Exploring Randomized Multipath Routing on Multi-Dimensional Torus Networks

Prajakt Shastry[1], Daniel Parker[2], Sanjiv Kapoor[1], Ioan Raicu[1,3]
Illinois Institute of Technology[1], University of Chicago[2], Argonne National Laboratory[3]
pshastry@hawk.iit.edu, dkparker@uchicago.edu, kapoor@iit.edu, iraicu@cs.iit.edu

***Abstract*— Network performance is a critical aspect of high-performance computers, and improving its performance is a major goal in the design of future systems; this work proposes to improve network performance through new routing algorithms, leveraging the rich multi-path topologies of multi-dimensional torus networks commonly found in supercomputers built in the past fifteen years. Virtually all torus networks in production today utilize the dimension order routing algorithm, which is essentially a static and deterministic routing strategy to allow internode communication. This static routing strategy has significant load balancing implications, leading to sub-par performance. We propose a new Random Distance Routing algorithm, which randomly distributes packets to different neighboring nodes that are closer to the destination, leading to global load balanced network. Through the CODES/ROSS [4] simulator, we show that the proposed randomized multi-path routing algorithm can increases throughput of a 5D-Torus network by 1.6X, as well as reduce latency by 40%.**

Keywords—torus networks; routing; multipath routing; dimensional order routing ; Codes; ROSS

I. Introduction

Many supercomputers use the Torus topology [1] based interconnect as the network fabric. Torus topologies are like a controlled mesh, where the edges wrap around and connect to a node on the other side of the network. Torus networks are known to have a low latency for communicating with their nearest neighbours. In a 3D torus network, each node is connected to 6 neighbours (Fig. 1), while in a 5D torus network, each node is connected to 10 neighbours.

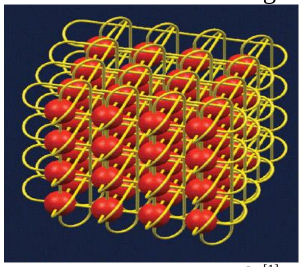

Fig 1. 3D torus network [1]

Most of the scientific applications use MPI for communication between the nodes. Numerous MPI implementations given by manufacturers of supercomputers use Dimension Order Routing, where dimensions are generally ordered longest first to shortest [2]. Dimensional Order Routing, being simple to implement gives a good average case performance, but cannot load balance well in case of heavy loads. Valiant routing [3] overcomes this by sending packets to a random intermediary node, which routes to the destination sending whole messages through that one fixed intermediary node. This routing can reducing hot spots, but looses locality of the nodes.

We introduce a simplistic randomized multipath algorithm for multi-dimensional torus networks called RDR – Random Distance Routing. RDR tries to improve performance of network by making local decisions to route the message leading to a globally load balanced network. RDR deploys multipath routing, distributing packets of the same message to different neighbouring nodes.

Fig. 2, Fig. 3, and Fig. 4 depict the hot spots in the network of a 10x10x10 torus network, where half the nodes are paired , at random, to send data to each of the corresponding other half of the nodes. The size and shade of the coloured dot represents the number of packets utilizing that link. A bigger and darker circle means more load. Our experiments illustrate the almost uniform utilization of links using RDR (right), where we get improved load balancing as compared to DOR (Fig. 2) and Valiant routing (Fig. 3).

II. Methodology

For evaluating the proposed algorithm, we used the CODES and ROSS framework at up to 20000 node scales. Workloads were synthetically generated, where half the nodes were sending, while the other half the nodes were receiving. Message size of 8KB and packet size is 512 bytes. Our algorithm, RDR, partitions the message into packets which are independently sent to the destination using the method RDR(V) at each node V that receives the incoming packet.

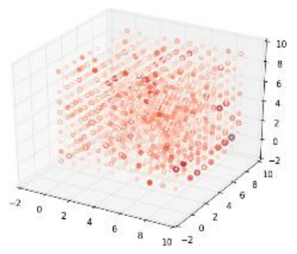

Fig 2. 3D torus with DOR

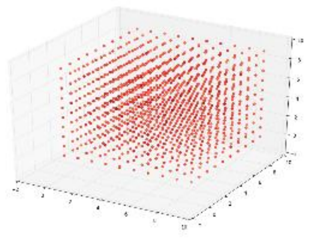

Fig 3. 3D torus with Valiant

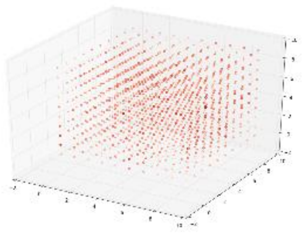

Fig 4. 3D torus with RDR

```
RDR(V):
    1. If V = destination: - Stop
    2. Mark neighbouring node W as viable if distance from W to destination is less
    3. Randomly select one of viable options. Send packet to selected option, to process using RDR (W)
```

III. Performance Evaluation

Fig. 5 shows that RDR gives a better link utilization while reducing the number of packets per link . DOR does not utilize all the links, and most packets are concentrated on some links on network. Valiant does distribute well, but as its packets travel a longer path, number of packets on link remain high..

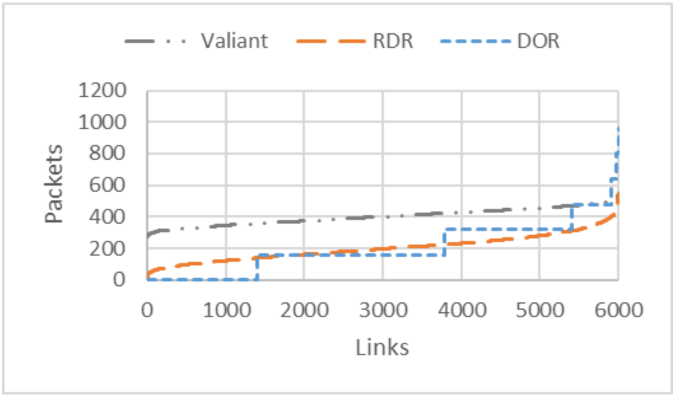

Fig 5. Link Utilization in 3D torus.

Fig. 6 shows that RDR performs better than DOR for all nodes, with 50% as an average improvement. Valiant routing gives worse latency than DOR, because valiant loses locality, by routing to random intermediary node.

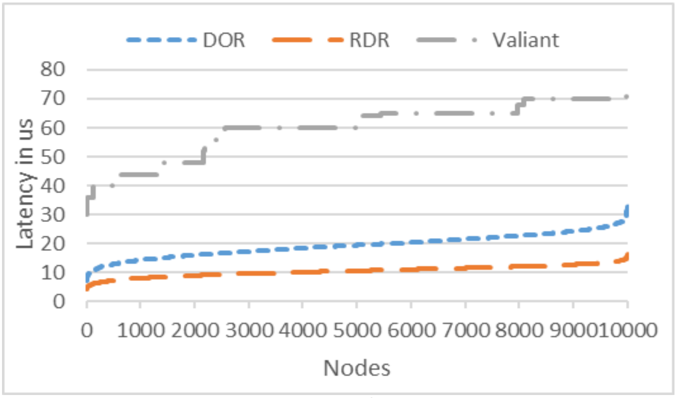

Fig. 6 Latency CDF plot for 5D torus nodes.

In Fig. 7 we illustrate the maximum throughput using RDR. Valiant has an average throughput distributed throughout all the nodes. As the links' bandwidth is 2 GB, RDR comes closest to achieving maximum throughput.

One potential negative aspect of RDR as compared to DOR and Valiant is the likelihood for out of order packets. However, we show that even if packets are delayed until all received packets are delivered to the application layer in the correct order, the overall latency can be lower compared to DOR.

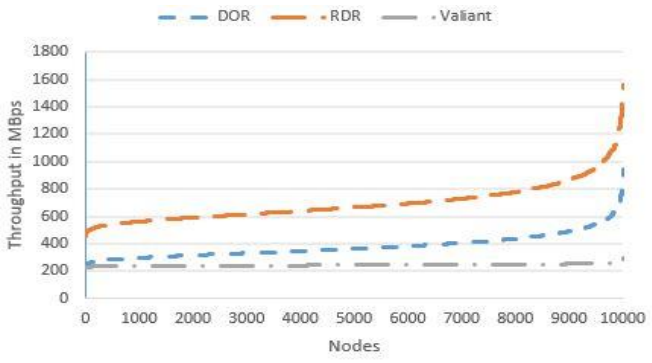

Fig. 7 Throughput CDF for all 5D torus nodes

IV. Related Work

There has been much research in designing routing protocols for torus networks. GOAL [8] adaptively distributes traffic using probability distribution. GOAL seems to reduce latency when compared with DOR by 40% in the worst case, but our data shows RDR can reduce latency by 50% on average. Other algorithms are RLB [6], Chaos [7], and ROMM [5]. All of these algorithms try to tackle the same set of problems, namely load balancing and deadlock avoidance. RDR gives an easy to implement solution that load balances well, which greatly increases the performance of the system, while addressing deadlocks using two virtual channels.

V. Conclusion And Future Work

RDR increases performance of the network by randomly distributing data, and achieves a 1.6X improvement in average case throughput and 50% reduction in latency when compared with DOR in the average case. RDR scales well with increasing number of nodes in the system.

We plan to do an extensive comparison between GOAL and RDR in future work. We also aim to improve on RDR by using network state for making routing decisions.

VI. References

[1] 3D Torus Network from the IBM BlueGene/L Supercomputer; https://asc.llnl.gov/computing_resources/bluegenel/images/torus.jpg, 2016

[2] J. Breckling, Ed., Looking Under the Hood of the IBM Blue Gene/Q Network, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3] L. G. Valiant. "A scheme for fast parallel communication." SIAM Journal on Computing, 11(2):350–361, 1982.

[4] Cope, Jason, Ning Liu, Sam Lang, Phil Carns, Chris Carothers, and Robert Ross. "Codes: Enabling co-design of multilayer exascale storage architectures." In Proceedings of the Workshop on Emerging Supercomputing Technologies, pp. 303-312. 2011.

[5] T. Nesson and S. L. Johnsson. ROMM routing on mesh and torus networks. In Proc. 7th Annual ACM Symposium on Parallel Algorithms and Architectures SPAA'95, pages 275– 287, Santa Barbara, California, 1995.

[6] A. Singh,W. J. Dally, B. Towles, and A. K. Gupta. Locality preserving randomized routing on torus networks. In *Proc.* 12th Annual ACM Symposium on Parallel Algorithms and *Architectures SPAA'02*, Winnipeg, Canada, 2002.

[7] Bolding, M. L. Fulgham, and L. Snyder. The case for chaotic adaptive routing. IEEE Transactions on Computers, 46(12):1281–1291, 1997

[8] Arjun Singh and William J Dally and Amit K Gupta and Brian Towles. GOAL: A load-balanced adaptive routing algorithm for torus networks. International Symposium on Computer Architecture (ISCA) ACM2003