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

Saptarshi Bhowmik[1], Nikhil Jain[2], Xin Yuan[1] and Abhinav Bhatele[3]

Florida State University[1], NVIDIA[2], Inc., University of Maryland, College Park[3]

Goals

Understand the impact of the following hardware parameters on GPU-based HPC platforms:

- Number of GPUs per node
- Interconnect link bandwidth
- Interconnect topology

Introduction

GPUs are increasingly used in high performance computing (HPC) platforms, resulting in an increase in per-node computational capacity and decrease in the number of endpoints in the system. As such, computational and communication capability of the system must remain balanced. Hardware architectural parameters such as the link bandwidth and the number of GPUs per node are crucial design parameters that determine this 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.

Methodology

The Tracer-CODES Simulation Framework

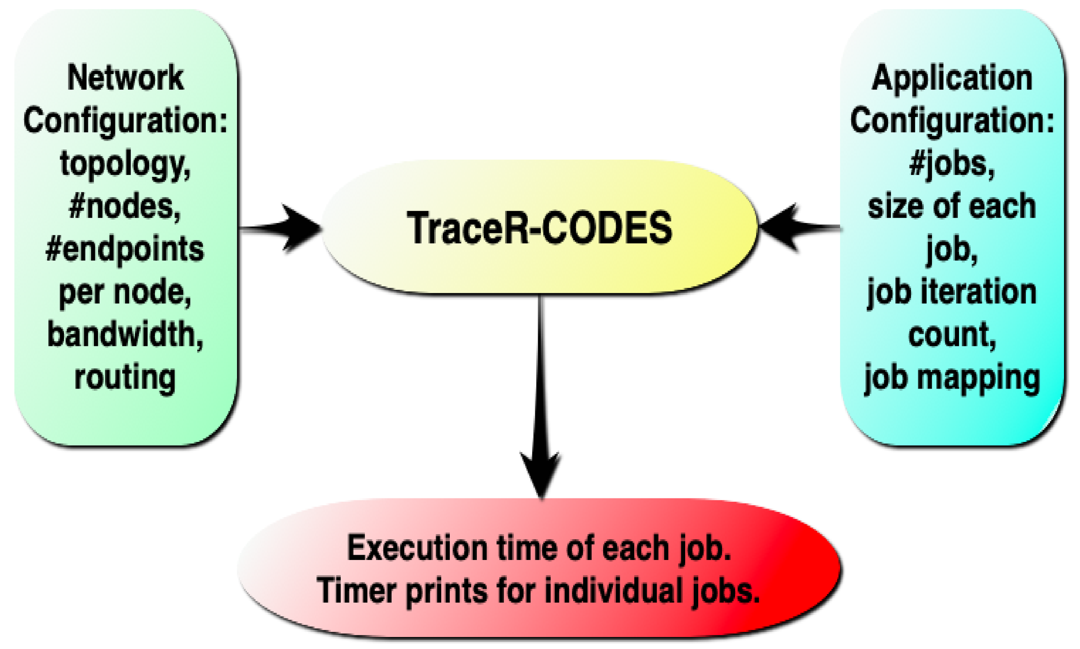

Figure 1: The TraceR-CODES workflow.

Applications Used for Generating Traces

Six benchmarks and proxy applications are used in the simulation.

Application	Computation Intensive	Communication Intensive
Stencil4d	✗	✓
Subcomm3d	✗	✓
Kripke	✓	✗
Laghos	✓	✗
AMG	✓	✓
SW4lite	✓	✓

Table 1: Computational and communication properties of applications.

Simulation Parameters

Network Topologies

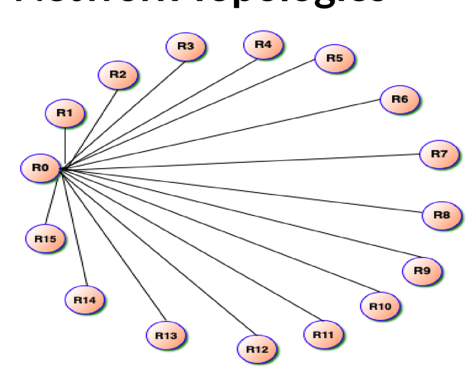

Figure 2: A 1D Dragonfly group [2] [4], (Intra-group links of router R0).

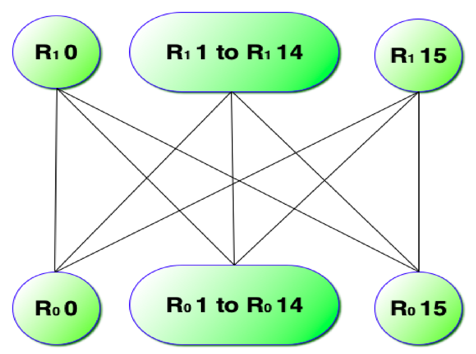

Figure 3: A Fat-Tree pod [3].

Network Sizes and GPUs per node

GPUs per node	1D Dragonfly	Fat-Tree
1 GPU/Node	16 Groups	8 Pods
2 GPUs/Node	8 Groups	4 Pods
4 GPUs/Node	4 Groups	2 Pods
8 GPUs/Node	2 Groups	1 Pods

Table 2: Network sizes for different GPUs per node, default is 1 GPU per node.

Total number of GPUs in the system = 2048

Bandwidth

Default Setting :
Link Bandwidth = 11.9 GB/sec
Internal Bandwidth = 23.8 GB/sec

We use eight other bandwidth settings: **x/16, x/8, x/4, x/2, 2x, 4x, 8x, 16x** for our simulations.

Results

Impact of Number of GPUs per Node

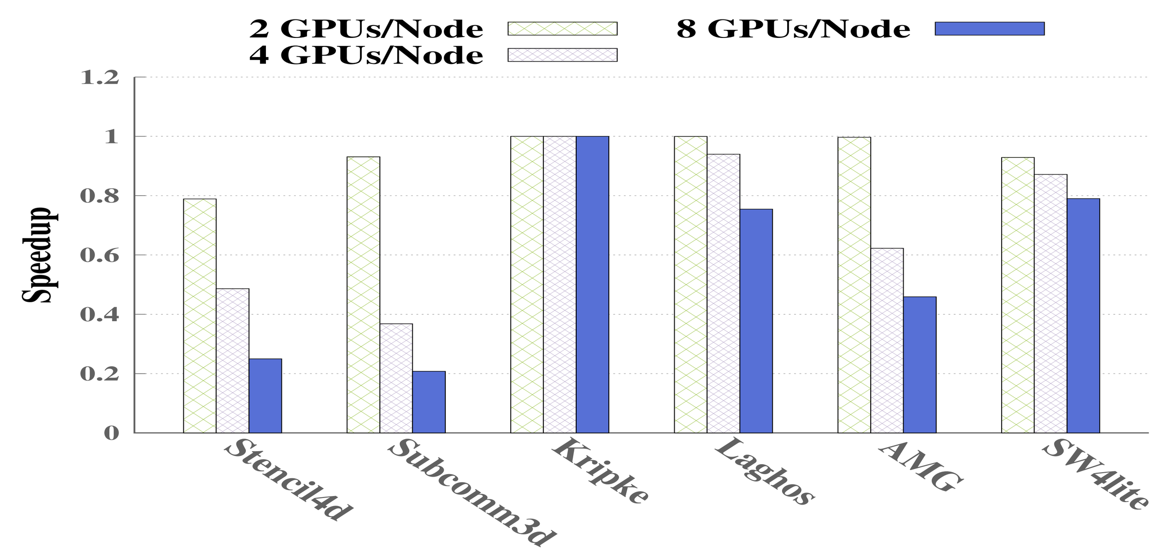

Figure 4: Speedup over default setting (1 GPU per node) for different GPUs per node, for all applications traces of 128 processes (Fat-Tree.)

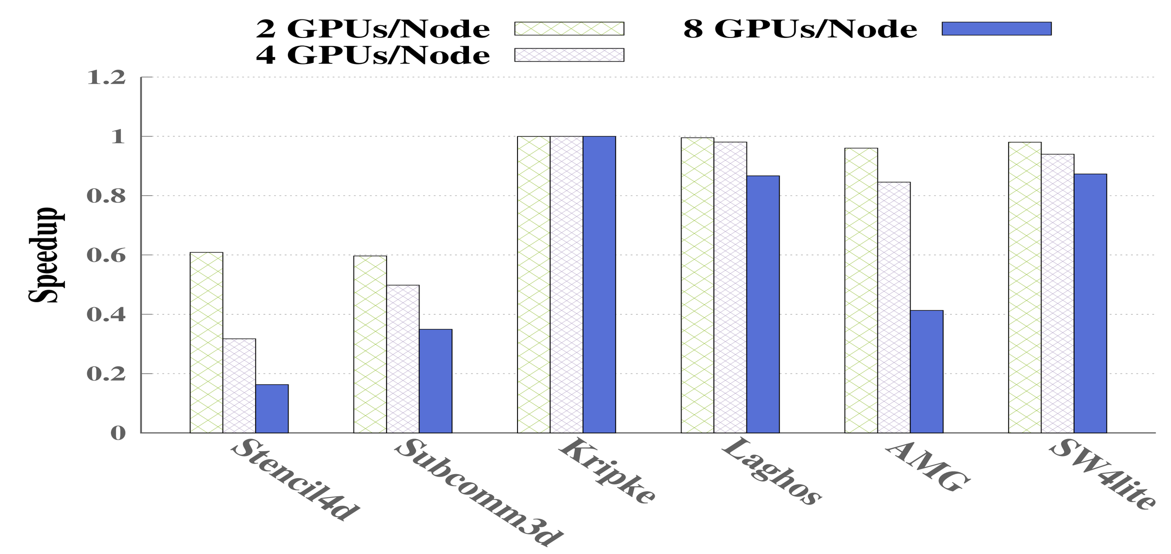

Figure 5: Speedup over default setting (1 GPU per node) for different GPUs per node, for all applications traces of 128 processes (1D Dragonfly.)

The performance of communication kernels, Stencil4d and Subcomm3d drops significantly while the performance of compute-intensive kernels, Kripke and Laghos remains similar, for both 1D Dragonfly and Fat-Tree topology.

Impact of Network Bandwidth

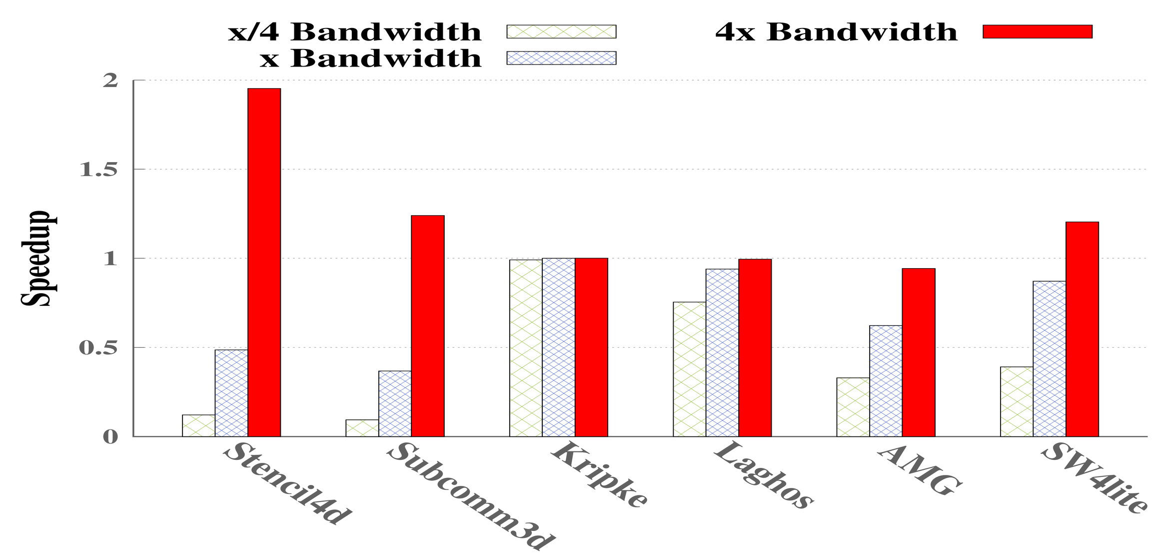

Figure 6: Speedup over default setting for various bandwidths in the 4 GPUs per node mode, for all applications of 128 processes, (in Fat-Tree.)

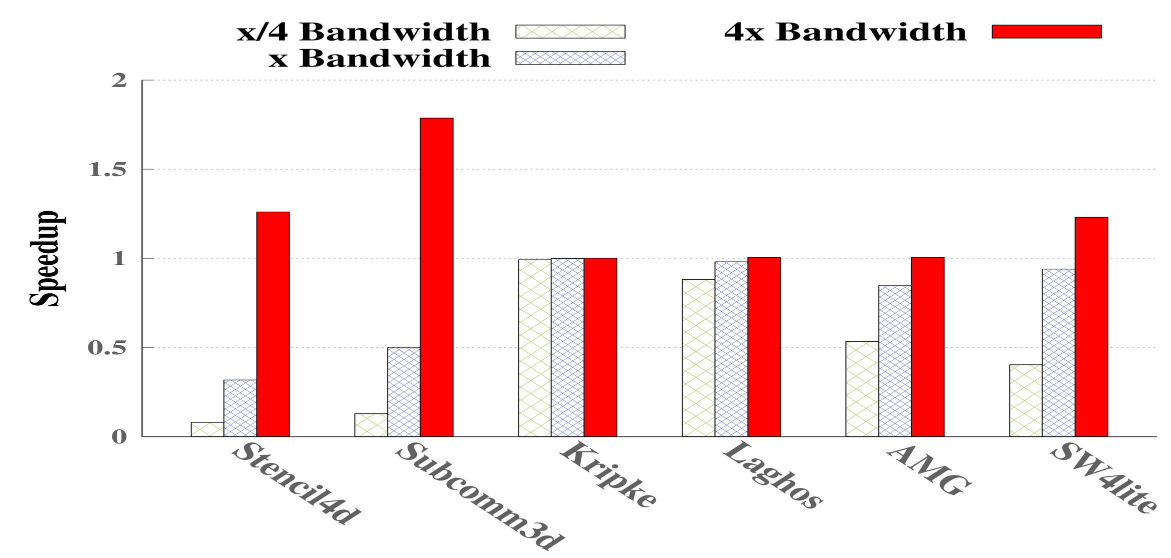

Figure 7: Speedup over default setting for various bandwidths in 4 the GPUs per node mode, for all applications of 128 processes (1D Dragonfly.)

For applications that are sensitive to communication -- Stencil4D, Subcomm3d, AMG, and SW4lite, as the number of GPUs per node increases, more link bandwidth is needed to sustain the performance -- insufficient bandwidth.

Conclusion

- As the number of GPUs per node increases, the node becomes more computation-intensive, and thus, there is a slowdown in application performance as the communication/computation ratio of the network reduces.
- As the number of GPUs 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. This indicates that a substantial benchmarking study will be needed to determine the best system configurations for GPU-based systems.

Future Work

- Considering other interconnect choices, such as HyperX.
- Using more proxy and production applications.
- Studying other system parameters such as NIC-level packet scheduling and buffer size.

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 Com- puter Architecture. IEEE, 2008, pp. 77–88.

[3] N. Jain, A. Bhatele, L. H. Howell, D. Bo¨hme, I. Karlin, E. A. Leo´n, M. Mubarak, N. Wolfe, T. Gamblin, and M. L. Leininger, “Predicting the performance impact of different fat-tree configurations,” in Proceed- ings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017, pp. 1–13

[4] 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.