A Simulation Study of Hardware Parameters for GPU-based HPC Platforms

Saptarshi Bhowmik*, Nikhil Jain‡, Xin Yuan*, Abhinav Bhatele†

*Department of Computer Science, Florida State University, Tallahassee, FL 32306 USA
‡NVIDIA, Inc., Santa Clara, CA 95051 USA
†Department of Computer Science, University of Maryland, College Park, MD 20742 USA
E-mail: *bhowmik@cs.fsu.edu, ‡nikhijain@nvidia.com, *xyuan@cs.fsu.edu, †bhatele@cs.umd.edu

***Abstract*—High Performance Computing (HPC) platforms are switching to GPU-based compute nodes; the resulting trend is the increase in per node computational capacity and the reduction of the number of endpoints in the system. This trend changes the computation and communication balance in comparison to the pre-GPU era HPC platforms, which warrants a re-study of the hardware architectural parameters. In this research, we perform a simulation study of the impact of crucial hardware parameters in GPU-based systems using HPC workloads that consist of representative HPC applications. The hardware parameters studied include (1) link bandwidth, (2) number of GPUs per node, and (3) interconnection network topology.**

***Index Terms*—GPU clusters, computation and communication balance, hardware parameters, performance.**

I. Introduction

GPUs are increasingly used in High Performance Computing (HPC) platforms. A compute node in high-end HPC systems often has multiple GPUs. The resulting trend is the increase in per-node computational capacity and the decrease in the number of endpoints in the system – the computation and communication ratio in such systems is different from that in pre-GPU era platforms. For a GPU-based platform to achieve high performance, it is imperative that computation and communication in the system remain balanced. Important hardware architectural parameters such as the link bandwidth and the number of GPUs per node are crucial design parameters that will determine the balance and thus, the overall performance of the system. In this research, we leverage the whole system simulation capability of TraceR-CODES [1] and use it to study the impact of hardware parameters using HPC workloads that consist of representative HPC applications. The parameters studied include (1) interconnect link bandwidth, (2) number of GPUs per node, and (3) interconnect topology. The study gives an insight into the hardware parameters and the results can be used to guide the design of GPU-based HPC platforms.

II. Methodology

We used the discrete event-driven simulator, TraceR-CODES [1] for the study. TraceR-CODES is capable of simulating the whole system with different hardware parameters. Moreover, it can simulate a workload in the system that consists of one or multiple applications by replying the traces of the applications.

A. Applications and Workloads

We selected 6 representative applications for our experiments that include two computation-intensive kernels Kripke and Laghos, two communication-intensive kernels Stencil4d and Subcomm3d, and two applications Sw4lite and Amg that have a mix of communication and computations. We profiled and collected the traces for these applications of different ranks using Score-P. The applications are summarized in Table I.

We are running 20 Workloads of randomly selected jobs from the six applications listed in Table I, from ranks 32, 64, 128, 256 ,and 512, to fill up the whole system. We made sure that each rank of an application appeared at least 4 times throughout all the 20 workloads.

TABLE I
Application Traces

Traces	Computation-intensive	Communication-intensive
Stencil4d	✗	✓
Subcomm3d	✗	✓
Kripke	✓	✗
Laghos	✓	✗
Amg	✓	✓
Sw4lite	✓	✓

B. Network Topologies

We used two popular interconnect topology 1D-Dragonfly [2], [3] and Fat-Tree [4] in our simulations. For 1D-Dragonfly, we started our simulation with 16 groups and 1 GPU per node. We then reduced the size of the network to 8 groups for 2 GPUs per node, 4 groups for 4 GPUs per node ,and 1 group for 8 GPUs per node. For Fat-Tree, we started with 8 pods for 1 GPU per node and kept on reducing the number of pods as we increased the number of GPUs per node. Ultimately, we had four configurations for Fat-Tree 8 pods for 1 GPU per node, 4 pods for 2 GPUs per node, 2 pods for 4 GPUs per node and 1 pod for 8 GPUs per node. For both networks, the maximum sized system (for 1 GPU per node) has 2048 total compute nodes (128 nodes per group for 1D-Dragonfly and 256 nodes per pod for Fat-Tree).

C. Bandwidth

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

D. GPUs per Node

We used 1 GPU per node for a maximum-sized network and then we subsequently increased it to 2 GPUs per node, 4 GPUs per node ,and 8 GPUs per node, while reducing the network size simultaneously.

III. Results

A. Impact of GPUs per Node

Figure 1 shows the application speedup with respect to the default setting of 1 GPU per node with the default link bandwidth (1x) on 1D-Dragonfly. As the total number of GPUs condense to a smaller number of compute nodes, the performance of communication kernels (Stencil4d and Subcomm3d) drops significantly while the performance of computational intensive kernels (Kripke and Laghos) remains similar. The performance of applications (Amg and Sw4lite) is also affected as shown in the figure. The results on Fat-Tree have a similar trend and are omitted.

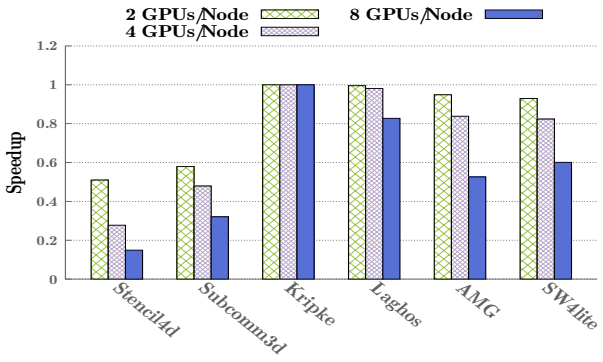

Fig. 1. Speed-up over default setting for different GPUs per node mapping, for all Applications of 128 ranks, in 1D-Dragonfly

B. Impact of Bandwidth

Figure 2 shows application speedup with respect to the default setting is 1 GPU per node and Link Bandwidth x (11.9 GB/sec) on 1D-Dragonfly. There are two key observations. First, for applications that are sensitive to communication performance (Stencil4D, Subcomm3d, Amg, and Sw4lite), as the number of GPU per node increases, more link bandwidth is needed to sustain the performance – insufficient bandwidth will significantly slow down the applications. This is even observed in the computation-intensive Laghos. Second, every application has a "sweet spot" where it is performing the best. This indicates that a substantial benchmarking study will be needed to determine the best system configurations for the GPU-based systems. Figure 3 shows the results with the Fat-Tree topology, the trend is the same. Thus, these conclusions apply across topologies.

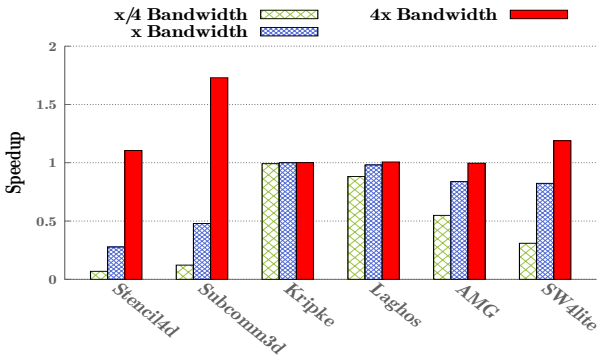

Fig. 2. Speed-up over default setting for various bandwidths in 4 GPUs per node mapping, for all Applications of 128 ranks, in 1D-Dragonfly

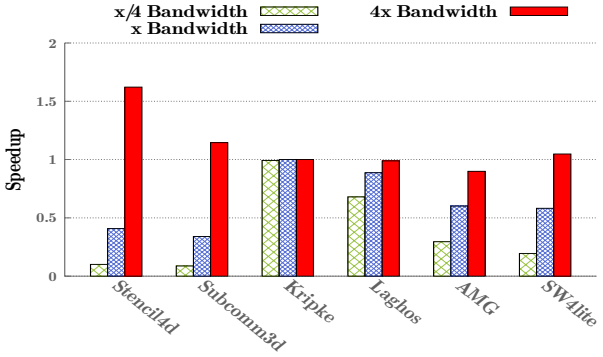

Fig. 3. Speed-up over default setting for various bandwidths in 4 GPUs per node mapping, for all Applications of 128 ranks, in Fat-Tree

IV. Conclusion and Future Work

We performed a simulation study of hardware Parameters for GPU-based HPC Platforms. What differentiates our study from others is that we used workloads consisting of representative applications. Our results shed light on the impact of hardware parameters on real applications. In the future, we plan to extend our study by (1) considering other interconnect choices, (2) using more applications, and (3) studying other system parameters.

Acknowledgment

This work was supported by funding provided by the University of Maryland College Park Foundation.

References

[1] B. Acun, N. Jain, A. Bhatele, M. Mubarak, C. D. Carothers, and L. V. Kale, "Preliminary evaluation of a parallel trace replay tool for hpc network simulations," in *European Conference on Parallel Processing*. Springer, 2015, pp. 417–429.

[2] 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.

[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*, 2020, pp. 1–11.

[4] N. Jain, A. Bhatele, L. H. Howell, D. Böhme, I. Karlin, E. A. León, M. Mubarak, N. Wolfe, T. Gamblin, and M. L. Leininger, "Predicting the performance impact of different fat-tree configurations," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017, pp. 1–13.