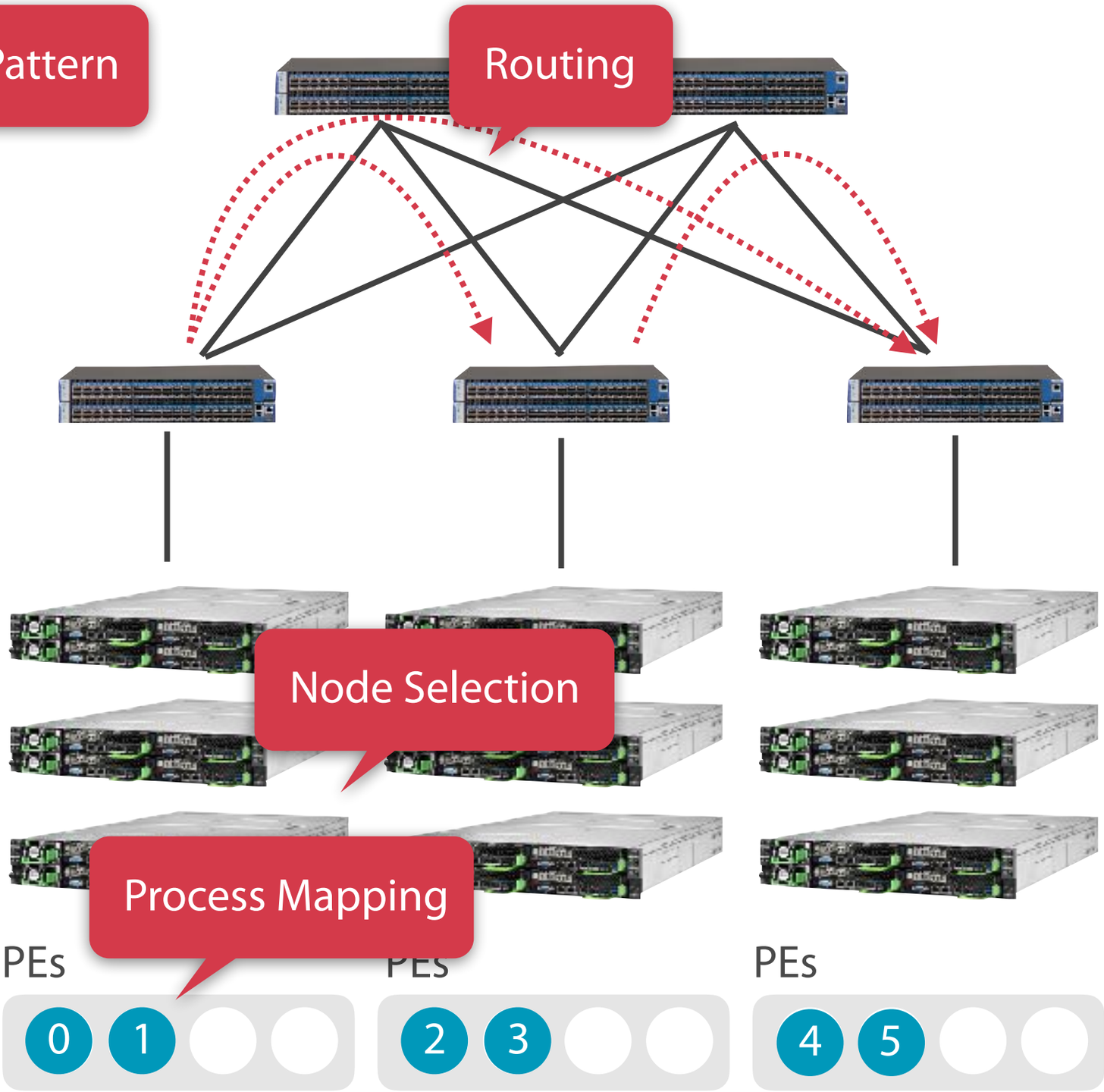
- How do we reconfigure the interconnect in accordance with the execution of applications?

A Toolset for Analyzing Application-aware Dynamic Interconnects

- How will our idea on various types of applications and clusters?

Need for a Holistic Analysis in SDN-enhanced MPI





slurm
workload manager

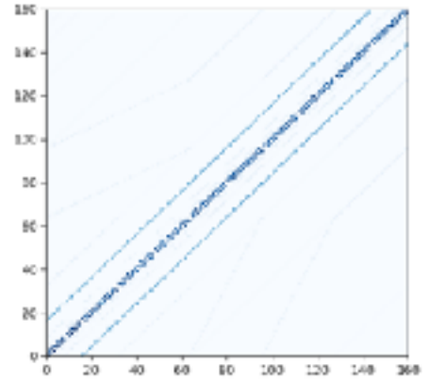
Q1: Impact of the Communication Pattern

How does the traffic load in the interconnect change for diverse applications?

- What kind of application benefits most from SDN-enhanced MPI?
- What happens if the number of processes scales out?

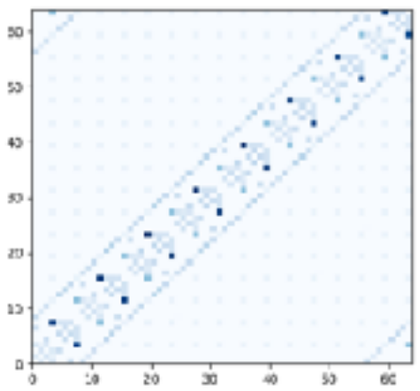
OpenFOAM

The Open Source CFD Toolbox



Gromacs

fast
flexible
free

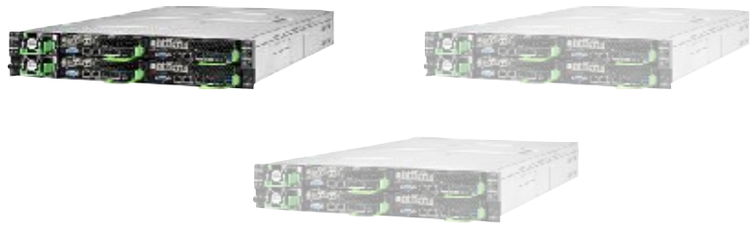


Q2: Impact of the Cluster Configuration

How does the traffic load in the interconnect change under diverse clusters with different configurations?

- How do job scheduling, node selection and process mapping affect the performance of applications?
- How does the topology of the interconnect impact the performance?
- What happens if the size of cluster scales out?

Job Scheduling
(*i.e.* which job should be executed next?)



Node Selection
(*i.e.* which node should be allocated to a given job?)

Process Placement
(*i.e.* which node should execute a process?)

Requirements for the Interconnect Analysis Toolset

We aim to develop a toolset to help answer these questions

- How does the traffic load in the interconnect change for diverse applications?
- How does the traffic load in the interconnect change under diverse clusters with different configurations?

Simulator-based approach is taken to allow rapid assessment

- Requirements for the toolset are summarized as:

1. Support for application-aware dynamic routing

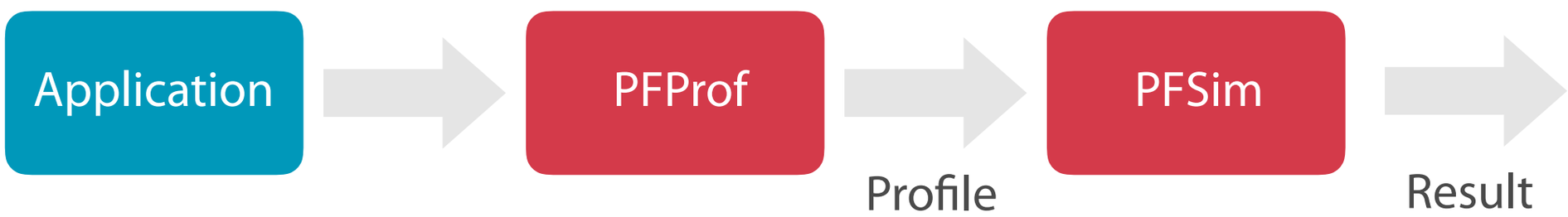
2. Support for communication patterns of real-world applications

3. Support for diverse cluster configurations

Overview of PFAnalyzer

PFProf (profiler) and PFSim (simulator) constitute PFAnalyzer [6]

- PFProf
  - Fine-grained MPI profiler for observing network activity caused by MPI function calls (Requirement 2)
- PFSim
  - Lightweight simulator to simulate traffic load in the interconnect targeting application-aware dynamic interconnects (Requirement 1, 2, 3)



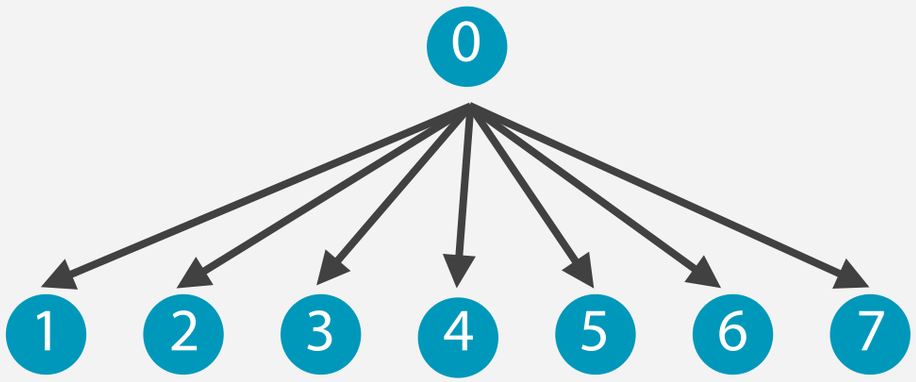
[6] Keichi Takahashi et al., "A Toolset for Analyzing Application-aware Dynamic Interconnects", *HPCMASPA 2017*

PFProf: Motivation

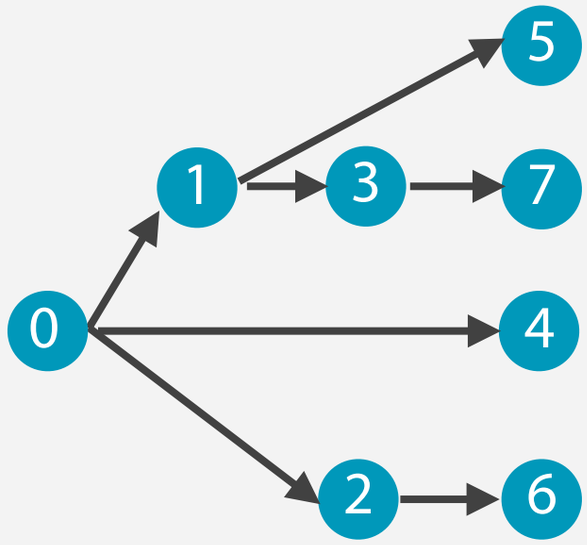
Existing profilers do not capture the underlying pt2pt communication of collective communication

- They are designed to support code tuning and optimization, not network traffic analysis.
- MPI Profiling Interface (PMPI) only captures individual MPI function calls.

Behavior of MPI_Bcast as seen from applications



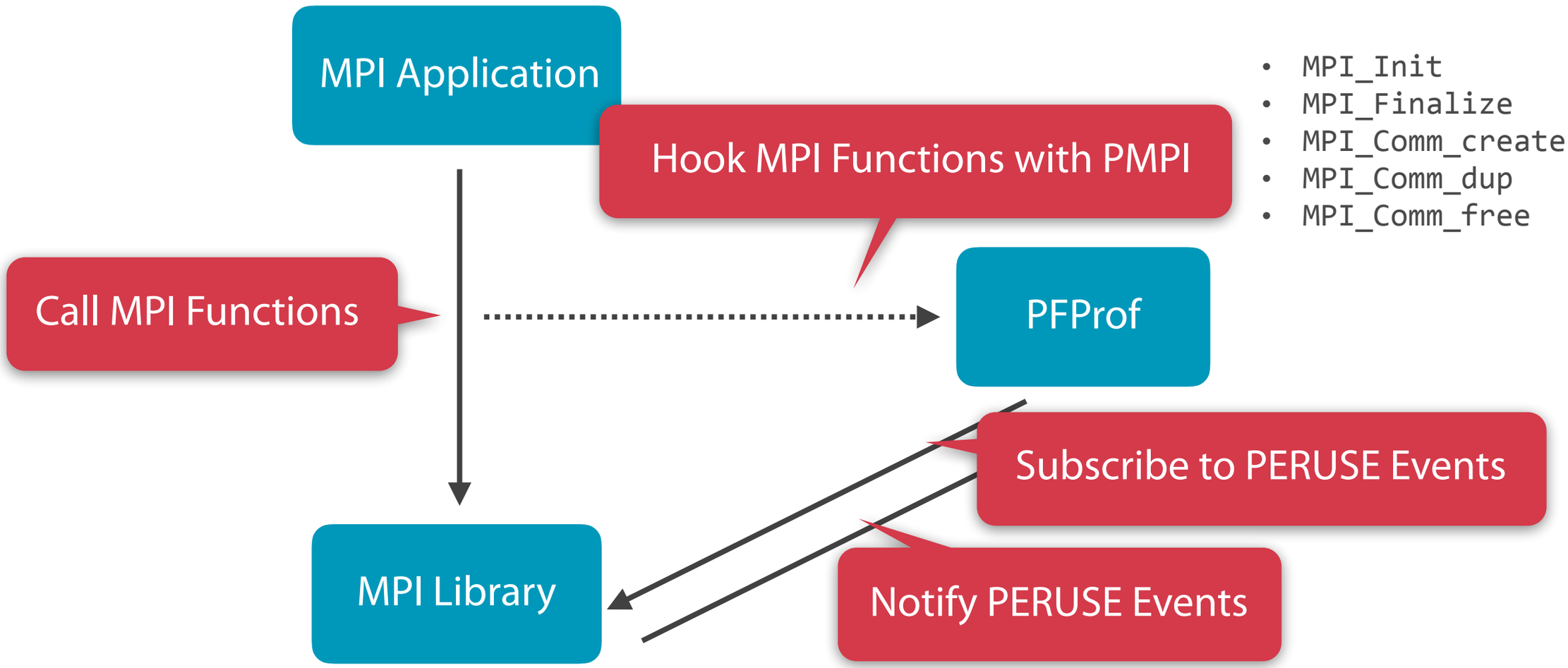
Actual communication performed



PFProf: Implementation

MPI Performance Revealing Extension Interface (PERUSE) is utilized

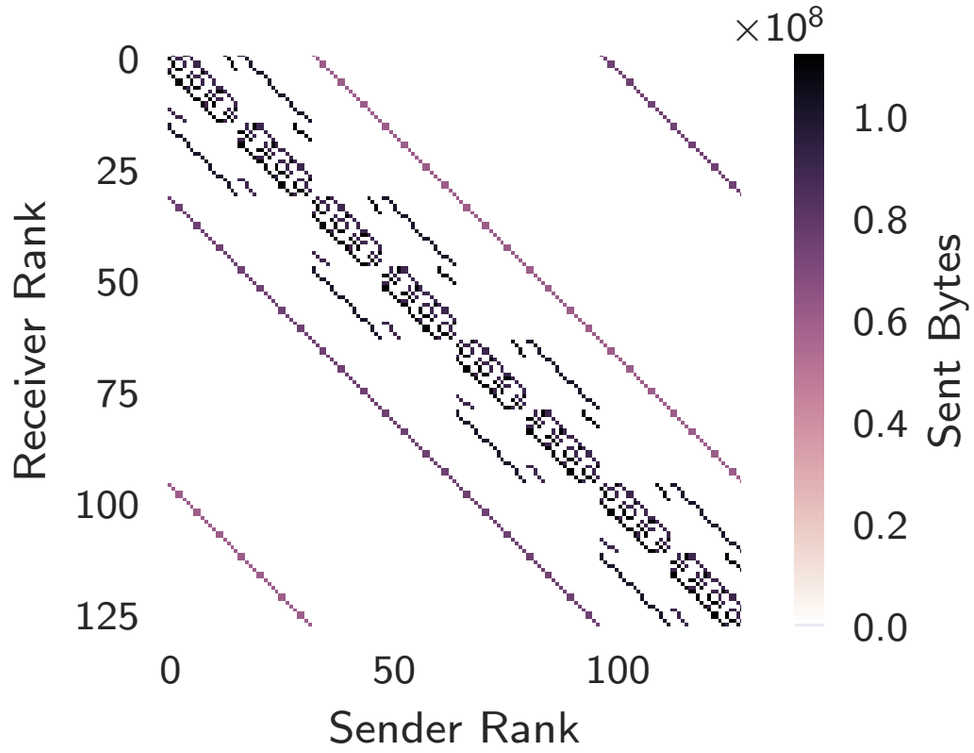
- PERUSE exposes internal information of MPI library
- Notifies you when a request is posted/completed, a transfer begins/ends, *etc.*



Representation of Communication Pattern

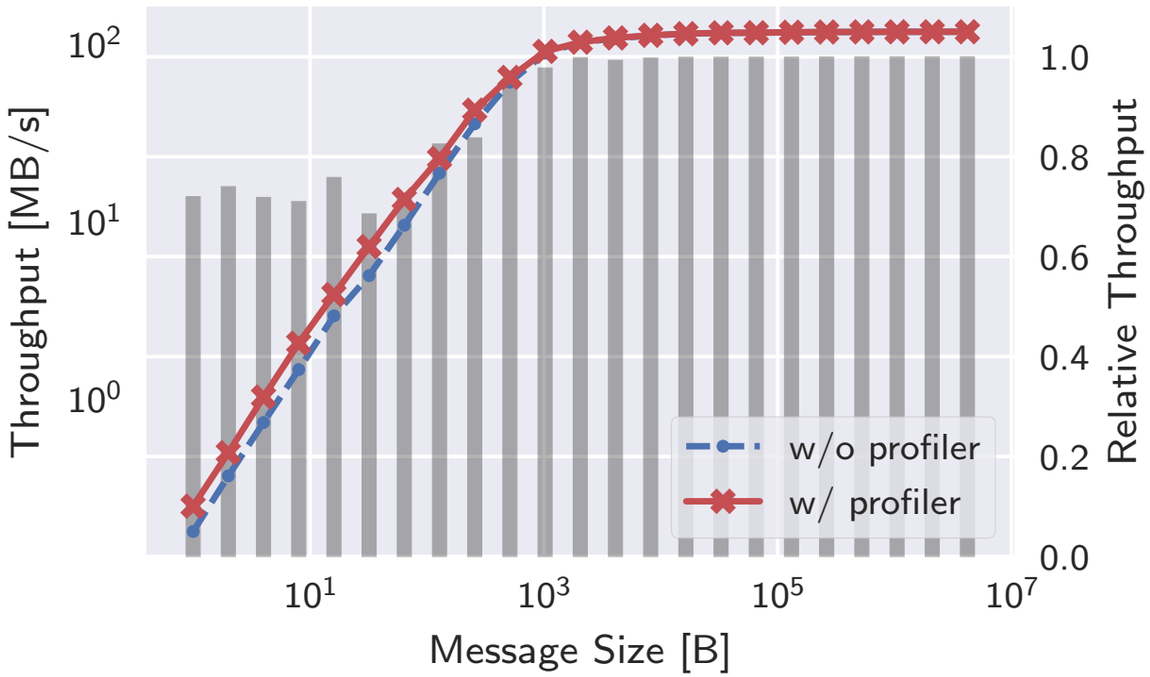
The communication pattern of an application is represented using its traffic matrix

- Defined as a matrix T of which element T_{ij} is equal to the volume of traffic sent from rank i to rank j
- Implies that the volume of traffic between processes as constant during the execution of a job

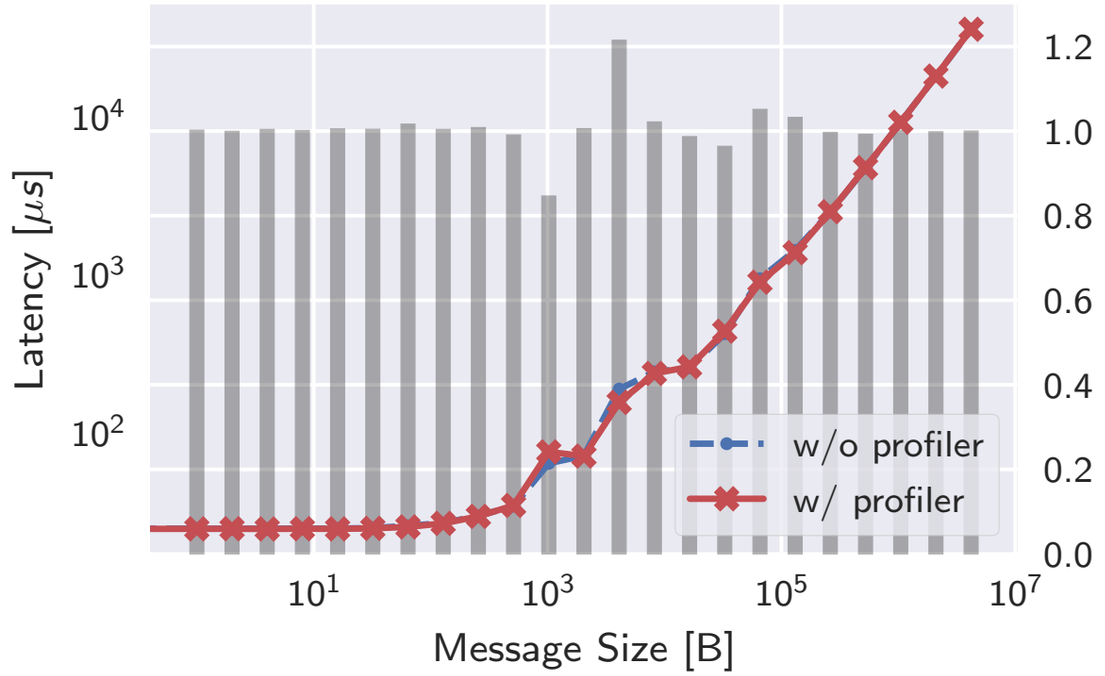


An example obtained from running the NERSC MILC benchmark with 128 processes

PFProf: Overhead Evaluation



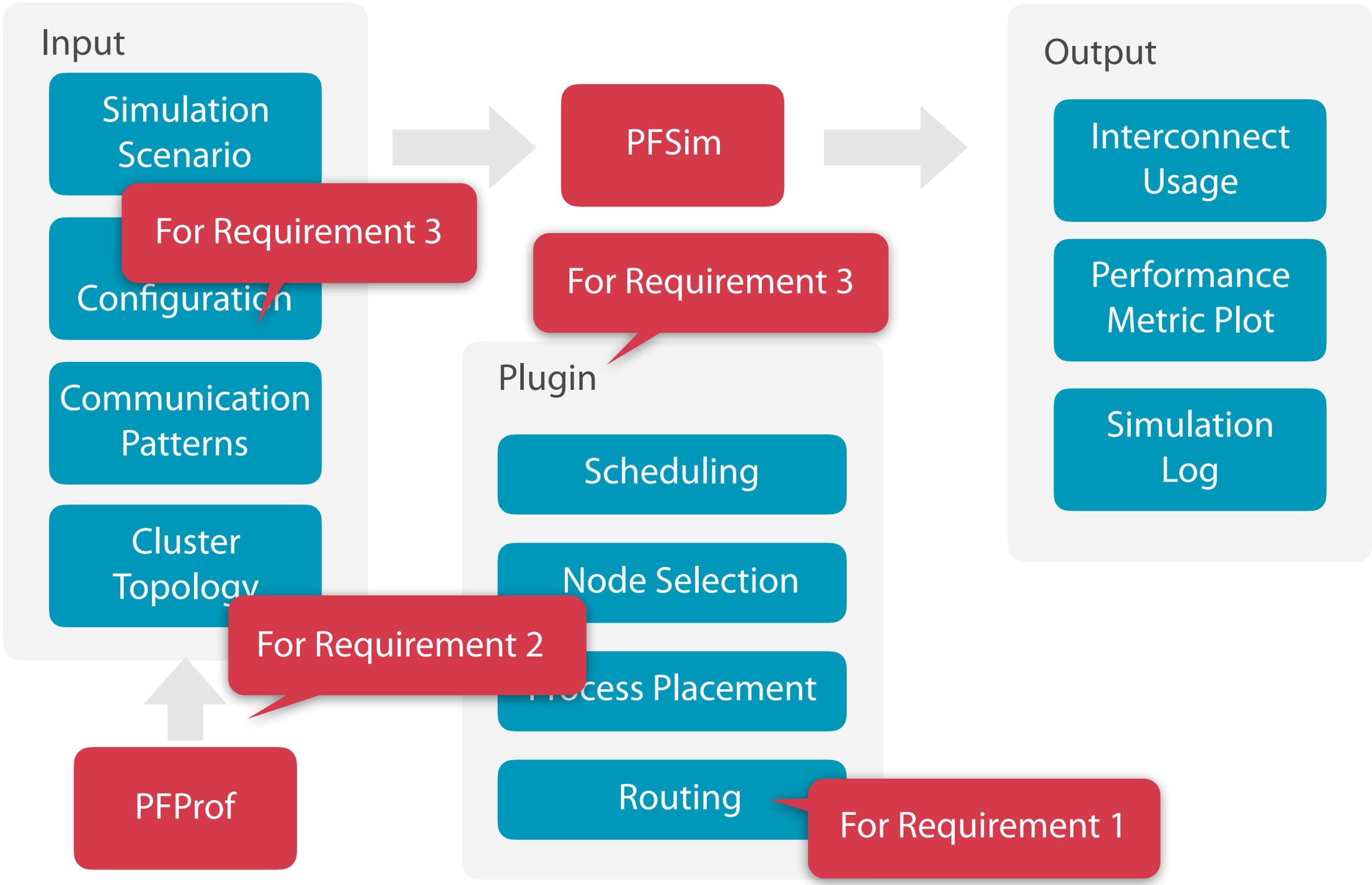
Throughput (osu_bw)



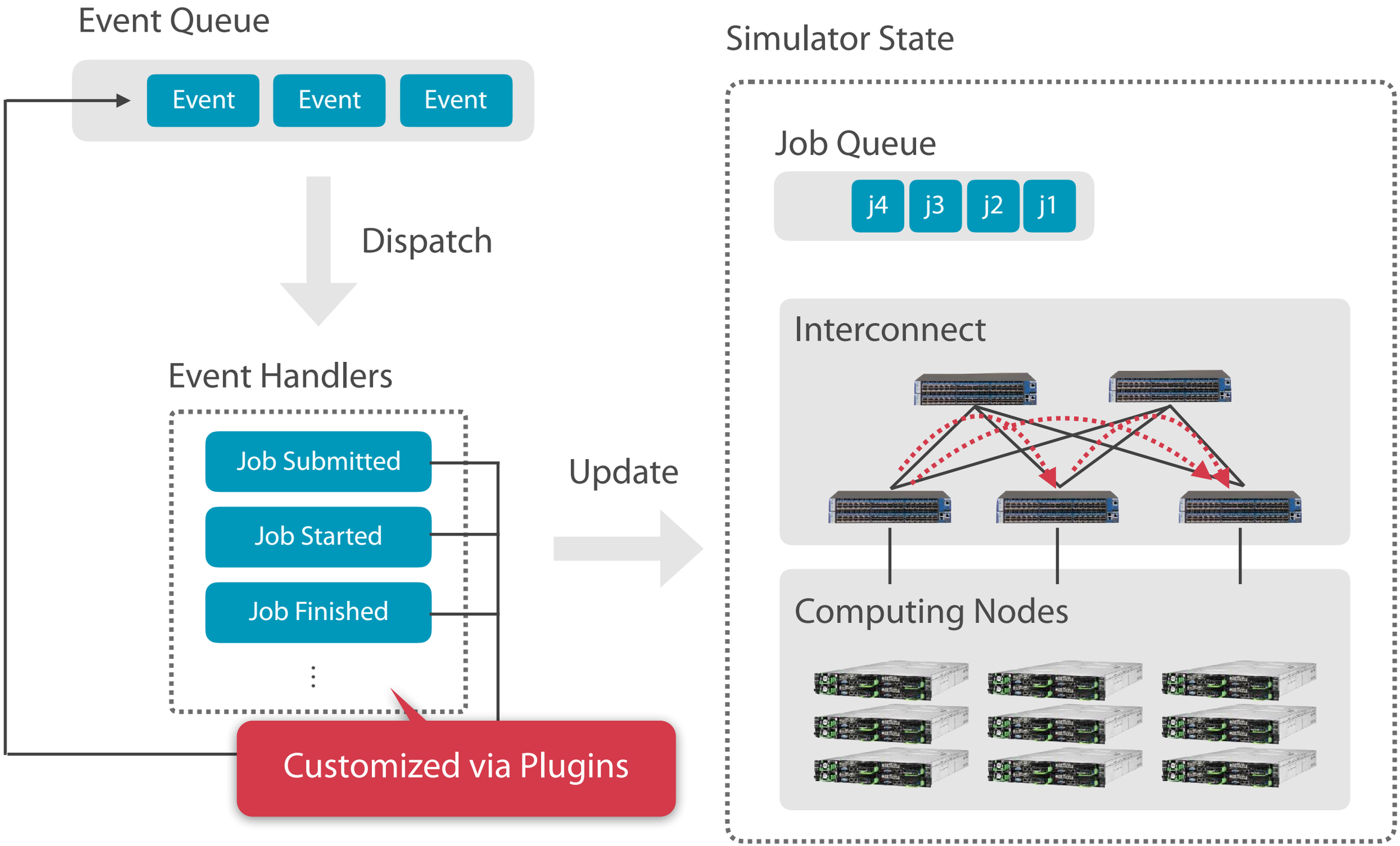
Latency (osu_latency)

Measured throughput and latency of pt2pt communication with and without PFProf using the OSU Microbenchmark

PFSim: Overview



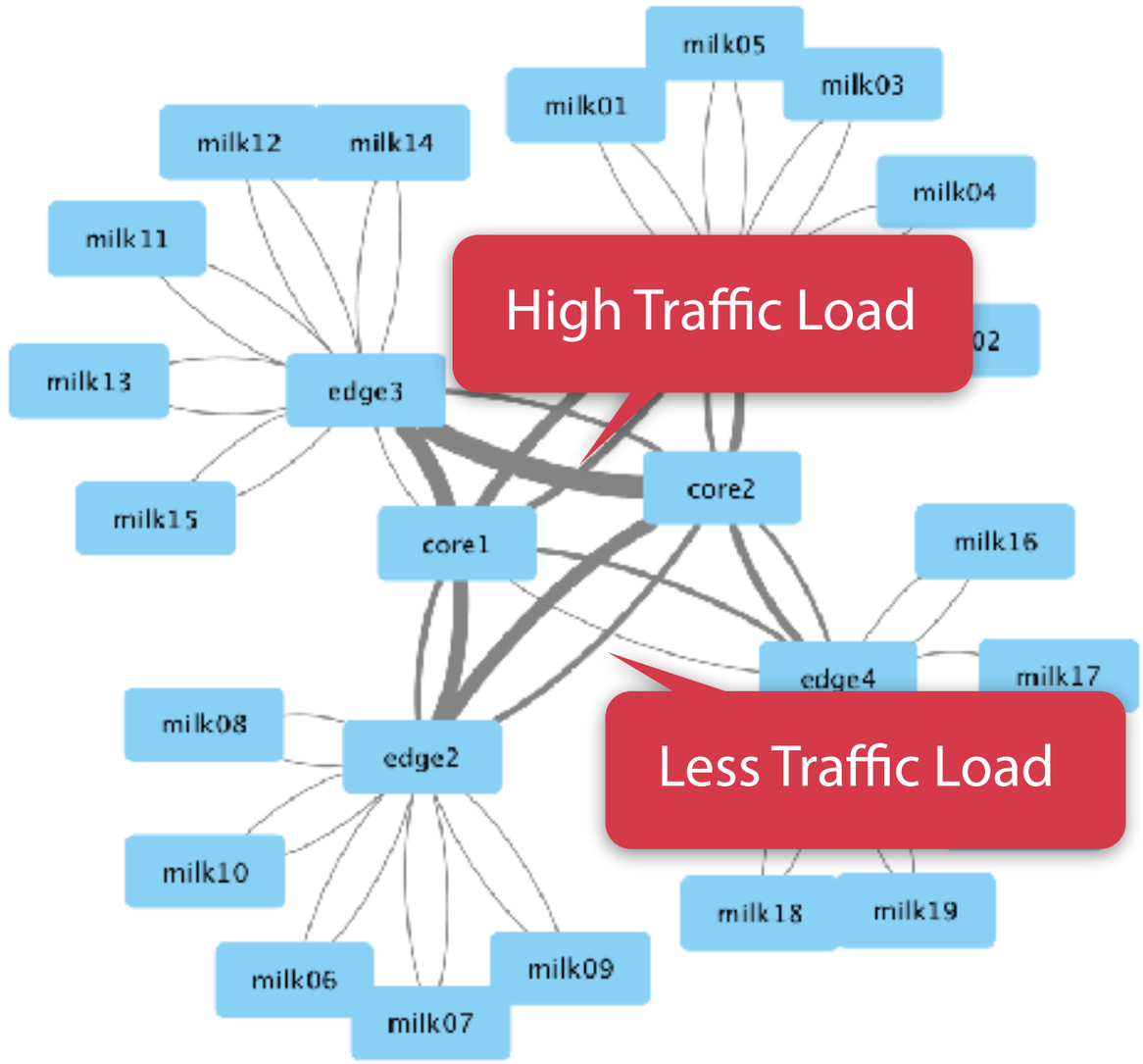
PFSim: Architecture



PFSim: Example Input & Output

```
topology: topologies/milk.graphml
output: output/milk-cg-dmodk
algorithms:
  scheduler:
    - pfsim.scheduler.FCFSScheduler
  node_selector:
    - pfsim.node_selector.LinearNodeSelector
    - pfsim.node_selector.RandomNodeSelector
  process_mapper:
    - pfsim.process_mapper.LinearProcessMapper
    - pfsim.process_mapper.CyclicProcessMapper
  router:
    - pfsim.router.DmodKRouter
    - pfsim.router.GreedyRouter
    - pfsim.router.GreedyRouter2
jobs:
  - submit:
      distribution: pfsim.math.ExponentialDistribution
      params:
        lambd: 0.1
    trace: traces/cg-c-128.tar.gz
```

Cluster Configuration (YAML)

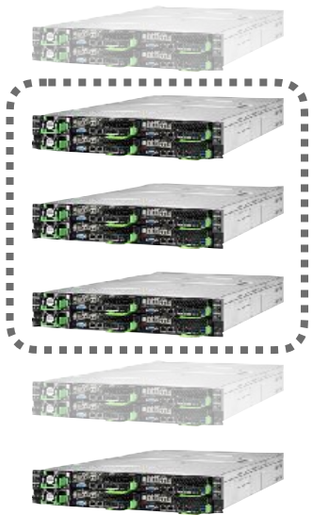


Interconnect Utilization
(Output GraphML visualized with Cytoscape)

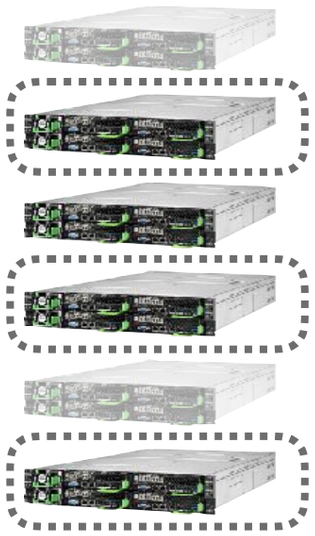
Simulated Configurations

Node Selection

Linear



Random



Process Placement

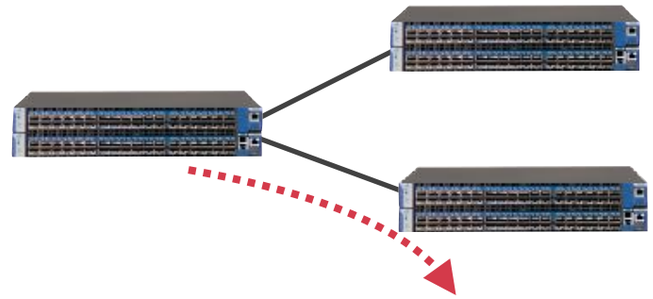
Linear

Cyclic

Routing

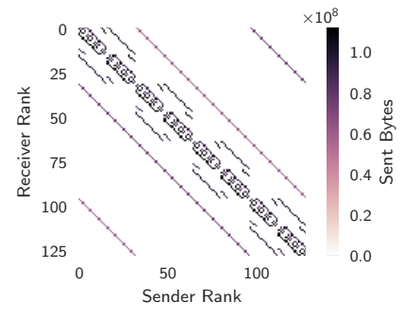
D-mod-K

Path selected solely based on the destination of flow



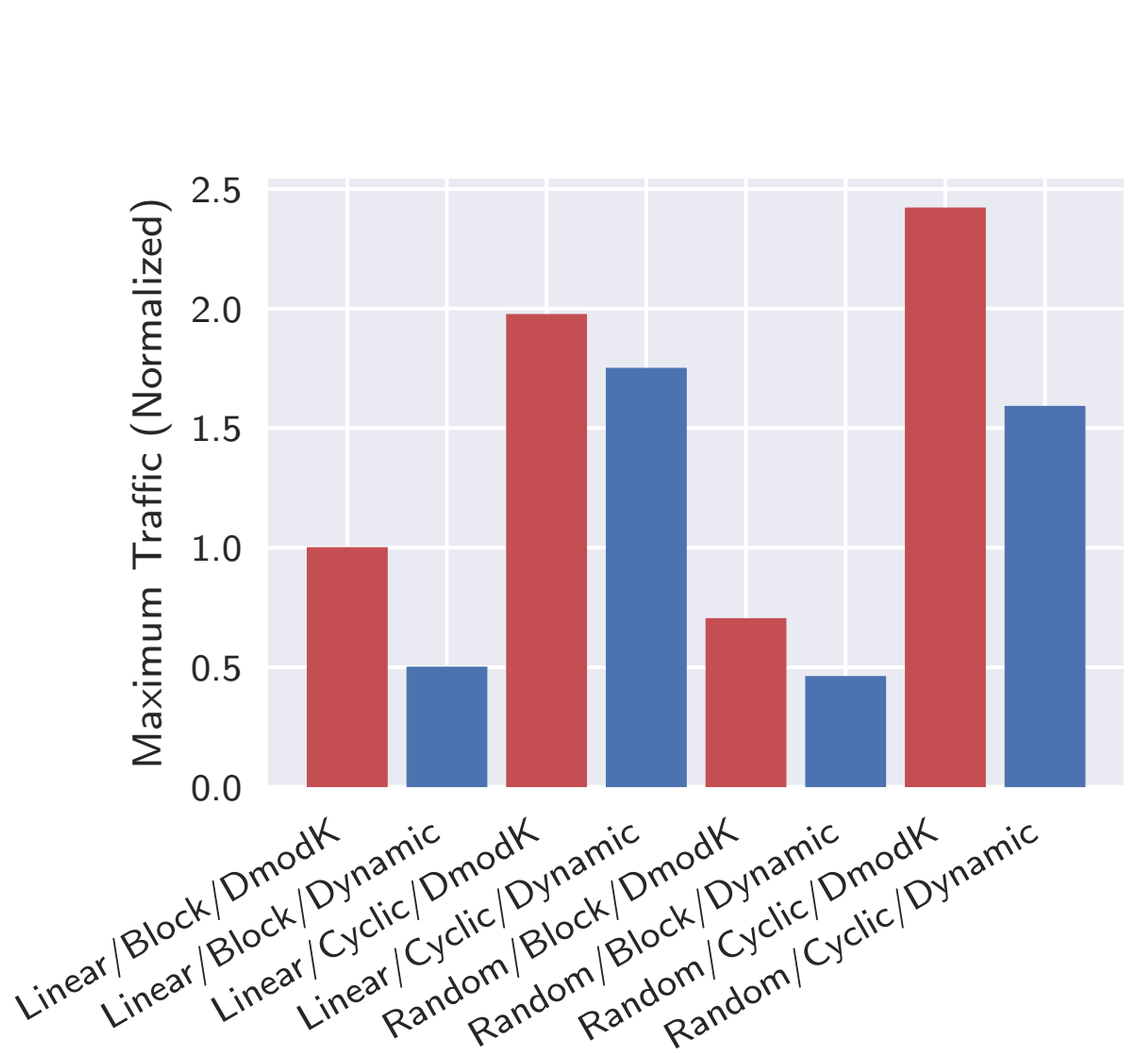
Dynamic

Path allocated based on communication pattern (heavy pairs first)

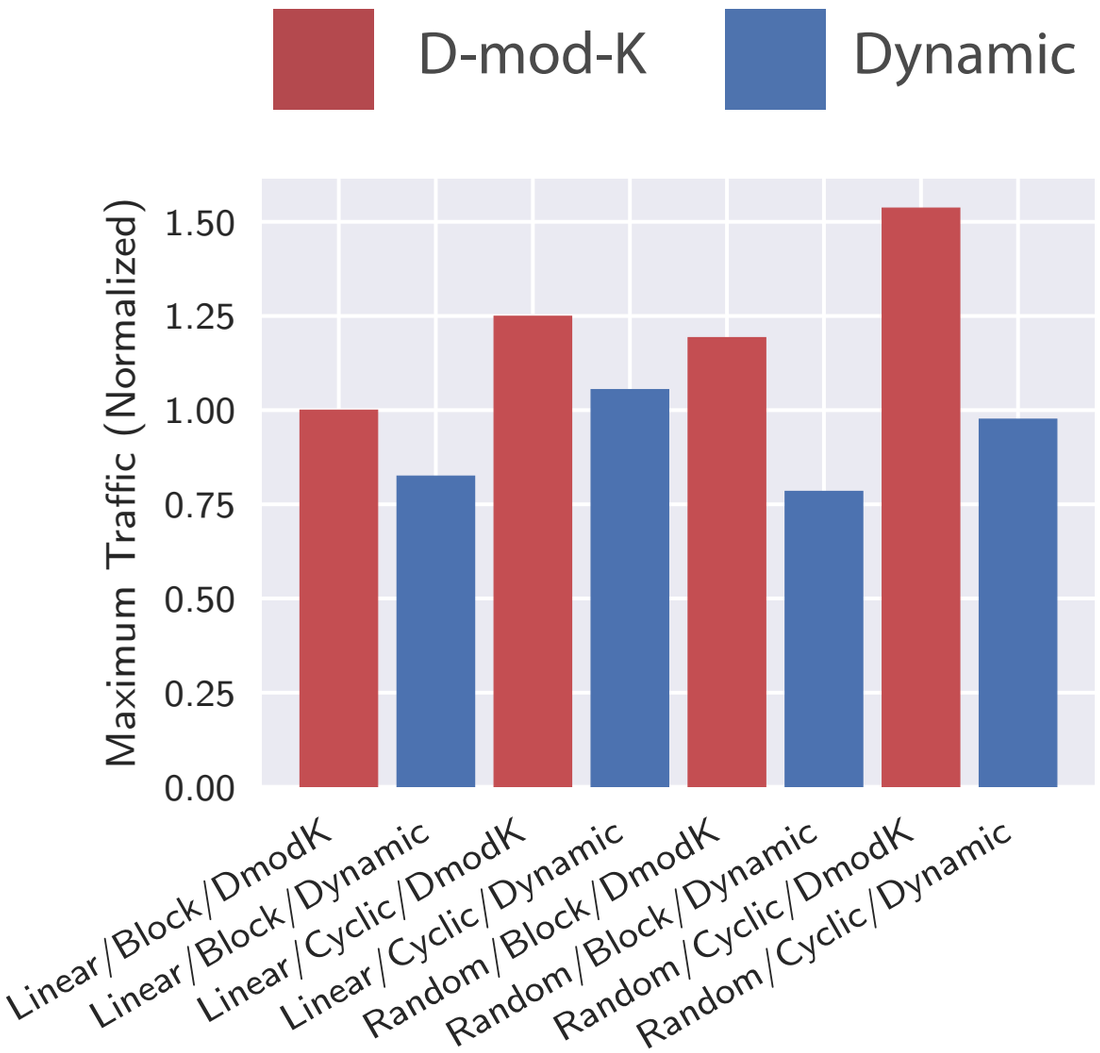


Simulation Results

Maximum traffic load on all links is plotted as a performance indicator



NAS CG Benchmark (128 ranks)



NERC MILC Benchmark (128 ranks)

Further Challenges

Simulation-based study of large-scale clusters with different topologies

- Currently, our institution owns only a small-scale experimental cluster employed with SDN

Integrate interconnect controller with scheduler and MPI runtime

- To support multiple jobs running in parallel
- To investigate the effect of node allocation and process placement

Better application-aware routing algorithms

- Currently, a simple greedy like algorithm is used
- How about optimization or machine learning?

Summary

Current static and over-provisioned interconnects might not scale well

- SDN allows us to build a more dynamic and application-aware interconnects
- Such architecture could improve the utilization of the interconnect and communication performance

Our achievements so far include:

- SDN-accelerated MPI primitives such as Bcast and Allreduce
- UnisonFlow, a coordination mechanism of computation and communication
- PFAnalyzer, a toolset for analyzing application-aware dynamic interconnects