An Analysis of Performance Variability on Dragonfly+ Topology

1st Majid Salimi Beni
Department of Computer Science
University of Salerno
Salerno, Italy
msalimibeni@unisa.it

2nd Biagio Cosenza
Department of Computer Science
University of Salerno
Salerno, Italy
bcosenza@unisa.it

***Abstract*—Large-scale compute clusters are highly affected by performance variability that originates from different sources. Among these sources, the network plays an essential role as a shared resource between users and their jobs in a supercomputer. In this paper, we analyze the effect of some network-related sources on the performance variability of a modern compute cluster equipped with a Dragonfly+ interconnect. Specifically, we focus on the impacts of job placement, communication patterns, routing strategy, and network background traffic on the performance variability of communication-intensive workloads.**

To quantify the effect of network congestion (background traffic) on the performance variability, we propose a heuristic that can successfully estimate the amount of communication on the network produced by other jobs running on the cluster simultaneously. Then, we show how this network congestion contributes to the performance variability of different communication patterns and real-world communication-intensive applications.

***Index Terms*—performance variability, topology, dragonfly+**

I. Introduction

Despite the massive development of modern compute clusters, there is still a large gap between computation and communication, highlighting the significance of optimizing data transmission over the network. Hence, modern network interconnects are required to deliver higher bandwidth and lower latency. An example of such interconnect is Dragonfly+ which provides high network utilization, scalability, and router buffer utilization in comparison to Dragonfly [3]. However, Dragonfly+ also suffers performance variability despite its improvements compared to its predecessors. This performance variability originating from different sources such as OS and MPI overheads, I/O, network congestion, etc., degrades the application and system performance and negatively affects the batch scheduler decision-makings. Accordingly, it is essential that we identify the sources of this variability, try quantifying them, and then mitigate the variance in the performance.

Since Different users use large-scale compute clusters simultaneously, some resources are shared among them. Between these shared resources, the network elements, including links and routers, which may be affected by contention, are identified as the main source of performance variability [2]. Although a part of this performance variability comes from sources like I/O, we focus on network congestion which is the main player. This paper conducts a performance variability analysis on a large-scale compute cluster equipped with a Dragonfly+ topology. For this purpose, we analyze several factors contributing to performance variability, such as communications patterns, message size, node allocation locality, and network background traffic. Further, we propose a simple heuristic to estimate the network background traffic and show how it contributes to the performance variability.

It should be noted that all the experiments are performed on Marconi100 supercomputer [1], with IBM Power System AC922 in which 980 nodes are connected through a Dragonfly+ topology. Each node has 16 cores at 2.6 GHz and 4 NVIDIA Volta V100 GPUs, and per-node memory is 256 GB. The Spectrum-MPI 10.4 is installed on the cluster, SLURM is the workload manager, and Adaptive routing is the default routing strategy. In all experiments, we allocate 16 nodes and assign one process to each node: 1 job with 16 processes. Notably, each experiment is repeated 1000 times, and the best distribution with the minimum error among 100 fitted distributions has been shown. Moreover, the background traffic is real-world and generated by the cluster's other users.

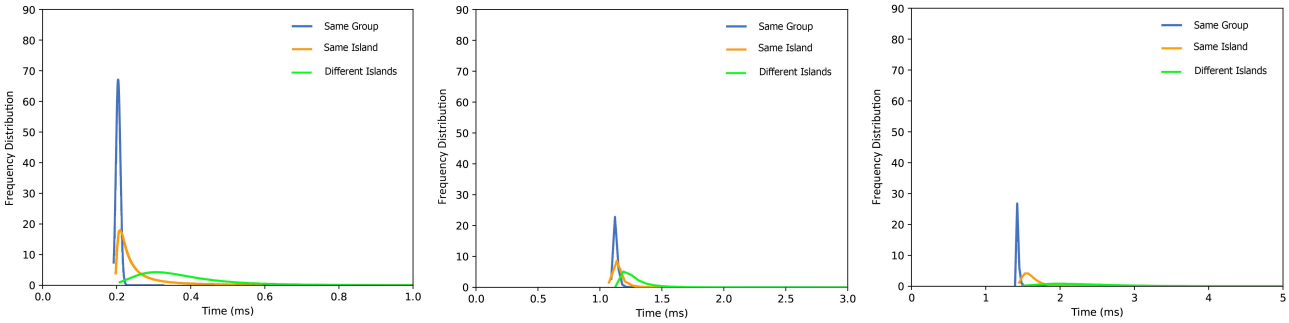
II. Node allocation locality and communication patterns

According to the topology of Dragonfly+, we define three node allocation locality hierarchies: 1. all nodes on the same group, 2. all nodes on the same island, and 3. all nodes on different islands, which the locality decreases consecutively. Figure 1 shows the distribution of 1000 iterations of performing three collective communications: MPI_Bcast, MPI_Reduce, and MPI_Alltoall, which are representatives of one-to-all, all-to-one, and all-to-all patterns. As shown, for three collectives, the node allocation on different islands has the longest distribution and therefore has the most variable performance, while allocating nodes more locally (on the Same group) shows the least variability in the performance. Moreover, Alltoall shows the worst distribution (most performance variability) when the nodes are distributed on different islands. It should also be noted that the network background congestion affects the less-local links more than local ones, which is why the different island's distribution is more long-tailed (skewed) and flattened.

III. Background traffic estimation

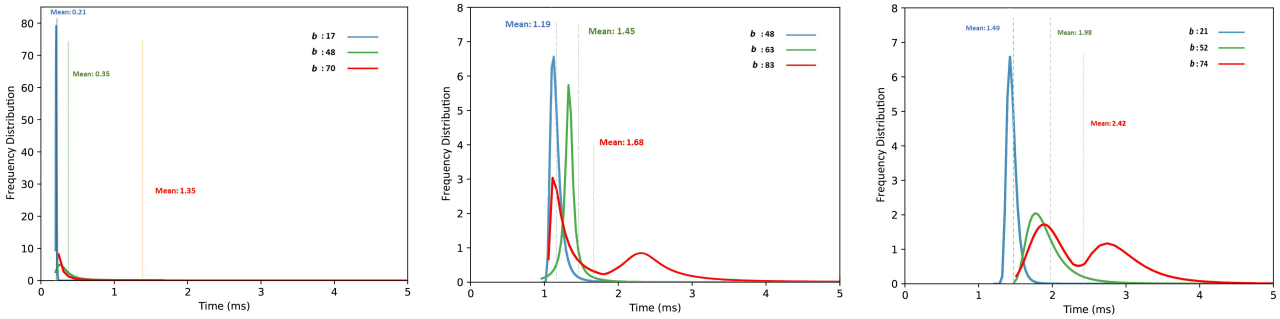
Since network congestion (network background traffic) is an essential factor contributing to performance variability, we propose a heuristic that can estimate the current background network activity based on the available information from the job scheduler. For this purpose, we present the following heuristic, in which b is an estimation of background traffic and

Authors' preprint
Not for redistribution. The definitive Version was published at 2022 IEEE International Conference on Cluster Computing (CLUSTER)
DOI: 10.1109/CLUSTER51413.2022.00061



(a) Broadcast (b) Reduce (c) Alltoall

Fig. 1: Distribution of 1000 iterations of collectives when nodes are allocated on different locality levels



(a) Broadcast (b) Reduce (c) Alltoall

Fig. 2: Distribution of 1000 iterations of collectives under different background traffics

is between 0 and 100. The bigger the b, the more probable the network is congested.

$$b = \frac{N_c}{N_t} * \frac{N'_c}{N_a} * 100 \tag{1}$$

In which N'_c is the number of nodes that contribute to communication, N_c is the number of unique nodes which are involved in communication, N_a refers to all of the allocated running nodes, and N_t is the total number of physical cluster nodes (980 in Marconi100). We distinguish between the nodes contributing to communication and the unique nodes contributing to communication since some nodes might be shared among jobs (may appear several times in the active nodes list), and we need to consider both cases in the heuristic. Since we have no information about the other users' communication patterns, we rely on the available data from the job scheduler. We have shown that the proposed heuristic can successfully predict the performance variability and estimate highly-varying runs on the three under-study collectives.

IV. Impact of Background Traffic on Communication

As shown in figure 2, for all the three collectives, with the increment in b, the tail of the distribution becomes more extended, which means some runs take longer than the majority. Also, Reduce and Alltoall's distribution of the highest b is dual because the routing algorithm chooses a non-minimal path when there is congestion on the minimal path. Moreover, the distribution of Alltoall possesses the longest tail because of this collective's higher communication intensity.

V. Conclusion

In this paper, we analyzed the performance variability of a compute cluster with Dragonfly+ and showed the effect of background traffic on the performance. For this purpose, we suggested a new estimation heuristic for background traffic that the scheduler can use further to make more accurate decisions. We have also shown that the network background traffic, among the performance variability sources, is the critical factor affecting the performance of different communication patterns on a Dragonfly+ topology. Generally, we observed that with the increase in the estimated background traffic, the communication not only takes a longer time, but it also becomes more unpredictable. The insights from this paper can be used to improve the scheduling decisions by SLURM.

References

[1] Marconi100. https://www.hpc.cineca.it/hardware/marconi100. Accessed: 2022–06–22.

[2] Abhinav Bhatele, Jayaraman J Thiagarajan, Taylor Groves, Rushil Anirudh, Staci A Smith, Brandon Cook, and David K Lowenthal. 2020. The case of performance variability on dragonfly-based systems. In 2020 IPDPS. IEEE, 896–905.

[3] Alexander Shpiner, Zachy Haramaty, Saar Eliad, Vladimir Zdornov, Barak Gafni, and Eitan Zahavi. Dragonfly+: Low cost topology for scaling datacenters. In 2017 HiPINEB. IEEE, 1–8.