Analyzing Hardware Parameters for GPU based HPC Platform

Saptarshi Bhowmik
Florida State University
bhowmik@cs.fsu.edu

Nikhil Jain
NVIDIA Inc.
nikhijain@nvidia.com

Abhinav Bhatele
University of Maryland
bhatele@cs.umd.edu

Xin Yuan
Florida State University
xyuan@cs.fsu.edu

***Abstract*—In today's world, , for increasing the computational capacity of a compute node and reducing the number of endpoints in an interconnect network, more number of HPC Platforms are switching to GPU based compute nodes. However, the actual performance of HPC system all together, is affected by various other environment and hardware parameters. Here, in this poster we are studying the effect of one crucial hardware parameter 1) Link Bandwidth and one simulation environment of 2)GPUs per node, on the performance of few common HPC applications, in the context of two popular topology - Fat Tree [?] and 1D - Dragonfly [?].**

***Index Terms*—performance, GPU per node, interconnect**

I. Introduction

Currently deployed networks are trying to reduce the number of endpoints by leveraging the GPU based compute nodes. These new GPU based compute nodes have larger computational prowess compared to a traditional CPU based compute nodes. However, there has been a continual discrepancy of computation and communication performance, with later, growing at a much slower rate. Along with that, increasing number of modern accelerations further dwindles this ratio. As such, there is a need to identify the environment and hardware specifications to make best possible use of the available system, and have a ideal communication/computation balance.

II. Methods

A. Trace Collection

We use 6 representative applications for our experiments. We profile and collect the traces for these applications using Score-P [**?**].

B. TraceR-CODES

We use the discrete event driven simulator, TraceR-CODES [**?**] to replay the application traces. The simulator network model is used to implement the interconnect topology, on top of which the traces are replayed.

C. Network Topologies

We use two popular interconnect topology 1D-Dragonfly and Fat-Tree for all our simulation [**?**]. For 1D-Dragonfly, we start our simulation with 16 groups and 1 GPU per compute node. We then reduce the size of the network to 8 group for 2 GPUs per compute node, 4 group for 4 GPUs per compute node and 1 group for 8 GPUs per compute node. For Fat-Tree, we start with 8 pods for 1 GPU per compute node and keep on reducing the number of pods as we increase the number of GPUs per compute node. Ultimately have four configurations for Fat-Tree 8 pods for 1 GPU per compute node, 4 pods for 2 GPU per compute node, 2 pods for 4 GPU per compute node and 1 pods for 8 GPU per compute node.

D. Bandwidth

We set the base bandwidth(x) for the Links as 11.9 Gb/s, which is the link bandwidths used in Quartz machines, and keep the internal bandwidth as 23.8 GB/sec. We use 8 more bandwidth, x/16, x/8, x/4, x/2, 2x, 4x, 8x, 16x, which are a proportion of the base bandwidths, for our simulations.

E. GPUs per node

We are using 1 GPU per node for a maximum sized network and then we are subsequently increasing it to 2 GPUs per node, 4 GPUs per node and 8 GPUs per node, with reducing the network size simultaneously.

F. Workload

We are running 20 Workloads of randomly selected jobs from the six application listed in the table below, from ranks 32, 64, 128, 256 and 512. We make sure that each rank of an application appears at least 4 times throughout all the 20 workloads.

TABLE I
Application Traces

Traces	Computation	Communication
Stencil4d	✗	✓
Kripke	✓	✗
Laghos	✓	✗
Subcomm-a2a	✗	✓
Sw4lite	✓	✓
Amg	✓	✓

III. Result

A. Impact of GPUs per Node

The below results show application speedup with respect to the default setting is 1 GPU per Node.

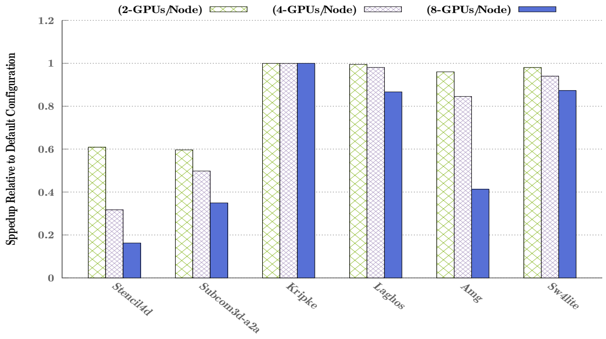

Fig. 1. Different GPUs per node mapping for all Applications of 128 ranks in 1D-Dragonfly

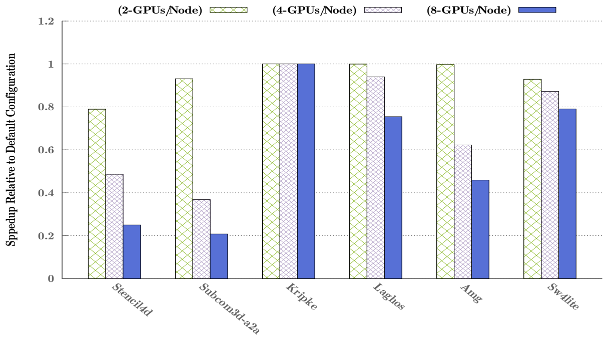

Fig. 2. Different GPUs per node mapping for all Applications of 128 ranks in Fat-Tree

Communication intensive applications experience more slowdown than applications with computation.

B. Impact of Bandwidth

The below results show application speedup with respect to the default setting is 1 GPU per Node and Link Bandwidth 11.9 Gb/s

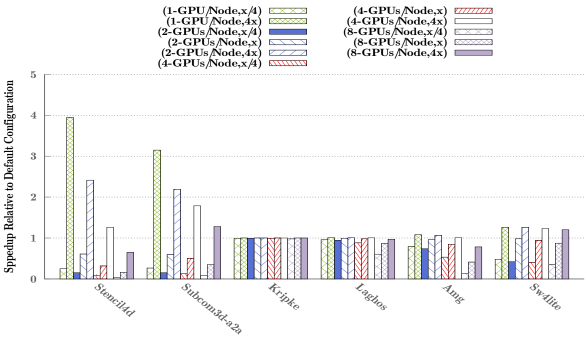

Fig. 3. Different Bandwidth and GPUs per node mapping for all Applications of 128 ranks in 1D-Dragonfly

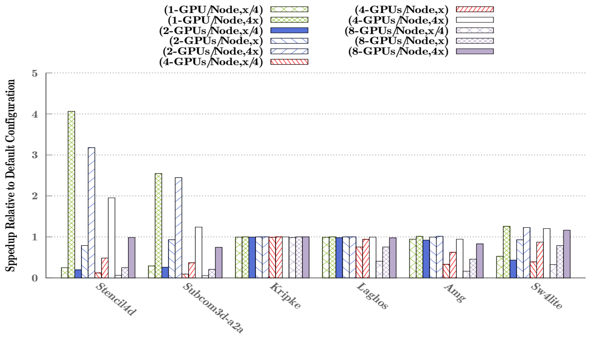

Fig. 4. Different Bandwidth and GPUs per node mapping for all Applications of 128 ranks in Fat-Tree

Applications performance speedup when the bandwidth is increased. The increase is more pronounced is full computational intensive applications.

IV. Conclusion

- As the number of GPU per node increases, the node becomes more intensive, and thus, there is a slowdown in application performance as the communication/computation capacity of the network reduces.
- As the number of GPU per node increases, more bandwidth is needed to substantiate the slowdown in application performance.
- Every application has a sweet spot where it is performing the best.

V. Future Works

- Study how other simulation environment and hardware design, such as NIC scheduling policies effect the performance of applications
- Profile more HPC applications and find the performance of those applications across the currently deployed GPU based interconnect topology,

References

[1] Jain, N., Bhatele, A., Howell, L. H., Böhme, D., Karlin, I., León, E. A., ... Leininger, M. L. (2017, November). Predicting the performance impact of different fat-tree configurations. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-13).

[2] Kim, J., Dally, W. J., Scott, S., Abts, D. (2008, June). Technology-driven, highly-scalable dragonfly topology. In 2008 International Symposium on Computer Architecture (pp. 77-88). IEEE.

[3] Knüpfer, A., Rössel, C., an Mey, D., Biersdorff, S., Diethelm, K., Eschweiler, D., ... Nagel, W. E. (2012). Score-p: A joint performance measurement run-time infrastructure for periscope, scalasca, tau, and vampir. In Tools for High Performance Computing 2011 (pp. 79-91). Springer, Berlin, Heidelberg.

[4] Acun, B., Jain, N., Bhatele, A., Mubarak, M., Carothers, C. D., Kale, L. V. (2015, August). Preliminary evaluation of a parallel trace replay tool for hpc network simulations. In European Conference on Parallel Processing (pp. 417-429). Springer, Cham.

[5] Alzaid, Z. S. A., Bhowmik, S., Yuan, X., Lang, M. (2020, June). Global link arrangement for practical Dragonfly. In Proceedings of the 34th ACM International Conference on Supercomputing (pp. 1-11). Magnetics Japan, p. 301, 1982].