Achieving the Performance of Global Adaptive Routing using Local Information on Dragonfly through Deep Learning

Ram Sharan Chaulagain* †, Fatema Tabassum Liza*, Sudheer Chunduri†, Xin Yuan*, Michael Lang‡,
*Florida State University, †Argonne National Laboratory, ‡Los Alamos National Laboratory
*{chaulaga, liza, xyuan}@cs.fsu.edu, †sudheer@anl.gov, ‡mlang@lanl.gov

***Abstract*—The Universal Globally Adaptive Load-balance Routing (UGAL) with global information, referred as UGAL-G, represents an ideal form of adaptive routing on Dragonfly. However, UGAL-G is impractical to implement since the global information cannot be maintained accurately. Practical adaptive routing schemes, such as UGAL with local information (UGAL-L), performs noticeably worse than UGAL-G. In this work, we investigate a machine learning approach for routing on Dragonfly. Specifically, we develop a machine learning based routing scheme, called UGAL-ML, that is capable of making routing decisions like UGAL-G based only on the information local to each router. Our preliminary evaluation indicates that UGAL-ML can achieve comparable performance to UGAL-G for some traffic patterns.**

I. Introduction

The Dragonfly network has been deployed in the Cray Cascade architecture and also proposed for the next generation Supercomputers [1]–[3]. The universal globally adaptive routing (UGAL) is the state-of-the-art routing scheme for Dragonfly [1]. It is well known that UGAL with global information (UGAL-G) which assumes each router has queue occupancy knowledge of all other routers of network can achieve high routing performance. However, UGAL-G is not practical to implement in real systems since global information cannot be maintained accurately. Practical routing schemes such as UGAL-L use local router queue information to make routing decisions [4]. The performance of UGAL-L is not as high as UGAL-G. It is thus desirable to develop a routing scheme that can achieve the performance of UGAL-G using only the local information.

Recent work on traffic-pattern aware routing (TPR) has shown that using a UGAL-like routing scheme, link-usage statistics counters and local port occupancy of router can infer the global information such as the traffic pattern [5]. In other words, local information can infer global information of network. Motivated by the results in ref [5], we seek to develop a scheme that makes routing decisions using local information, but can achieve the performance of UGAL-G. The idea is to train a deep neural network with local information maintained by a router and UGAL-G's routing decisions and use the trained model to make routing decisions. This new routing scheme is called UGAL-ML. We evaluate UGAL-ML using uniform and adversarial traffic patterns on a Dragonfly network. Our initial results show that UGAL-ML performs better than UGAL-L and almost matches the performance of UGAL-G for these traffic patterns, which demonstrates that this is a promising scheme.

II. Methodology

This research consists of three sub-tasks: (1) collecting the counters maintained by router (local information) with UGAL-G's routing decision , (2) training a neural network model for each router, and (3) using trained models to drive routing function of routers and evaluate the performance.

A. Data Collection

UGAL makes a choice between a minimal path or a randomly selected non-minimal path (also called VLB path). The goal is to train a neural network model so that it will make similar decision as UGAL-G by only examining local information. The data is collected using the Booksim 2.0 [6], a flit accurate network simulator. The simulator runs UGAL-G, and we augment the simulator to store the local information as well as UGAL-G's routing decision for each packet. The local information stored include the port occupancy for minimal path, the port occupancy for non-minimal path, the hop count for minimal path, the hop count for the non-minimal path, and the following link-usage counters which are updated over 50 cycles sliding window period.

- **DestC**$_i$: Counter that records number of packets sent to router i inside same group.
- **Port_thr**$_i$: Counter that records number of through packets, packets originated from other routers in the same group, passing through port i inside same group.
- **DestGrpC**$_g$: Counter that records number of packets sent to a different group g.
- **Thr_GrpC**: Counter that records number of through packets, packets originated form other routers, sent to a different group g.

Details of these counters can be found in ref [5]. For each UGAL-G routing decision, the counters for the destination router and/or groups, are stored.

We collect data for two traffic patterns: uniform random traffic and adversarial shift traffic at different injection loads.

B. Neural Network architecture and Training

Fully connected Neural Network also called Dense Neural Network (DNN) is used for training the router counters and routing decision. We have explored many other networks, and observed a DNN with 10 hidden layers gives high accuracy.

The hidden layers used ReLU and the output layer used softmax as the activation function. Also, categorical cross entropy is utilized for the loss function. The optimizer used for the model is Adam optimizer with lr=0.0001. The neural network is trained to predict path selection as routing decision given router counter values.

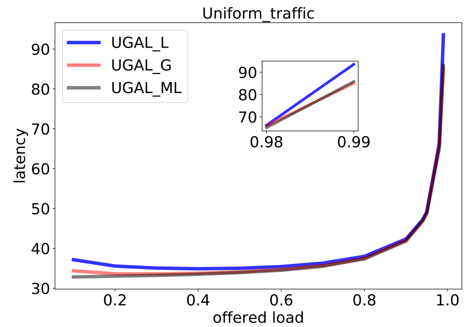
The average accuracy of each router models is around 88% to 91% when trained with both adversarial and uniform random traffic data combined. 10-fold cross validation is used to estimate accuracy of neural network models.

C. Routing implementation in Booksim

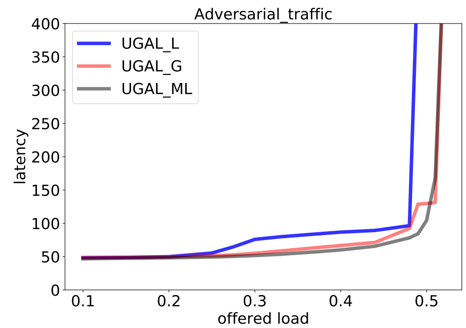
We incorporate UGAL-ML in Booksim 2.0, which is written in C++. To use deep learning models, we have used a library called frugally-deep [7]. Frugally-deep library converts keras saved model files into json file. Booksim reads those json files of each router and later use them to predict routing decisions. UGAL-ML is shown in Algorithm 1.

Algorithm 1 UGAL-ML routing Algorithm

1. Load DNN models(json files) of each router to booksim
2. When a packet arrives, source router calculates min path and non min path.
3. Source router passes the snapshot of local counters to its corresponding model, which returns routing decision to select min path or non-min path.



(a) Latency Curve of Uniform Traffic Pattern



(b) Latency Curve of Adversarial Traffic Pattern

III. Preliminary Results

BookSim 2.0 [6], is used to simulate the Dragonfly network. We have selected a fully connected Dragonfly network with g=9, a=4 ,k=7, and h=2 parameters with total 36 routers, 72 terminals and 9 groups. However, UGAL-ML is independent to topology scale as it solely rely on router local counters. The routing algorithms are evaluated using uniform and adversarial shift traffic patterns.

Figure 1a shows the packet latency for the uniform traffic pattern at different traffic loads. At low load, UGAL-ML has lower latency than UGAL-G, which in turn has lower latency than than UGAL-L. At high load, all three schemes have similar saturation throughput. Figure 1b shows the packet latency for the adversarial traffic at different traffic loads. As can be seen in the figure, UGAL-ML and UGAL-G have a similar saturation throughput, which is higher than that for UGAL-L.

IV. Conclusion and Future Work

Our preliminary results indicate that deep neural networks have the potential to make effective routing decisions using local information on Dragonfly. In the future, we plan to train UGAL-ML for other traffic patterns and other Dragonfly topologies.

V. Acknowledgment

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357 and computing resources supported by the National Science Foundation under Grants CRI-1822737, and the XSEDE Bridges resource at the Pittsburgh Supercomputing Center (PSC) through allocation TG-ECS190004.

References

[1] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *2008 International Symposium on Computer Architecture*. IEEE, 2008, pp. 77–88.

[2] D. D. Sensi, S. D. Girolamo, K. H. McMahon, D. Roweth, and T. Hoefler, "An in-depth analysis of the slingshot interconnect," arXiv.2008.08886, 2020.

[3] Z. S. A. Alzaid, S. Bhowmik, X. Yuan, and M. Lang, "Global link arrangement for practical dragonfly," in *Proceedings of the 34th ACM International Conference on Supercomputing*, ser. ICS '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3392717.3392756

[4] M. S. Rahman, S. Bhowmik, Y. Ryasnianskiy, X. Yuan, and M. Lang, "Topology-custom ugal routing on dragonfly," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–15.

[5] P. Faizian, J. F. Alfaro, M. S. Rahman, M. A. Mollah, X. Yuan, S. Pakin, and M. Lang, "Tpr: Traffic pattern-based adaptive routing for dragonfly networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 931–943, 2018.

[6] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. E. Shaw, J. Kim, and W. J. Dally, "A detailed and flexible cycle-accurate network-on-chip simulator," in *2013 IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS)*, April 2013, pp. 86–96.

[7] Dobiasd, "Dobiasd/frugally-deep," August 2020. [Online]. Available: https://github.com/Dobiasd/frugally-deep