Evaluating Adaptive Routing Performance on Large Scale Megafly Topology

Md Nahid Newaz
Department of Computer Science and Informatics
Oakland University
Michigan, USA
mdnahidnewaz@oakland.edu

Md Atiqul Mollah
Department of Computer Science and Informatics
Oakland University
Michigan, USA
mollah@oakland.edu

Peyman Faizian
Department of Computer Science
Florida State University
Florida, USA
faizian@cs.fsu.edu

Zhou Tong
Department of Computer Science
Wheaton college
Massachusetts, USA
tong_tony@wheatoncollege.edu

Extended Abstract

Interconnection network is a key major component of High Performance Computing(HPC) clusters and supercomputers. To construct HPC systems of exascale computing capacity and beyond, thousands of computing nodes need to be interconnected with a network topology that can deliver high throughput and low latency at a large scale. To design such scalable high performance interconnect in a cost-effective way, many new network topologies have been proposed in recent years as alternatives to the prevalent yet expensive fat-tree based interconnect designs. Among those, the Dragonfly topology, first introduced by Kim et al. [1], is particularly well-known for its scalability and low diameter. Dragonfly-based interconnects have been used to construct several high-performance clusters in the current Top500 [2] list of the world's fastest supercomputers such as Piz Daint [3] and Trinity [4].

More recently, a new variation of Dragonfly named Megafly, and also known as Dragonfly+, has been introduced and successfully deployed to construct Niagara [5] the fastest supercomputer accessible for academic use in Canada. Figure-1 represents one of many possible variations of Megafly topology. Megafly offers an even higher scalability and better path diversity than the canonical Dragonfly design and thus, is a prominent candidate for use in future supercomputing interconnects. Similar to Dragonfly, Megafly topology heavily relies on the load balancing features of adaptive routing among groups of routers to maximize its throughput performance. However, the practical adaptive routing algorithms of present day, namely Universal globally Adaptive Load-Balance(UGAL) Routing [6] and Progressive Adaptive Routing(PAR) [7], do not leverage important properties specific to the Megafly topology, such as **path distribution** and **minimal path diversity**. As a result, these algorithms often fall short of a robust performance on Megafly interconnects. For example, The UGAL algorithm on Megafly suffers from limited congestion detection capability that leads to inaccurate and thus performance-limiting adaptive routing decisions. The PAR algorithm, on the other hand, may offer limited load balancing capability despite its accuracy to detect congestion on Megafly networks.

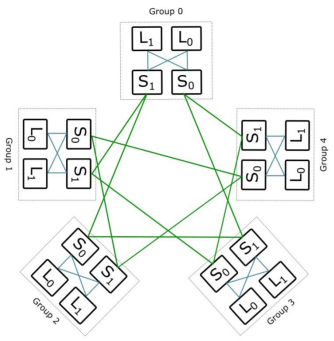

Fig. 1. Megafly Topology with 2 routers in each level of a group and total 5 groups that is mfly(2,5).

In this research poster, we present experimental results to demonstrate that the throughput performance of PAR - currently best known routing scheme for Megafly - is not optimal across all variations of scale and wiring schemes. More specifically, we show that the performance of PAR significantly degrades with the increase in the number of groups in the Megafly network when each spine router of a Megafly group can no longer be directly connected to all other router groups. This, considerably limits the scale at which Megafly can be built and run efficiently with PAR. To overcome such limitations, we propose two new routing algorithms KAPR and KU-GCN, both inspired from Piggyback Routing first proposed in [7]. To the best of our knowledge, this is the first work dedicated to the development of new adaptive routing schemes specifically for Megafly.

To test the performance benefits of our new routing schemes on Megafly topology, we evaluate their performance through trace-driven simulations of parallel MPI application workloads. We choose SST-Macro [8] version 9.0, an open-source coarse-grain parallel discrete event simulator as our simulation platform due to its popularity and suitability to simulate large scale distributed architectures. SST-macro supports offline mode replay of MPI communication traces on a variety of built-in interconnects including Megafly(referred as Dragonfly Plus in SST-Macro). We extend the existing Megafly implementation by adding support for wiring schemes and our new routing schemes. We use SSTMacro to replay a variety of NAS parallel benchmarks and synthetic application traces and measure the corresponding cumulative communication time of all MPI ranks.

In our evaluation we used three variations of Megafly topologies. The following Megafly topologies are used in our evaluation:

- *mfly(12,9):* A high connectivity topology made of 9 groups of radix-24 routers and 1,296 nodes. Each group has 24 routers(12 leaf and 12 spine) and 144 global links outgoing to the remaining 8 groups. The minimal path diversity is 18.
- *mfly(10,11):* A medium connectivity topology made of 11 groups of radix-20 routers and 1,100 nodes. Each group has 10 leaf routers, 10 spine routers and 100 global links outgoing to the remaining 10 groups. The minimal path diversity is 10.
- *mfly(8,17):* A low connectivity topology made of 17 groups of radix-16 routers and 1,088 nodes. Each group has 8+8=16 routers and 64 global links connected with the remaining 16 groups. The minimal path diversity is 4.

We note that despite having similar network sizes, the three Megafly topologies we consider have very different network degrees. The scale of the topologies are determined relative to the largest possible topology sizes for their corresponding network degrees. Our respective labeling of these three topologies as small-, medium- and large-size is consistent with the Dragonfly+ topology definition given by Shpiner et al. [9].

Our evaluation results for synthetic traces shows that for random permutation traffic our KAPR and KUGCN outperform PAR by at least 4%, 5.4% and 13% on *mfly(12,9)*, *mfly(10,11)* and *mfly(8,17)* respectively. On adversarial Shift traffic, KU-GCN performs strictly better than PAR on all three topologies with a performance gain ranging from 1.4% on *mfly(12,9)* to 11% on *mfly(10,11)* and as high as 30% on *mfly(8,17)*.

In NAS parallel benchmark traces once again our KU-GCN routing performs at least on par with PAR on *mfly(12,9)* and *mfly(10,11)* and it shows significant performance gains on *mfly(8,17)* by margins of 18.5% (IS), 32.5% (CG), 8.3% (MG), 43.5% (FT) and 2.3% (BT). The performance of KAPR, on average, is 3.4% slower than PAR on *mfly(12,9)* and *mfly(10,11)* and 5.2% faster than PAR on *mfly(8,17)*. The performance difference between KAPR and PAR on *mfly(8,17)* shows high variability across the benchmarks.

Finally, our proposed routing scheme requires global congestion information notifications which is subject to latency from the queuing, transmission and propagation of piggy-backed/congestion notification packets. To observe how network latencies affect the routing performances of KAPR and KU-GCN, we introduce a fixed latency in our simulations between the time of congestion events and the time when such events are perceived at the source router. Our evaluation shows that our routing schemes can tolerate network delays of up to and beyond 100 microseconds and still outperform PAR. This make our routing schemes easily compatible with practical HPC interconnect fabrics where it is commonplace to have end-to-end network latencies below one microsecond [10,11].

In summary, in this research poster we introduced two new adaptive routing schemes that can improve the communication performances of any Megafly based interconnects. The new routing schemes perform significantly better than the currently widely adopted progressive adaptive routing scheme in large scale and/or connectivity constrained Megafly designs.

REFERENCES

[1] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in Proceedings of the 35th Annual International Symposium on Computer Architecture, ISCA '08, (Washington, DC, USA), pp. 77–88, IEEE Computer Society, 2008.

[2] "Top500 supercomputers sites." http://www.top500.org.

[3] Swiss National Supercomputing Centre, "Piz Daint— CSCS." https://www.cscs.ch/computers/piz-daint/.

[4] B. J. Archer and M. Vigil, "The Trinity system," in Nuclear Explosive Code Development Conference (NECDC), (Los Alamos, New Mexico), Oct. 20–24, 2014. Also appears as Los Alamos Technical Report LA-UR-15-20221.

[5] M. Ponce, R. van Zon, S. Northrup, D. Gruner, J. Chen, F. Ertinaz, A. Fedoseev, L. Groer, F. Mao, B. C. Mundim, M. Nolta, J. Pinto, M. Saldarriaga, V. Slavnic, E. Spence, C.-H. Yu, and W. R. Peltier, "Deploying a top-100 supercomputer for large parallel workloads: The niagara supercomputer," in Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning), PEARC '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[6] A. Singh, Load-Balanced Routing In Interconnection Networks. PhD thesis, Stanford University, 2005.

[7] N. Jiang, J. Kim, and W. J. Dally, "Indirect adaptive routing on large scale interconnection networks," in Proceedings of the 36th Annual International Symposium on Computer Architecture, ISCA '09, (New York, NY, USA), pp. 220–231, ACM, 2009.

[8] H. Adalsteinsson, S. Cranford, D. A. Evensky, J. P. Kenny, J. Mayo, A. Pinar, and C. L. Janssen, "A simulator for large-scale parallel computer architectures," Int. J. Distrib. Syst. Technol., vol. 1, pp. 57–73, Apr. 2010.

[9] A. Shpiner, Z. Haramaty, S. Eliad, V. Zdornov, B. Gafni, and E. Zahavi, "Dragonfly+: Low cost topology for scaling datacenters," in 2017 IEEE 3rd International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era (HiPINEB), pp. 1–8, IEEE, 2017.

[10] R. Zambre, M. Grodowitz, A. Chandramowlishwaran, and P. Shamis, "Breaking band: A breakdown of high-performance communication," in Proceedings of the 48th International Conference on Parallel Processing, ICPP 2019, (New York, NY, USA), Association for Computing Machinery, 2019.

[11] P. Knebel, D. Berkram, A. Davis, D. Emmot, P. Faraboschi, and G. Gostin, "Gen-z chipsetfor exascale fabrics," in 2019 IEEE Hot Chips 31 Symposium (HCS), pp. 1–22, 2019.