Topologic-aware Allocation Policies for Jellyfish

Jose A. Pascual, Javier Navaridas, Alejandro Erickson and Ian

ABSTRACT

1. INTRODUCTION

Jellyfish topology [1] has been recently proposed as a high bandwidth and low latency interconnect for large scale data centers and HPC systems. In opposite to recenctly proposed server-centric datacenter networks (DCN), jellyfish is an indirect (switch-centric) network in which the servers are connected to the switches. This network is a *degree-bounded* random regular graph (RRG) among the top-of-rack (ToR) switches, in which all the nodes have the same degree, are bidirectional and are connected randomly. RRGs also provide other desiderable properties for an interconnect such as low diameter, high connectivity among nodes and easy incremental expansion.

As stated in the original paper [1], routing in jellyfish is a challenge. Although jellyfish provides high connectivity among switches, classical routing policies are not able to exploit the path diversity offered. In that work the authors evaluated two well-known routing policies, shortest path (SP) and Equal-cost Multi-path (ECMP), assesing that the use of the shortest paths does not provide enough path diversity to utilize the full capacity of the network. This issue was solved using the K-Shortest Path (KSP) [2] routing policy that uses more paths at the cost of being longer. Although KSP perfoms well compared to SP and ECMP, the author in [3] showed that jellyfish has several features that make it ineffective. In particular they stated the possibly large number of source-destinations (SD) pairs that will share the same K shortest-paths and the random number of short paths between each pair of switches.

These works have studied jellyfish both theoretically, puting bounds to topological properties, and empirically evaluating the performance of several communication patterns. However none of them have consider the natural scenario in which this topology could be used: data centers or HPC centers where many applications run concurrently. To the best of our knowledge, there is no work devoted to evaluate the performance of such applications in this topology.

The assignemnt of resources to application has been widely studied in the context of HPC. Those works clearly differenciate three stages: selection of the application to be executed, allocation of the resources to that application and mapping of the taks that compose the application to the physical servers.

The objective of this work is the evaluation of the performance of these routing policies in multi-application scenarios.

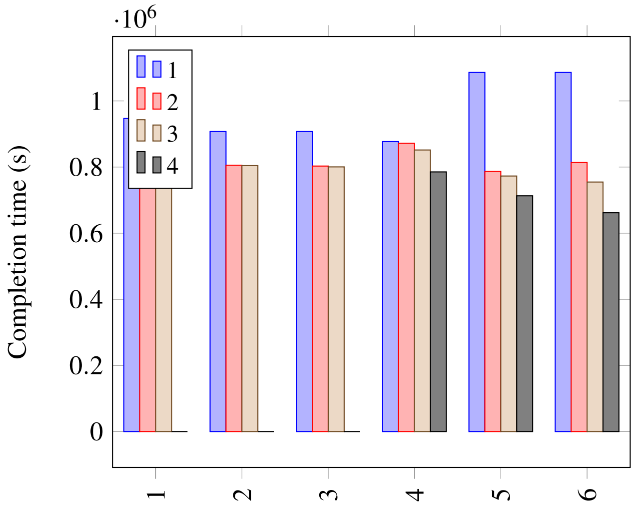

Figure 1: Completion time in second employed to process three traffic patterns comparing.

The rest of the paper is organized as follow. Section 2

TODO: Write oraganization

2. PRELIMINARIES

3. ALLOCATION IN JELLYFISH

When an application is submitted to be executed, the allocator is in charge of finding the appropiate set of resources to place it. In this section we analize the allocation strategies used in [3].

- Sequential:
- Random:
- : Trace-based allocation:

4. LOCALITY AND CONTIGUITY IN JELLYFISH

5. EXPERIMENTAL SET-UP

6. ANALYSIS OF THE RESULTS

7. CONCLUSIOSN ANS FUTURE WORK

8. REFERENCES

[1] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, (Berkeley, CA, USA), pp. 225–238, USENIX Association, 2012.

[2] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[3] X. Yuan, S. Mahapatra, W. Nienaber, S. Pakin, and M. Lang, "A new routing scheme for jellyfish and its performance with hpc workloads," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, (New York, NY, USA), pp. 36:1–36:11, ACM, 2013.