ACM SRC Poster: Optimizing All-to-All Algorithm for PERCS Network Using Simulation

Ehsan Totoni and Laxmikant V. Kale
Department of Computer Science
University of Illinois at Urbana-Champaign
{totoni2, kale}@illinois.edu

ABSTRACT

Communication algorithms play a crucial role in the performance of large-scale parallel systems. They are implemented in runtime systems and used in most parallel applications as a critical component. As vendors are willing to design new custom networks with significantly different performance properties for their new supercomputers, designing new efficient communication algorithms is an inevitable challenge. This task is desirable to be done before the machine comes online since inefficient use of the system before the new algorithm's availability is a huge waste of a possibly hundreds of millions of dollars resource. Here, we demonstrate the usability of our simulation framework, BigSim, in meeting this challenge. Using BigSim, we observe that the commonly used Pairwise-Exchange algorithm for all-to-all communication pattern is suboptimal for a supernode of the PERCS network (two-level directly connected similar to Dragonfly topology). We designed a new all-to-all algorithm for it and predict a five-fold performance improvement for large message sizes using this algorithm.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems—*Design studies*

General Terms

Algorithms, Design, Measurement, Performance

1. INTRODUCTION

Many supercomputers are being deployed with tens or hundreds of millions of dollars in cost. They are complex machines that typically have hundreds of thousands of cores or even more than a million cores. Their interconnection topologies are also very complicated to connect this enormous number of cores effectively. Interconnection topologies are becoming more intricate as lower latencies and higher bandwidths are required on the path to Exascale. This is especially important as topologies different than well-established ones (e.g. other than 3D Torus) are being proposed and used in newer machines. These new topologies make the task of designing and implementing runtime communications and collectives libraries very difficult.

For new supercomputers in general, porting and tuning applications and runtimes can take months to years. In this period, the machine will be used with much less efficiency than possible just to make use of it. Given the four to five-year effective life time of typical supercomputers, this is a huge waste of resources! Designing efficient communication and collective algorithms inside runtime systems is particularly important because performance of most parallel applications depends on them.

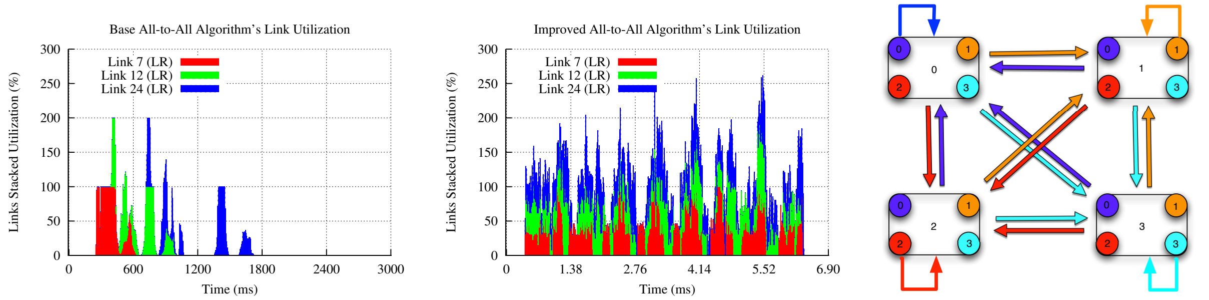
Our approach is to use detailed simulation and analysis to tune the applications and runtimes before the machine comes online. BigSim [1] simulation framework has been developed over the years for this purpose. Here, as a case study, we show how we identified that the standard Pairwise-Exchange algorithm for MPI_Alltoall is inefficient on the PERCS [2] architecture and how we designed a new algorithm. PERCS is a two-level directly connected network (similar to Dragonfly topology) and will be used in many IBM machines in the future. In addition, Dragonfly is proposed as a possibility in the Exascale study report (Kogge et al.)and other vendors (such as Cray) are considering variants of it. Furthermore, the simulation-based methodology used in this work can be used to design other communication algorithms for different networks as well. Overall, our algorithm is designed for All-to-all inside a "Supernode" of PERCS and shows five-fold improvement over the older algorithm. A more comprehensive version of this work will appear in another publication [3].

1.1 BigSim Simulation Framework

BigSim [1] simulation framework addresses the above mentioned issue by its unique "emulation followed by simulation" approach. The emulation runs the user application at the target scale using a much smaller machine. For example, an application can be run on an existing one hundred thousand machine pretending to have one million cores. This is made possible using processor virtualization feature of Charm++ and AMPI runtime systems. Other than revealing the scaling bugs by running the application at scale, emulation produces the application traces needed for simulation.

These traces contain the dependencies of computations and messages, as well as salient features of computation blocks. The simulator of BigSim uses these traces to produce different performance and timing outputs. In this work, we plugin a packet-level network model of PERCS network (which has been validated extensively [3]) inside the simulator to model the machine accurately.

SC'11 Companion, November 12–18, 2011, Seattle, Washington, USA.
ACM 978-1-4503-1030-7/11/11.



(a) Link utilization of pairwise-exchange (b) Link utilization of new algorithm (c) Sends in first phase of the new all-to-all algorithm

Figure 1: New all-to-all algorithm illustration

1.2 PERCS Architecture

PERCS (Productive, Easy-to-use, Reliable Computing System) is an architecture by IBM that uses a two-level directly-connected network [2]. In PERCS, the system is divided into *supernodes*, containing 32 nodes each. Each supernode is divided into four drawers (eight nodes per drawer). 24 GB/s LLocal (LL) links connect each node to the seven other nodes in its drawer and 5 GB/s LRemote (LR) links connect it to the other 24 nodes in its supernode. 10 GB/s D links connect all supernodes to one another.

Each node contains four POWER7 chips, with total of 32 cores, and a Hub chip. The POWER7 chips are connected with 192 GB/s of bandwidth to the Hub chip. This Hub chip interfaces the node with the network.

2. ALLTOALL OPTIMIZATION

`MPI_Alltoall` is an important collective operation, which is used in many parallel applications and kernels such as FFT and Matrix Transpose. We narrow our focus to All-to-all of large messages inside a supernode, because of its many practical interests. The Pairwise-Exchange algorithm is the dominant approach for all-to-all of large messages, which is based on tightly coupled send-receive operation of $P/2$ pairs of tasks in each step. Figure 1(a) is BigSim's output that shows link utilizations of three different links of a node at the same time for this algorithm. It can be seen that Pairwise-Exchange algorithm does not utilize all the links simultaneously for the whole all-to-all duration. Thus, in our new algorithm, we try to exploit all the links of a node by having each core send to a different node in each phase of the algorithm:

1. $t = n * c$ tasks are running on n nodes with c cores each.

2. Each task has to send $t - 1$ messages. Any core can reach a particular set of c cores by using the direct link between the destination node and its home node.

3. In phase i $(0 \leq i \leq n-1)$, core j $(0 \leq j \leq c-1)$ on every node sends data to the set of cores residing in the $((j+i) \bmod n)$th node.

Figure 1(c) illustrates one phase of the algorithm by using a four node fully connected network (four cores per node). An arrow with the same color as a core shows the core's destination in that phase. Using this algorithm we obtain the link utilization graph of Figure 1(b), which shows that all the links are being used for the whole all-to-all duration.

Figure 2 compares the execution time of our algorithm with Pairwise-Exchange algorithm and theoretical peak of the links. Note that the theoretical calculation does not consider other limitations of the hardware, such as bandwidth of each node to the network and limited buffer sizes of the Hub chip. We see up to 80% reduction in execution time of all-to-all for large messages, which corresponds to a five-fold improvement.

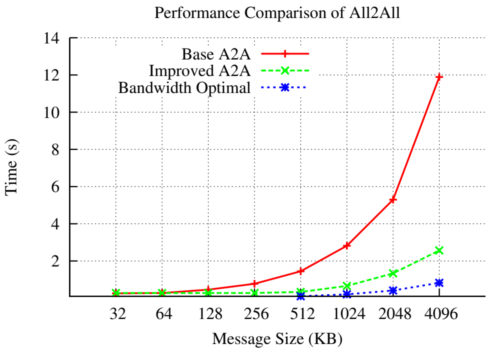


Figure 2: Performance comparison of different algorithms for all-to-all

3. REFERENCES

[1] Gengbin Zheng, Terry Wilmarth, Praveen Jagadishprasad, and Laxmikant V. Kalé. Simulation-based performance prediction for large parallel machines. In *International Journal of Parallel Programming*, volume 33, pages 183–207, 2005.

[2] B. Arimilli, R. Arimilli, V. Chung, S. Clark, W. Denzel, B. Drerup, T. Hoefler, J. Joyner, J. Lewis, Jian Li, Nan Ni, and R. Rajamony. The PERCS High-Performance Interconnect. In *2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI)*, pages 75 –82, August 2010.

[3] Ehsan Totoni, Abhinav Bhatele, Eric Bohm, Nikhil Jain, Celso Mendes, Ryan Mokos, Gengbin Zheng, and Laxmikant Kale. Simulation-based performance analysis and tuning for a two-level directly connected system. In *Proceedings of the 17th IEEE International Conference on Parallel and Distributed Systems*, December 2011.