On Massively Parallel Simulation of Large-Scale Fat-Tree Networks for HPC Systems and Data Centers

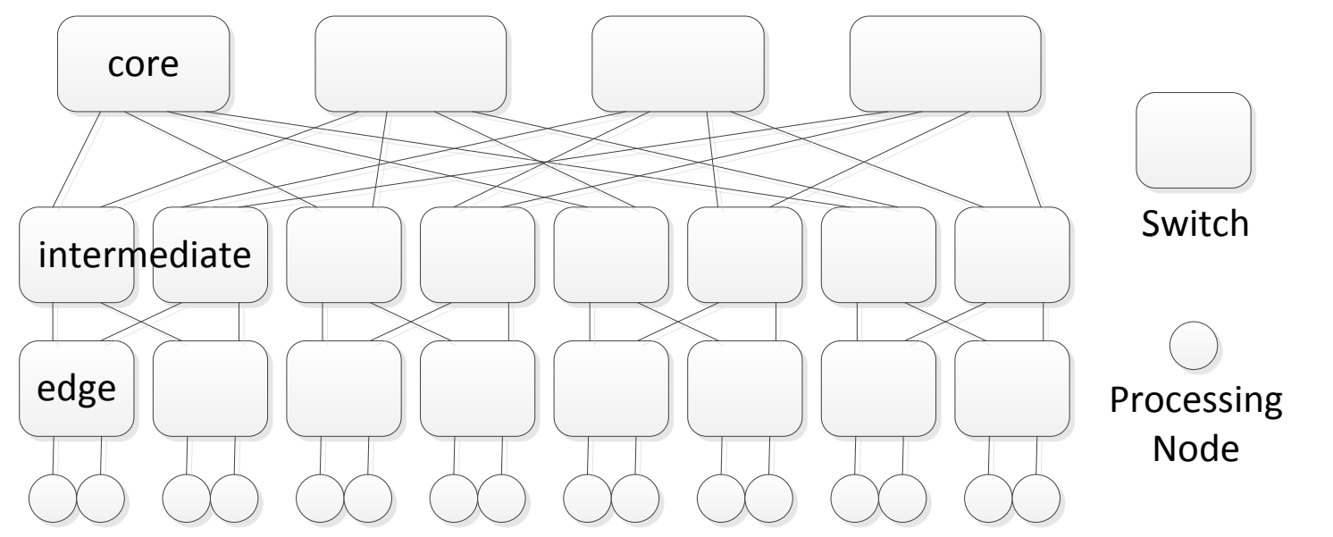
Ning Liu, Xian-He Sun, Dong (Kevin) Jin

Department of Computer Science, Illinois Institute of Technology, Chicago, Illinois.

ILLINOIS INSTITUTE OF TECHNOLOGY

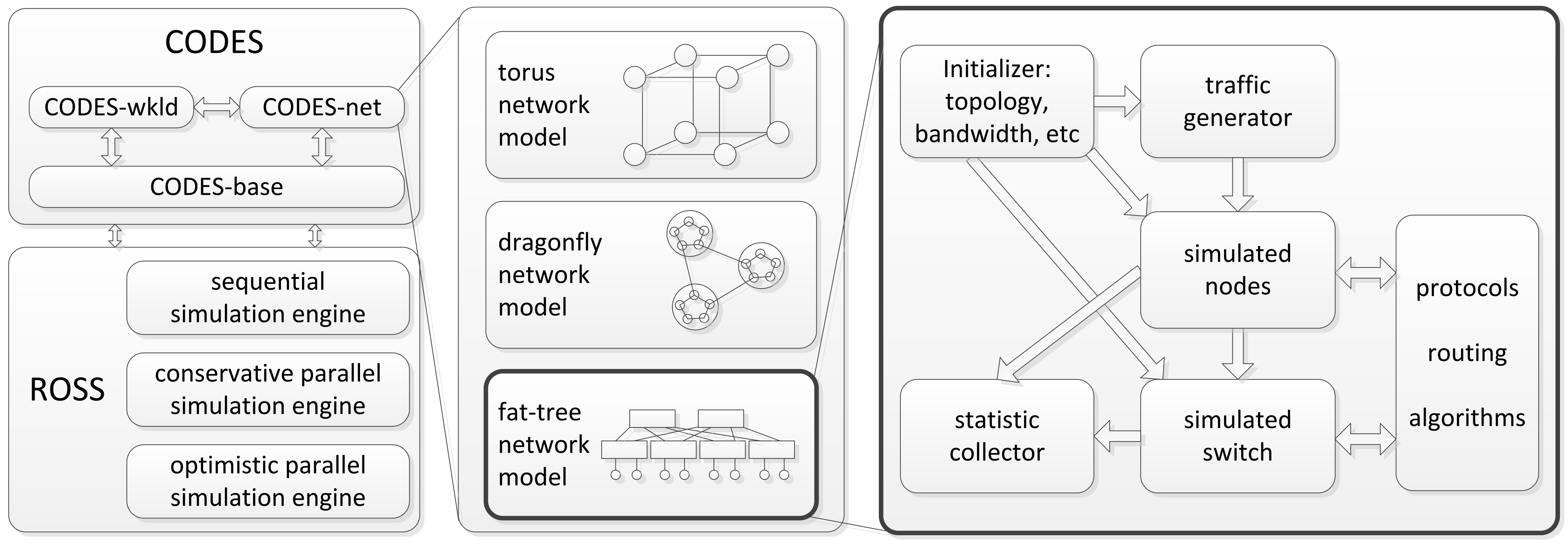
Introduction & Motivation

Fat-tree topologies have been widely adopted as the communication network in data centers in the past decade. Nowadays, high-performance computing (HPC) system designers are considering using fat-tree as the interconnection network for the next generation supercomputers [1]. For extreme-scale computing systems like the data centers and supercomputers, the performance is highly dependent on the interconnection networks. In this work, we present FatTreeSim, a PDES-based toolkit consisting of a highly scalable fat-tree network model, with the goal of better understanding the design constraints of fat-tree networking architectures in data centers and HPC systems, as well as evaluating the applications running on top of the network. FatTreeSim is designed to model and simulate large-scale fat-tree networks up to millions of nodes with protocol-level fidelity.



The design, evaluation and deployment of data center and HPC system is a systematic and time-consuming process. As the key component, the communication network has a significant impact on system performance. Large-scale data center and HPC system network architecture need to support a wide range of applications, each with different communication and I/O requirements. In distributed computing community, it is projected that a single data center can scale out to host millions of virtual machines or even physical servers and serve multi-millions jobs/tasks. The requirements for building a data center network at such a scale also differ with that of the traditional data centers. The communication network must guarantee the high availability and reliability, desirable bisection bandwidth, and support for multi-tenancy. To quantify the design trade-offs of a network at a scale, it is desirable to build a large scale simulation toolkit that is capable of evaluating different design points in an efficient and cost-effective manner. A fat-tree or folded-Clos topology is the conventional and yet still the most prevalent design choice for data center communication networks.

FatTreeSim Architecture



```
procedure GT                                   ▷ generate packet stream
    $t$ =processing delay
    $\tau = rng(I)$
    if RandomDestinationTraffic then
        $dst = rng(maxnodeID)$
        Generate packet (header contains $dst$ )
    else if NearestNeighborTraffic then
        $dst = neighborID$
        Generate packet (header contains $dst$ )
    else
        Unsupported traffic
    end if
    Call NSP procedure with $t$
    Call GT procedure with $\tau$
end procedure
```

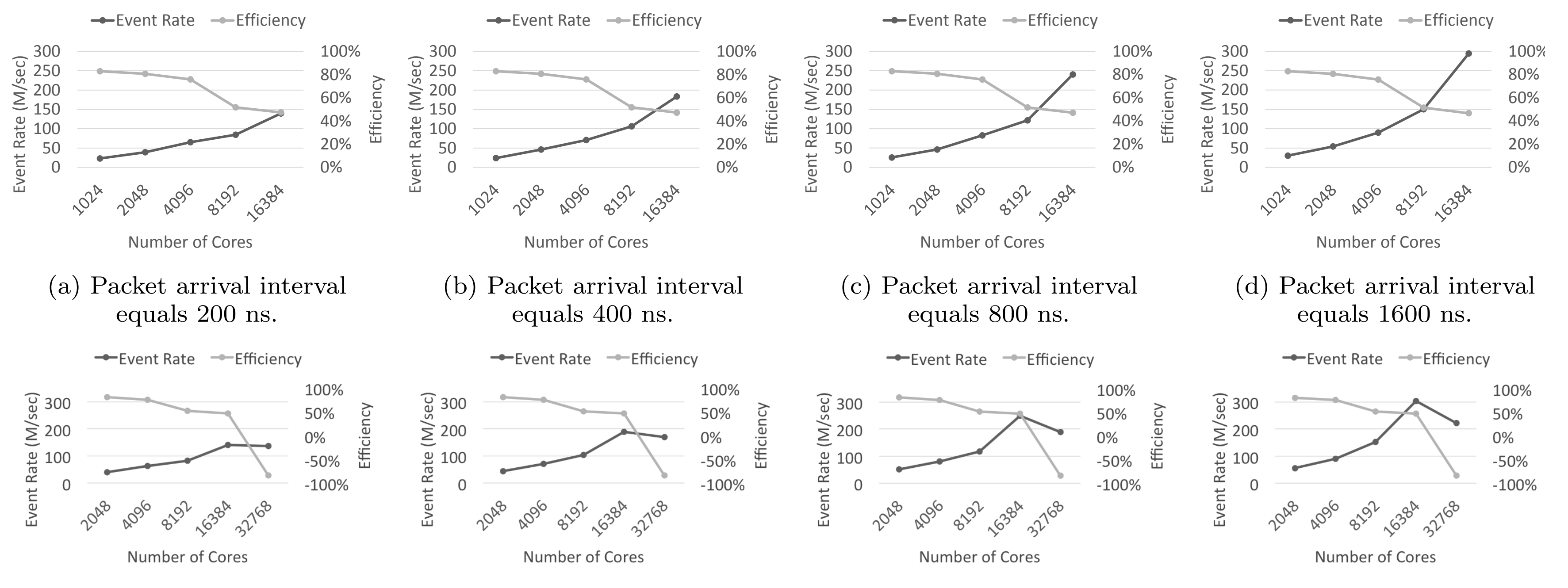
The above figure illustrates FatTreeSim system architecture in CODES and ROSS ecosystem. ROSS[2] is short for Rensselaer Optimistic Simulation System, which features optimistic parallel discrete-event simulation using reverse computation. CODES[3] is short for Enabling Co-Design of Multilayer Exascale Storage Architectures. CODES project is a collaboration between Rensselaer Polytechnic Institute and Argonne National Laboratory. CODES is based on ROSS and leverage the parallel simulation engine provided by ROSS. Currently, CODES is comprised of multiple modules: CODES-base, CODES-workload and CODES-net. CODES-net is comprised of multiple network models. FatTreeSim is designed to be part of the CODES-net module and is the fat-tree network model illustrated in this Figure. FatTreeSim is comprised of multiple components: Initializer, Traffic Generator, Nodes, Switches, Protocols & Routing Algorithms and Statistic Collectors.

Experiment Testbed

Mira [6] , an IBM Blue Gene/Q supercomputer at the Argonne Leadership Computing Facility, is equipped with 786,432 cores, 768 terabytes of memory and has a peak performance of 10 petaflops. Mira's 49,152 compute nodes have a PowerPC A2 1600 MHz processor containing 16 cores, each with 4 hardware threads, running at 1.6 GHz, and 16 gigabytes of DDR3 memory. A 17th core is available for the communication library.

IBM's 5D torus interconnect configuration, with 2GB/s chip-to-chip links, connects the nodes, enabling highly efficient computation by reducing the average number of hops and latency between compute nodes. The Blue Gene/Q system also features a quad floating point unit (FPU) that can be used to execute scalar floating point instructions, four-wide SIMD instructions, or two-wide complex arithmetic SIMD instructions. This quad FPU provides higher single thread performance for some applications.

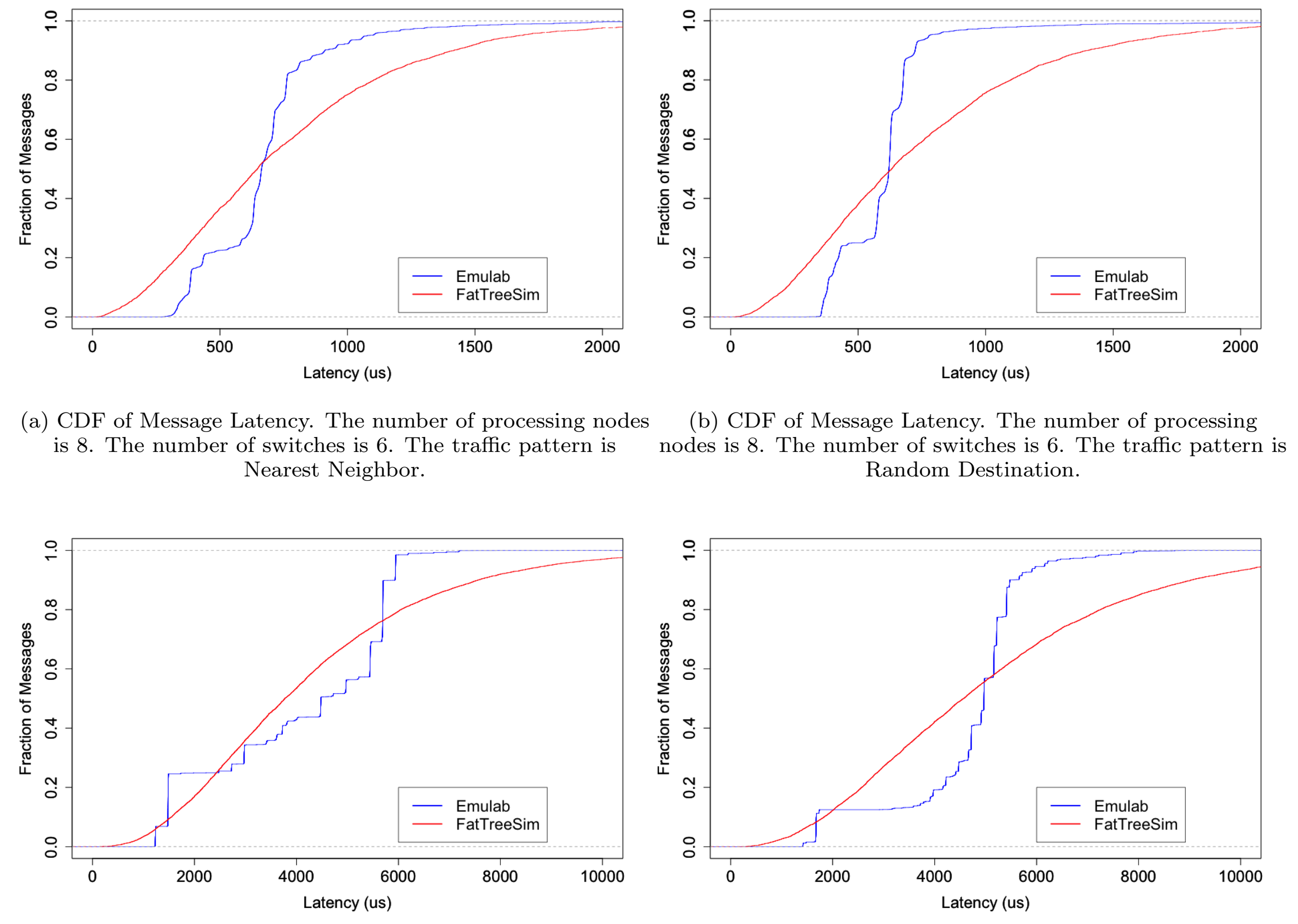
Scalability Experiments On ALCF Blue Gene/Q: Mira



(a) Packet arrival interval equals 200 ns.
(b) Packet arrival interval equals 400 ns.
(c) Packet arrival interval equals 800 ns.
(d) Packet arrival interval equals 1600 ns.
(e) Packet arrival interval equals 200 ns.
(f) Packet arrival interval equals 400 ns.
(g) Packet arrival interval equals 800 ns.
(h) Packet arrival interval equals 1600 ns.

The above figures present the FatTreeSim Scalability Experiment on Blue Gene/Q. The fat-tree model consists of 524,288 processing nodes and 20,480 switches. The total number of committed events is 567 billion. In each top subfigure, we vary the number of cores from 1,024 to 16,384 through running experiments on c1, c2, c4, c8 and c16 modes. From top-left subfigure to top-right subfigure, we vary the packet arrival interval from 200 ns to 1,600 ns. Experiments on the top subfigures are conducted using 1 Blue Gene/Q rack. In each bottom subfigure, we vary the number of cores from 2,048 to 32,768 through running experiments on c1, c2, c4, c8 and c16 modes. From top-left subfigure to top-right subfigure, we vary the packet arrival interval from 200 ns to 1600 ns. Experiments on the top subfigures are conducted using 2 Blue Gene/Q racks. The traffic pattern is random destination.

Accuracy Experiment on Emulab & Functionality Experiment on YARNsim



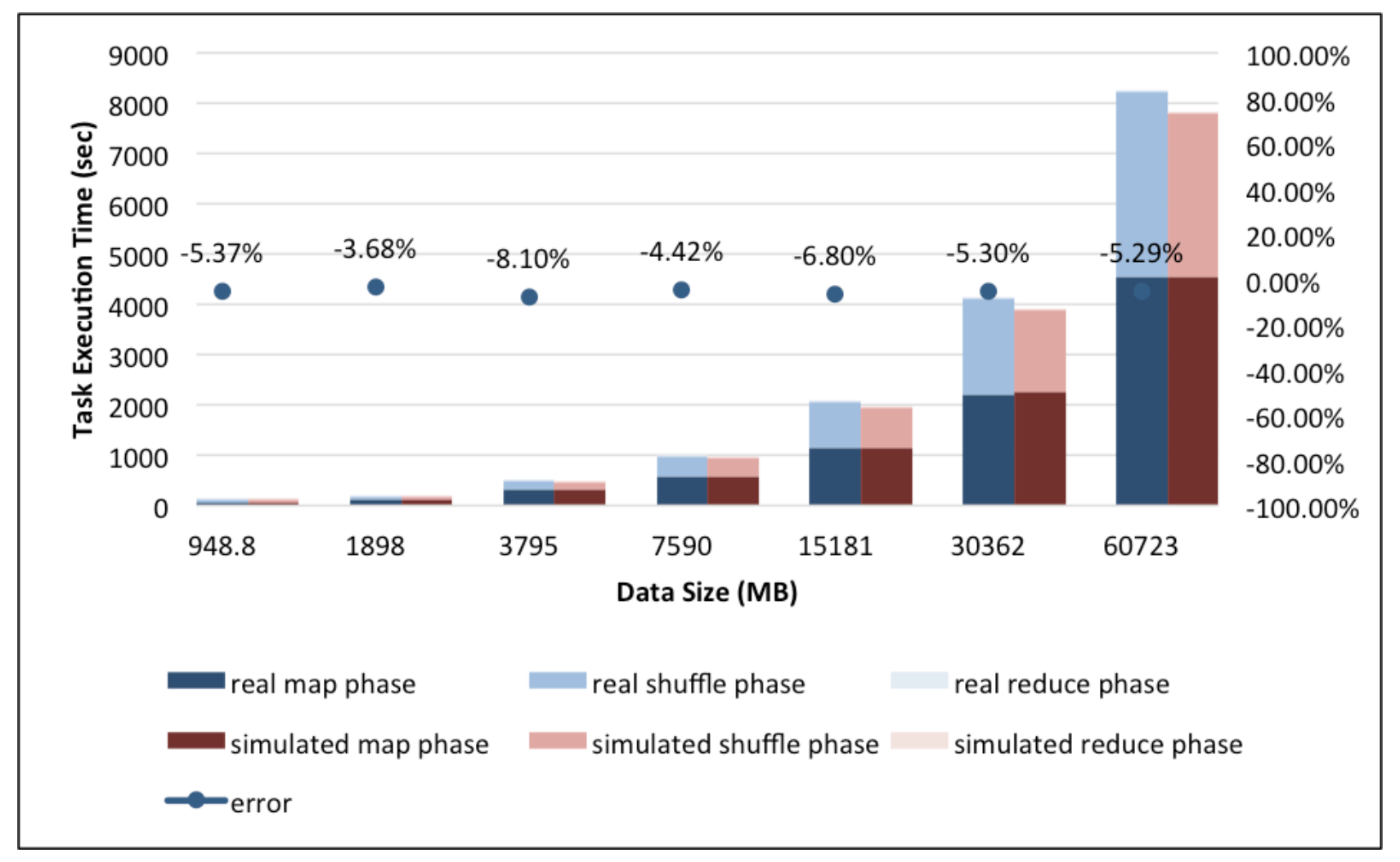
(a) CDF of Message Latency. The number of processing nodes is 8. The number of switches is 6. The traffic pattern is Nearest Neighbor.

(b) CDF of Message Latency. The number of processing nodes is 8. The number of switches is 6. The traffic pattern is Random Destination.

(c) CDF of Message Latency. The number of processing nodes is 16. The number of switches is 20. The traffic pattern is Nearest Neighbor.

(d) CDF of Message Latency. The number of processing nodes is 16. The number of switches is 20. The traffic pattern is Random Destination.

To further evaluate the accuracy of FatTreeSim, we conducted experiments on Emulab. In these test, we record the latency for each message from both the Emulab cluster and FatTreeSim and report the results in the CDF plots. We used two different configurations: a 4-port 2-tree and a 4-port 3-tree. The message size is 1,024 bytes and the number of messages is 1,000 per node. In all experiments, we observed that the curve for simulation is much smoother than the curve for Emulab. This is attributed to the fact that we model only one outgoing buffer in each outgoing port. If multiple messages are sent through this port, congestion will occur and this single point queuing effect lead to a unique waiting time for each packet.



We also conducted experiments on YARNsim[4]. YARNsim is a simulation system for Hadoop YARN. One can simulate basic Hadoop and HDFS services in YARNsim. In this experiment, we use a bio-application [5] developed in University of Delaware. The purpose of this experiment is to Demonstrate the FatTreeSim can be used by existing simulation system like YARNsim.

References

[1] https://www.olcf.ornl.gov/summit/
[2] C. Carothers, D. Bauer, and S. Pearce. ROSS: a high-performance, low memory, modular time warp system. In Fourteenth Workshop on Parallel and Distributed Simulation, 2000. PADS 2000. Proceedings, pages 53–60, 2000.
[3] J. Cope, N. Liu, S. Lang, P. Carns, C. D. Carothers, and R. Ross. CODES: Enabling co-design of multilayer exascale storage architectures. In Proceedings of the Workshop on Emerging Supercomputing Technologies (WEST), USA, June 2011.
[4] N. Liu, X. Yang, X.-H. Sun, J. Jenkins, and R. Ross. Yarnsim: Simulating hadoop yarn. In 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2015, Shenzhen, China, May 4-7, 2015, 2015.
[5] B. Zhang, D. T. Yehdego, K. L. Johnson, M.-Y. Leung, and M. Taufer. Enhancement of accuracy and efficiency for RNA secondary structure prediction by sequence segmentation and MapReduce. BMC Structural Biology, 13(Suppl 1):S3, Nov. 2013.
[6] https://www.alcf.anl.gov/user-guides/mira-cetus-vesta

Acknowledgements

ManyThanks to:
Dr. Jonathan Jenkins, Dr. Misbah Mubarak & Dr. Robert Ross
Math & Computer Science Department (MCS)
Argonne National Laboratory (ANL)
http://www.anl.gov/
Dr. Christopher Carothers
Department of Computer Science
Rensselaer Polytechnic Institute (RPI)
http://www.cs.rpi.edu/
and
Argonne Leadership Computing Facility (ALCF)
https://www.alcf.anl.gov/

Contact Information

Department of Computer Science
Illinois Institute of Technology, Chicago
URL: http://www.cs.iit.edu/
Email: nliu8@hawk.iit.edu
dong.jin@iit.edu