High Performance Computing with 10GB Networking at a Small Institution

Stephen Akers, Rachel Frantsen, Eric Oseid, and Richard Brown
Department of Mathematics, Statistics, and Computer Science
St. Olaf College
Northfield, MN 55057
stephenaker1@gmail.com, rachelfrantsen@gmail.com, rab@stolaf.edu

Abstract

The development of 10 Gigabit fiber network technology has increased the potential speed of high performance parallel computing resources around the world. At St. Olaf, the Computer Science program employs undergraduate students as specialized staff known as “cluster managers”. We manage several Beowulf clusters and specialized servers used for research, as well as a computer science department network that is implemented and managed independently of the St. Olaf IT-managed network. Our project implemented fiber optic networking along with programmable 10-Gb managed network switches, creating two new St. Olaf computer science networks, one as a cluster of virtual machines on a few dozen hosts, and one to replace the existing computer science network. Our objective was to explore how we could use fiber connection to improve the performance of our high-performance computing resources. We will discuss the architecture of the networks, security design, performance choices, and a report of our experiences throughout the implementation of the project.

1 Introduction

1.1 Parallel and Distributed Computing

In response to the recent dramatic increase of the size and complexity of computational problems, Parallel and Distributed Computing (PDC) is becoming a widely-used technique amongst computer scientists. PDC is the process of breaking a large problem down into smaller subproblems and then solving those subproblems using simultaneous computations. By doing this, the runtime required to complete a large task can be reduced significantly. For example, a program that would take 1000 seconds to execute sequentially could be reduced to roughly 250 seconds by distributing the computational load among four processors.

1.1.1 Platforms for PDC: Multi-core CPUs and Beowulf Clusters

Multi-core CPUs are the most commonly used PDC platform. A multi-core CPU is a CPU equipped with multiple computational units that can run programs in parallel. Another common PDC platform is a Beowulf cluster. A Beowulf cluster is composed of multiple consumer-grade computers (referred to as nodes) which are connected and networked together so that they can communicate with each other. A programmer can use a Beowulf cluster to parallelize a computation by writing programs that take advantage of each node in the cluster. This paper concerns St. Olaf's Beowulf clusters.

1.1.2 Importance of Connection Speed

The St. Olaf computer science department maintains and uses a number of Beowulf clusters, however, the computations were bottlenecked by the 1 Gigabit network cables that connect the cluster nodes. Because the nodes must communicate with each other to facilitate parallelism, the communications must be fast for the cluster to reach maximum performance. The 1 Gigabit connections between the nodes used to make node to node communication relatively slow.

1.1.3 Project Objectives

In order to remedy this problem, the St. Olaf computer science department installed 10 gigabit fiber connections in the computer science network. This change increased the performance of St. Olaf's PDC resources. This paper describes our team's efforts to implement this transition. It will cover the changes in the network topology, our methods for and techniques, some analysis of the ethical issues that go into this new change, and some benchmarking which will establish the improvements in the new network.

1.2 Project Context

The work and research done on this project was carried out in fulfillment of St. Olaf's computer science capstone seminar. In this seminar, students, typically seniors, complete a semester-long group project. These projects are frequently interdisciplinary, and for many students, offer a chance to delve into specialized topics not otherwise offered through the St. Olaf course curriculum. In the case of this project, three project team members each filled specialized roles, including network topology, security design, and implementation testing, and were all working as cluster managers.

1.2.1 Cluster Manager Initiative

Several students in the St. Olaf computer science program are employed as "cluster managers". These students manage and develop specialty computing resources for the Math, Statistics, and Computer Science (MSCS) department. They also provide consultation and support for student and faculty research projects in CS and other departments, and implement enhancements to the CS network, which is managed independent of the college's IT network. The project detailed in this paper serves a wide range of equipment used by multiple departments, including including several 64-core machines, computers equipped with Intel Xeon Phi accelerators, a machine with 7 GPUs, and a multicore machine that hosts multiple virtual machines.

1.3 Prior Work

1.3.1 MistRider Cluster

Much of the work done during this project directly involved the MistRider cluster, a Beowulf cluster implemented at St. Olaf in 2008, which uses about 35 quad-core desktops with hard disks of a terabyte each [1], and is maintained by cluster managers. MistRider's nodes run on Xen Hypervisor virtual machines underneath the Fedora operating systems of its hosts, which consist of the "Link" machines available for students in the CS labs. This allows the Link machines to serve as cluster nodes as well as standard computer lab machines. The difference in performance between HPC on MistRider and HPC on a physical cluster is negligible, as asserted by experiments on MistRider done in 2011 [2]. In the summer preceding our project, cluster managers and IT staff laid the fiber cables running from MistRider nodes to a patch box in the server room.

1.3.2 Previous Network Version

Previously, the MistRider head node - the node that controls the other nodes - was connected to the cluster nodes via 1 gigabit network cables, as shown in Figure 1 below.

The head node is located in the CS department's server room and the cluster nodes are spread out in two classrooms. The relatively slow node to node communication caused by the 1 gigabit connection was a significant bottleneck in the MistRider cluster. The head node was also connected to the outside world through the CS Department's gateway machine. The gateway machine served IP addresses to the MistRider nodes using DHCP and allowed them to communicate to the internet with NAT. This man-in-the-middle configuration also slowed down MistRider's access to the internet. The updated MistRider configuration remedied both these issues.

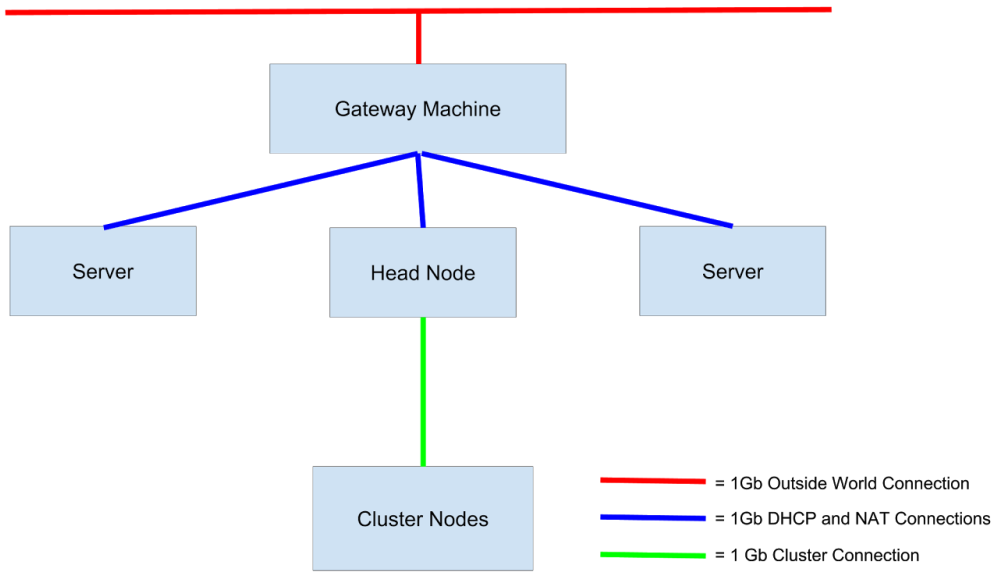

Figure 1: Original Topology of the MistRider Cluster

2 The New Network Design

2.1 Architecture and Topology

Our implementations required the use of two switches to route network traffic. Fortunately, an NSF infrastructure grant funded the acquisition of the two Cisco Nexus 9000 switches that we needed.

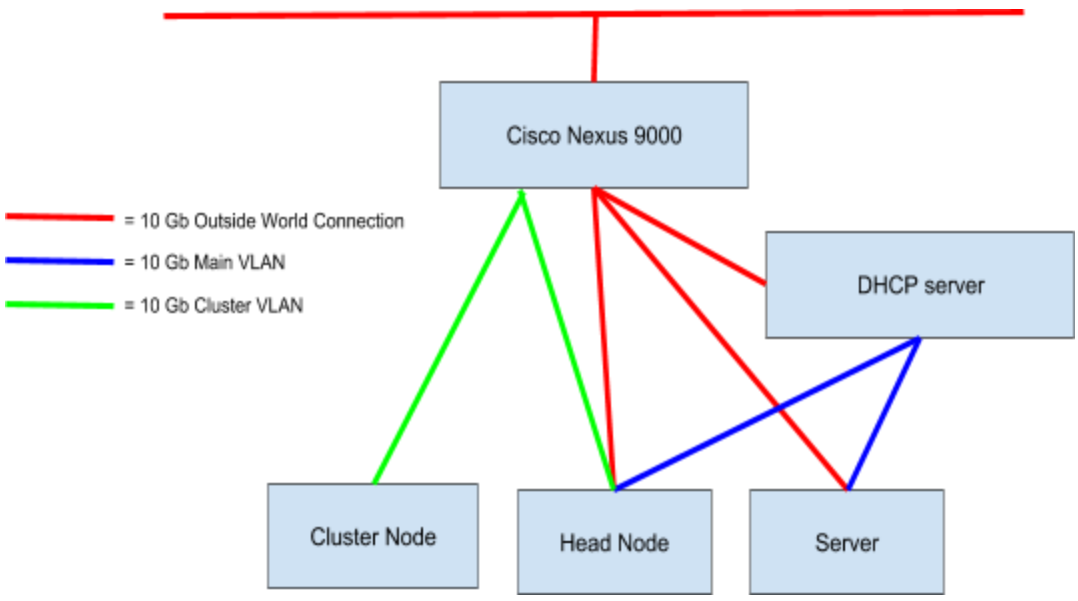

Figure 2: New Network Topology.

With this new topology, the cluster nodes will communicate over 10 gigabit connections which will facilitate faster parallel computing. The servers will also be able to connect with the outside world without routing through the gateway machine. Lastly, because the servers and head nodes will have global IPs, they can be accessible from anywhere in the world. This raises some ethical concerns that we addressed later in this paper.

3 Implementation and Process

3.0.1 Bridging Devices and Setting Up Networks

Since fiber cables had already been wired from the labs to a patch box in the server room, our first step consisted of connecting all the ports on the patch box to the Cisco Nexus 9000 Switch, setting a few static IPs and then confirming we had a connection between the different nodes. Then, since the head node for the MistRider cluster is a VM on one of our servers, we had to bridge the fiber port on the server to the VM head node and set up DHCP for all the nodes, making sure to keep the 1G network in place. For the rest of the fiber network, we bridged the second fiber port on the fiber card in our VM host to serve to all the VMs, along with connecting the other servers to the second Nexus switch. We also connected our DHCP server to the Nexus switch to be able to serve DHCP over the new network.

3.0.2 Fabric Discovery Error

One of the challenges that we had to overcome during this process was a Nexus 9000 bug which we will refer as the Fabric Discovery Error. When the switch was starting up, we encountered an error stating “fabric discovery in progress, show commands are not fully functional.”

This means that the switch became stuck when trying to connect to an existing network. This error greatly reduced the functionality of the switch, and we knew that we would have to fix it in order to use the switch for the fiber network. We solved this error by copying the operating system image from the first switch, uploading it to the broken switch, and setting the new image as the default using the loader> prompt. While future students are unlikely to encounter this specific error, we wrote documentation on this process as they may need to reset a switch or change its image.

Sometime after the Capstone course concluded, we began having more issues with StoFiber 2. We concluded that the switch itself was broken. While we were able to fix the fabric discovery error, there were other underlying issues that caused StoFiber 2 to be unuseable.

3.0.3 CSNet Gateway Machine Crash

In the middle of the semester, the hard drive of the gateway machine, through which all internet traffic was routed in the old network, unexpectedly failed. This suddenly monopolized the time of the cluster managers involved in the project and caused a delay. There was no recent backup of the gateway machine, so in order to resolve the problem, we migrated and confirmed working of all the network services from the old gateway onto a new virtual gateway machine.

3.0.4 Setting Up SFP+

Our project required enhanced small form-factor pluggable (SFP+) transceivers, which support up to 16 Gigabit/sec connections. By default, SFP+ ports do not function on the Ubuntu distribution. Typically Linux comes with the ixgbe driver installed in the kernel which enables the machine to use 10 Gigabit Intel network connections like SFP+. However, the Ubuntu driver only recognizes Intel SFP+ transceivers. This means that in order to use a transceiver from a third party, i.e. Fiberstore, then ixgbe must be configured to allow unsupported SFP+ transceivers. We first tried editing the ixgbe module configuration file with “allow unsupported SFPs”, then unloading and reloading the module using Modprobe, the program responsible for managing modules in the Linux kernel. This allows the interface to be used, but unfortunately only lasts until the system is turned off, so we found a more permanent solution. For the module to permanently

allow unsupported SFPs, we edited the grub default boot file to load the module with the specified configuration.

3.0.5 Installing Benchmark Software

In order to test the performance of our cluster over the new network, we decided to use the High Performance Computing Challenge (HPCC) suite of benchmarks. This benchmark suite appealed to us in part because MistRider had previously been benchmarked in 2012 with these same standards [1]. Prerequisites to the HPCC suite included implementations of both Message-Passing Interface (MPI) and the Basic Linear Algebra Subroutines (BLAS). To satisfy these requirements we chose OpenMPI and the Automatically Tuned Linear Algebra System (ATLAS).

The implementation of all these components, as well as getting them to work together for our purposes, proved to be an extensive experimentation and debugging process. At many points in this process, it was difficult to tell which component an error resided. For example, both OpenMPI and MPICH (MPI over CHameleon portability system) returned the same error when attempting to run the benchmark. During our debugging process, we wrestled with building and linking files, and configured a fresh installation of OpenMPI specifically to accommodate the HPCC benchmarks.

4 Testing and Benchmarking

In order to be able to present concrete evidence of the improvements made to the network, we used the Netperf network performance test as well as the HPCC suite of benchmarks, both standards in high performance computing (HPC) performance evaluation. HPCC consists of seven separate performance tests: High Performance Linpack (HPL); parallel matrix transposition (PTRANS); memory bandwidth (STREAM); matrix multiplication (DGEMM); Fast Fourier Transformation (FFTE); random access (RandomAccess); network bandwidth and latency (b_eff). Collectively, these benchmarks approximate the demands that a real-world or educational application might place on our system.

4.1 Results

Figures 3 through 7 detail the results of HPCC benchmarking over both the old copper (1G) connections and the new fiber (10G) ones. In both cases, the same nodes were examined so that the only variable would be the connection type. We also used Netperf for network performance benchmarking to supplement our results from HPCC.

It is easy to see from the data that the 10G fiber connections are a huge improvement over the 1G copper ones, in cluster performance as well as network performance. However the most noticeable result would be the Netperf results in Figure 7. The improvement in speed is markedly clear.

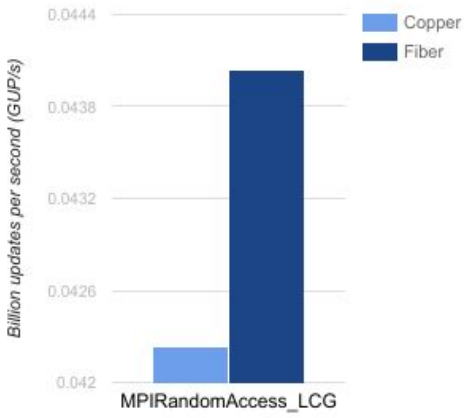

Figure 3: MPIRandomAccess_LCG.

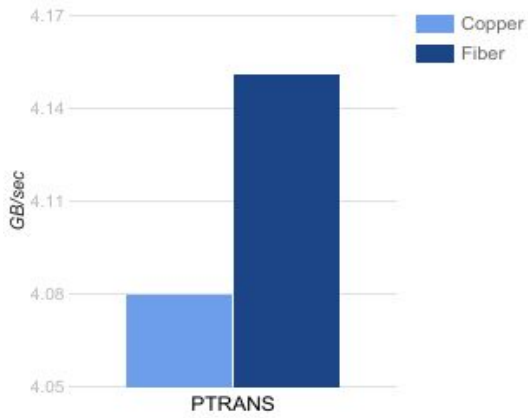

Figure 4: PTRANS.

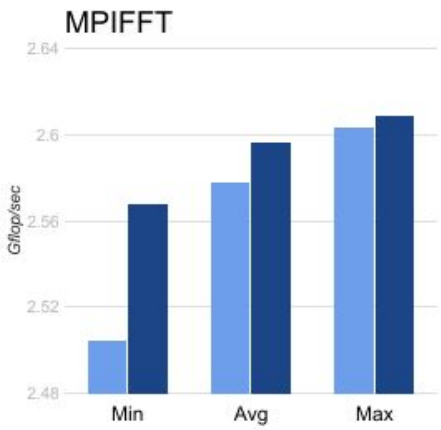

Figure 5: MPIFFT.

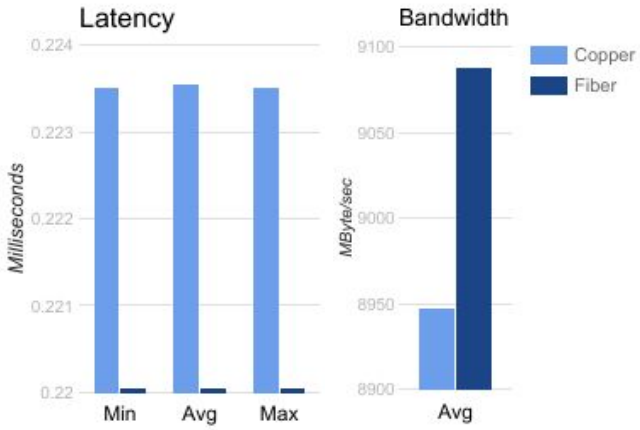

Figure 6: Latency-Bandwidth.

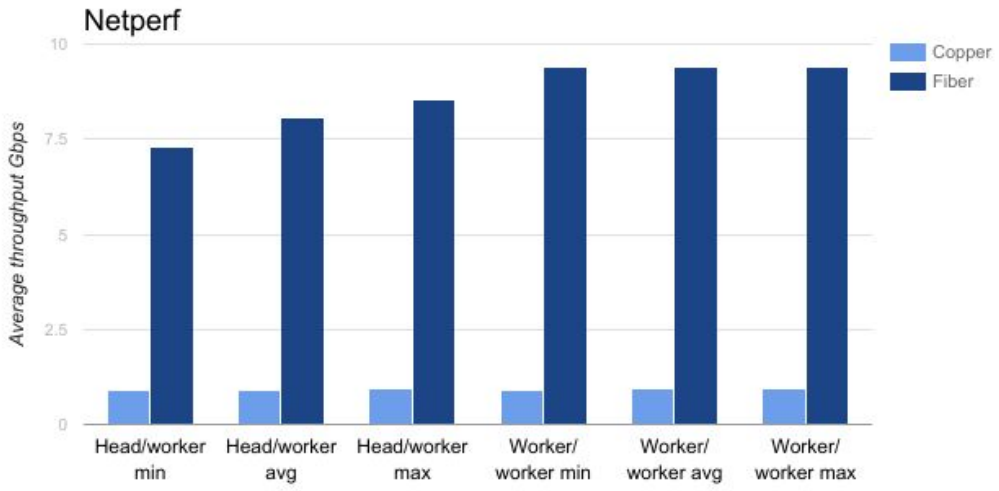

Figure 7: Netperf Network Performance Benchmark.

5 Security and Ethical Analysis

In the new network topology, all of the CS servers are now able to connect directly to the outside world and possess global IP addresses. This will enable the servers to connect to the outside world without the bottleneck of traveling through a NAT server, but it also raises some serious questions about the security of the computer science network. To get an idea of what considerations we should take into account, we consulted with St. Olaf College/Carleton College information security personnel Rich Graves, and our conclusions can be generalized in four categories.

5.0.1 Access Control Lists versus Firewall

The two most common means of controlling network traffic are access control lists (ACLs) and firewalls. ACLs contain a collection of IP addresses, ranges of IP addresses, and ports. Depending on the configuration of the ACL, these IP addresses are either allowed or disallowed from entering or exiting the network. Firewalls do everything that ACLs do, but they also provide information on the traffic that is entering the network. Firewalls are also capable of redirecting traffic rather than simply rejecting it. Both methods are commonly used, but ACLs are far less fully featured than firewalls and provide less control over traffic.

Currently, the fiber network only uses ACLs to control traffic. The Cisco switches that we use come equipped with ACL functionality, so we were able to implement them without relying on any auxiliary software. Furthermore, we decided early on that we would only allow a very specific range of IP addresses into the network. ACLs were satisfactory for this purpose. In the future, we would like to add a firewall to our network.

This would give us greater control of our network and would enable us to allow a wider range of outside world users into our network. In order to simplify the use of a firewall, we hope to use an automated IT program like Ansible to manage it. This would keep our firewall functional and effective without requiring cluster managers to spend too much time managing it.

5.0.2 Web Pages

The web pages hosted on the servers will be the most likely targets for attacks, most likely spamming or search engine optimization attacks. Search engine optimization attacks, or "SEO poisoning", consists of using search engine optimization techniques to get web sites infected with malware to prominent positions in online search results and steal users' sensitive data after their machines are infected. However, this can be easily avoided by keeping all software up to date, ensuring that the most current safety features are installed and active.

5.0.3 SSH Servers

The secure shell protocol (SSH) is used to operate network services securely over a network that may or may not be secured. In its most familiar form, SSH allows users to remotely log into servers. In our meetings with infosecurity personnel, we were assured that SSH is a very secure protocol, and for this reason, we needn't be extensively concerned about the security of our SSH servers. The biggest threats to these servers would be phishing attacks in which students or staff are lured into revealing their login credentials. However, if we are still concerned about risky traffic, we could easily configure ssh to use a port other than 22, and that would filter most random attacks.

5.0.4 Two-Factor Authentication

Our current password system is generally acceptable, but we could make it more robust by implementing some type of two-factor or multi-factor authentication. The most straightforward methods of doing this would be to send a temporary authentication code to a user's cell phone or to ask a security question previously set by the user upon registration.

6 Conclusion

6.1 Ongoing Work

Our team, with assistance from St. Olaf's IT department and other cluster managers are in the process of ensuring that all the nodes in the 10 GB cluster are functioning normally. We are also working to make more nodes in the cluster accessible to the outside world. By the end of the process, we hope to have a complete cluster functioning with 10 gigabit fiber connections. This cluster will also be accessible from the outside world due to its globally unique IP addresses.

At the conclusion of our project, the IT department decided that they would move some of the lab machines over to the CS network, and that cluster managers should be responsible for maintaining the lab machines in the CS network. While this will create more responsibility for cluster managers, it shows immense trust in the cluster managers as it takes a step forward, and tests the waters for similar improvements on the St. Olaf IT network in the future.

6.2 Final Words

St. Olaf's computer science program takes pride in being able to give its students experience in using cutting edge computing techniques like parallel and distributed computing. For this reason, it is important for the department to make sure that the clusters are using up-to-date technology. The introduction of 10 gigabit fiber connections into the MistRider cluster is a great step forward in ensuring that St. Olaf's PDC facilities are using the most current technology possible. The 10 GB network will enable St. Olaf's computer science students to get experience using powerful and relevant technology that they will encounter in the technology industry. Furthermore, it will allow St. Olaf to provide powerful cluster computing to both its students and its affiliate institutions.

References

[1] Elizabeth Shoop, Richard Brown, Eric Biggers, Malcolm Kane, Devry Lin, and Maura Warner. "Virtual Clusters for Parallel and Distributed Education." SIGCSE Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, 517-522. February 2012.

[2] Eric Johnson, Patrick Garrity, Timothy Yates, and Richard A. Brown. "Performance of a Virtual Cluster in a General-purpose Teaching Laboratory." IEEE International Conference on Cluster Computing. September 2011.