Award Number: CNS-551555
Title: CRI: Acquisition of an InfiniBand Cluster with SMP Nodes
Institution: Florida State University
PIs: Xin Yuan, Robert van Engelen, Kartik Gopalan

A Flexible Cluster Infrastructure for Systems Research and Software Development

Xin Yuan
Department of Computer Science, Florida State University
Tallahassee, FL 32306
xyuan@cs.fsu.edu

Abstract

This document reports the current status of our CRI project. We designed a flexible cluster infrastructure that can be configured to emulate various high-end clusters. This computing research infrastructure will facilitate systems research and development, and greatly improve the research productivity of the participating groups. Due to the unexpected delay in our infrastructure acquisition (the first vendor selected was unable to deliver the system), the proposed infrastructure is still in the process of being installed by the second vendor. As a result, we can only discuss in this document some of the expected outcomes of our research, facilitated by the proposed infrastructure. In particular, we will describe how this infrastructure will speed up our research in developing adaptive libraries for Message Passing Interface (MPI) collective routines.

1 Introduction

This CRI grant provides funding for a much needed research experimental platform that will be used in various system-oriented research projects that involve extensive research and software development at the systems level. To ensure that the techniques developed in our research projects are valid and practical, the experimental platform must closely match real-world systems since performance optimization techniques are usually platform dependent. Most of contemporary high-end clusters are composed of commodity SMP/Multi-core nodes connected by high speed system area networks such as InfiniBand, Myrinet, and Quadrics. This grant allows us to install such a system that is dedicated to our research. With this research infrastructure, the PIs can develop practical techniques that can be applied to real-world high-end clusters by either extending their existing results (on other platforms) or devising new schemes. Having a dedicated cluster will significantly improve our research productivity as will be discussed later in this report.

We designed the infrastructure such that the system can emulate clusters with different networking technologies and nodal architectures. The networking technologies supported by the infrastructure include InfiniBand DDR (double data rate, 20Gbps), InfiniBand SDR (single data rate, 10Gbps), and Gigabit Ethernet. The nodal architectures supported include hybrid SMP/multi-core nodes, multi-core nodes, SMP nodes, and single CPU nodes. The flexibility of the infrastructure will allow us to develop techniques for many different types of clusters of workstations. We will describe the infrastructure in more detail in the next section.

The grant started in March 2006. The final design of the infrastructure was completed in Summer 2006. In Fall 2006, we placed an order to a small hardware vendor that was selected by a formal bidding process at Florida State University (FSU). The formal bidding process was fairly lengthy and the order went out of FSU in late November, 2006. The vendor promised to deliver the system in 4 to 6 weeks (specified in the bidding document). In late February 2007 (about 12 weeks after the order was placed), a cluster was built, but had many problems. We continued to interact with the vendor, hoping that the company could resolve the problems in this cluster. Several weeks after we first tested the cluster, it became clear to us that this vendor does not have the expertise to deliver a viable system. We had to make a painful decision to cancel the order and to look for a second vendor. The order was officially canceled late March 2007. By the time that we canceled the order, the cluster still could not correctly run all programs in the NAS parallel benchmarks. From the experience with the first vendor, we learned a hard lesson: small cluster vendors may offer low prices for their clusters, but they may not be able to deliver what they promised. Thus, in the second round, we only interact with large equipment vendors such as Dell, IBM, and HP. Eventually, we decided to purchase a cluster solution from Dell. An order has been placed to Dell and we expect to have a complete system in several weeks.

Due to the unexpected delay in our infrastructure acquisition, we have to change our research agenda. Several of our research projects were delayed because of the delay of the infrastructure. However, we believe that the availability of this infrastructure will significantly improve our research productivity due to the nature of our research projects. In this report, we will discuss some of the expected outcomes, facilitated by the proposed infrastructure. Since most of our research projects are similar in nature, we will focus on describing how the infrastructure will speed up the research in one of our projects: developing adaptive libraries for Message Passing Interface (MPI) collective routines. The educational projects that will make use of this infrastructure are not affected by the delay since we planned to use the cluster starting in Fall 2007.

2 A flexible cluster infrastructure

The main design objective of the research infrastructure is to achieve maximum flexibility in configuring the networking architecture and the nodal architecture. Having accesses to different types of clusters facilitates the development of techniques for those clusters. In the original proposal, we propose to acquire an InfiniBand cluster with dual-processor SMP nodes. Since the time that we submitted the proposal, there is major development in both the InfiniBand technology and the nodal architecture: (1) multi-core processors have become ubiquitous in 2006-2007; (2) the InfiniBand DDR (double data rate) technology that offers a 20Gbps link speed has become more prominent. Accordingly, we redesign the infrastructure so as to keep up with the technology advances.

Our final design materialized to be a cluster solution that we ordered from Dell. The solution is illustrated in Figure 1. The cluster has a total of 18 nodes: one front-end node, one administrative node, and 16 compute nodes. The compute nodes are Dell PowerEdge 1950 with two Intel Xeon E5345 2.33GHz quad-core processors and 8GB memory. Such a node can be used to emulate the four different kinds of cluster nodes that we are interested in: a hybrid multi-core SMP node when all cores and both processors are used, a multi-core node when only one processor is used, an SMP node when two cores on two processors are used, and single processor nodes when only one core is used. There are two switches for data communications, one is a CISCO SFS-7000D, an InfiniBand DDR/SDR switch, and the other one is a Nortel 5510 Gigabit Ethernet switch. CISCO SFS-7000D can be configured to either operate at the DDR rate (20Gbps) or the SDR rate (10Gbps). Hence,

this network organization gives us the flexibility to emulate three types of network connections: InfiniBand DDR, InfiniBand SDR, and Gigabit Ethernet. In addition to the two switches for data communications, another switch, which is used to perform administrative tasks, also connects all nodes.

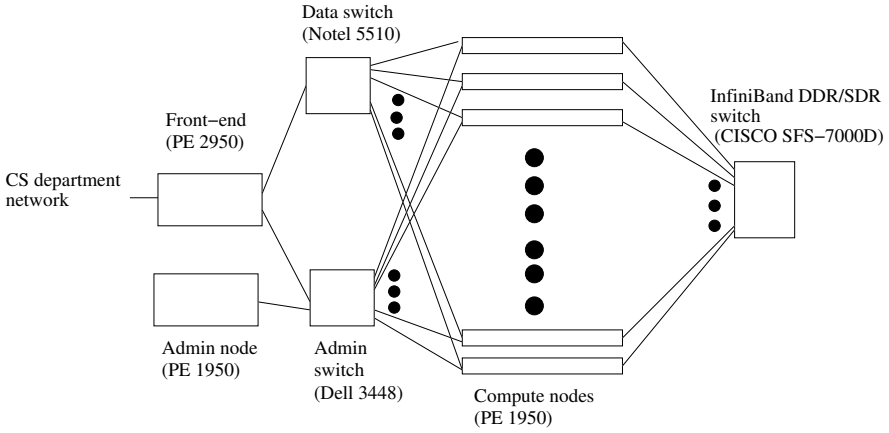

Figure 1: The cluster solution from Dell

This infrastructure can be configured to emulate many different types of clusters, including the following common cluster configurations: hybrid SMP/Multi-core nodes with InfiniBand (DDR or SDR) connections, SMP nodes with InfiniBand (DDR or SDR) connections, Multi-core nodes with InfiniBand (DDR or SDR) connections, hybrid SMP/Multi-core nodes with Gigabit Ethernet connections, SMP nodes with Gigabit Ethernet connections, Multi-core nodes with Gigabit Ethernet connections, single CPU nodes with Gigabit Ethernet connections. These configurations allow us to develop performance optimization techniques for a wide range of clusters. Moreover, the infrastructure enables the development of performance optimization techniques that can be applied uniformly across different cluster platforms.

3 CRI as a catalyst for research advances

This infrastructure will be used in a number of research projects as well as educational projects. Since most of our research projects are similar in nature, the usage of the infrastructure in these projects will be somewhat similar (in a time multiplexed manner). For this reason, we will only describe how this infrastructure will significantly improve the research productivity of one particular project: developing adaptive libraries for MPI collective routines. Next, we will describe the research in the project, discuss the challenges in the research, and illustrate how the infrastructure will significantly improve the research productivity.

3.1 The research project

The research project, "developing adaptive libraries for MPI collective routines," is funded by National Science Foundation (Award number: CCF-0541096). In this project, we propose to use the adaptive approach to overcome the performance problems in the current MPI collective operation implementations. MPI collective operations are communications that involve more than two parties. Examples include broadcast, all-to-all, and barrier communications. Such operations are used in most MPI applications and they account for a significant portion of the communication

time in some applications [5]. Unfortunately, some fundamental issues in collective operations are still not well understood and current implementations are somewhat problematic [3, 4].

The major challenges in developing efficient MPI collective communication routines that are efficient across platforms and applications lie in the facts that many factors, including networking technology, topology, and application behavior, have strong impacts on the performance of the operations. As a result, it is impossible for one algorithm to achieve good performance in different situations. In other words, the current MPI library implementation paradigm, which fixes the communication algorithm for a given data size and/or networking technology, cannot achieve high performance across platforms and applications. The idea in this research is to develop adaptive libraries that allow different communication algorithms to be used in different situations. By adapting to platforms and applications, the library routine can achieve the best performance for a given platform and application.

We have developed two prototype adaptive libraries: STAGE-MPI (Static Tuned Automatic Generation of Efficient MPI collective routines) [1] and STAR-MPI (Self Tuned Adaptive Routines for MPI collective operations) [2]. Our preliminary experience has shown that the adaptive techniques result in more robust routines than existing MPI implementations. However, our experience with the adaptive techniques also reveals some pressing issues that must be addressed for realizing efficient MPI collective routines in general and using the adaptive approach to achieve this purpose in particular. We plan to address these issues in the next stage of this project. Next, we will describe some of the important issues.

The first issue is about the general understanding of an "efficient" collective communication algorithm. The MPI developers community has been developing and analyzing collective communication routines under the assumption that all processes arrive at the collective operation at the same time. However, as shown in our recent study [3], in practice, it is much more common that processes arrive at a collective operation at different times and that the time differences are usually sufficiently large to affect the performance. We will use the term *process arrival pattern* to denote the timing when processes arrive at an operation. The performance of collective communication algorithms is sensitive to the process arrival pattern. Hence, to achieve high performance in practical cases, it is vital to understand the impact of process arrival pattern on different platforms and to develop process arrival pattern aware collective communication algorithms. The second issue relates to the architecture of SMP/multi-core clusters. There are three layers of communications in such a cluster, each having a different performance: within a processor between cores, intra-node, and inter-nodes. In addition, the inter-node communication performance can sometimes match memory operations. For example, using InfiniBand DDR gives about 11Gbps user level bandwidth. Hence, the interference between local memory accesses and remote memory accesses within one collective operation must be understood. How to achieve good communication performance in such an environment needs further investigation. Last but not the least, the software adaptation mechanisms for adaptive MPI libraries are still not mature. Efficient software adaptation mechanisms that allow the best performing communication algorithm to be determined without introducing significant performance penalties must be developed.

3.2 Improving research productivity with the infrastructure

The infrastructure will be a catalyst for research advances in this project as it will dramatically speed-up our progress towards addressing the crucial research issues discussed in the previous section. First, understanding the impacts of process arrival patterns takes a lot of experiments since there are many parameters in the problem space: different collective operations, different

collective algorithms, different platforms, and different process arrival patterns. Our research in this area currently makes use of the high-end clusters in supercomputing centers (through an NSF Teragrid grant). The research bottleneck is the job turn-around time: it usually takes a day to obtain the results for each run. With such a job turn-around speed, we feel that it is difficult to obtain a full understanding of the impact of process arrival patterns. The infrastructure will eliminate the bottleneck in our current research in addition to providing accesses to different types of clusters.

Second, experimentation is essential in designing efficient architecture specific collective algorithms for SMP/multi-core clusters. Efficient algorithms for such systems are usually obtained by fine tuning algorithm parameters such as logical topologies, data segment sizes, ways to overlap intra-node communications, inter-node communications, and memory accesses, etc. Without a dedicated cluster, it is next to impossible to thoroughly probe the whole design space. At this time, our research in this direction is also greatly limited by the job turn-around time in clusters in supercomputing centers. This infrastructure will solve the problem.

Third, designing efficient adaptive mechanisms requires a lot of tuning: having a dedicated cluster is essential. Moreover, the infrastructure facilitates the development of common adaptive mechanisms that are efficient across different cluster platforms.

In short, experimentation and software development are the essential components in our research. The infrastructure, which is dedicated and can emulate many types of platforms, will greatly improve research productivity and facilitate research that is not possible without such an infrastructure.

4 Conclusion

We summarize the current status of our CRI project, "CRI: Acquisition of InfiniBand Clusters with SMP Nodes." The proposed infrastructure should be fully operational in Summer 2007 although there was unexpected delay in the infrastructure acquisition. Our research and educational projects will significantly benefit from the infrastructure.

References

[1] A. Faraj and X. Yuan. "Automatic Generation and Tuning of MPI Collective Communication Routines." The *19th ACM International Conference on Supercomputing* (ICS'05), pages 393-402, Cambridge, MA, June 20-22, 2005.

[2] A. Faraj, X. Yuan, and D.K. Lowenthal, "STAR-MPI: Self Tuned Adaptive Routines for MPI Collective Operations," the *20th ACM International Conference on Supercomputing* (ICS'06), pages 199-208, Cairns, Australia, June, 2006.

[3] A. Faraj, P. Patarasuk, and X. Yuan, "A Study of Process Arrival Patterns for MPI Collective Operations," the *21th ACM International Conference on Supercomputing* (ICS'07), Seattle, WA, June 16-20, 2007.

[4] J.Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel, and J. Dongarra, "Performance Analysis of MPI Collective Operations," *IEEE IPDPS*, 2005.

[5] R. Rabenseinfner, "Automatic MPI counter profiling of all users: First results on CRAY T3E900-512," In *Proceedings of the Message Passing Interface Developer's and User's Conference*, pages 77-85, 1999.