The Measure Polytope: Rapid Context Switching within a Universal Network Architecture.

Kurt Keville, MIT, {kkeville}@mit.edu,
Dan Vickery, MIT, {drv}@mit.edu

Overview

The search for a High Performance Computing Cluster that will work well or even sufficiently over a wide range of problems has been made difficult historically by the selection, or lack thereof, of a High-Speed Interconnect (HSI) that was appropriate to diverse applications with their implicit unique network needs. The Measure Polytope design attempts to resolve this through contextualizing jobs and virtualizing network devices on demand.

Background

High Performance Computing Clusters generally fall into one of a small number of classes. These classes break down mostly under associations with the problem they are designed to solve. Some problems are CPU-bound by nature and some are network-bound. Due mostly to phenomena associated with Moore's Law, many parallel programmers in this space have designed to code their problem in such a way as to make the network the bottleneck to performance increases. This problem is easier to build around. There are certain problems that are "embarrassingly parallel" and do not really demand much of the network. For these problems, standard Ethernet seems to be sufficient. But unfortunately, most real-world problems in research (notably those that use CFD and QCD codes) usually do not fall under this heading. What is needed is a cluster design that will accommodate a wide range of network needs; low latency and / or high bandwidth, specific to the code being run.

Test Methodology

We put together a 64 PE system comprised of 8 Quad-core AMD Opteron™ SMP server systems in order to perform throughput tests. Each node was the same; it was fully populated with RAM and was running the latest version of MPI/GAMMA on the first NIC. This embedded micro-cluster is very dense; we have no moving parts, the node uses a RAMdisk rather than a hard drive and boots from the net. Additionally, we replaced the CPU fans with heatpipes.

Low latency commodity interconnect research has generally revolved around Active Messages, VIA, and other OS-bypass methods. Certainly much research has been done in the area of separating high-bandwidth and low-latency concerns if possible. In the proprietary interconnect world, much more so that the commodity world, RDMA has been investigated thoroughly, as has various improvements to InfiniBand. The appearance of HyperTransport™ onto the clustering scene has brought with it a dizzying array of interconnect options. It is now conceivable (at least on dual CPU motherboards) that you can dedicate a HyperTransport port to your network without giving up one that could otherwise be used somewhere else more important; we can dedicate a port to the memory bus, a port to the other CPU, and a port to the PCI-e bus that our GPU and NICs are on with little contention for bandwidth.

We are running MPI/GAMMA on a single NIC (this project no longer supports commingling IP on the GAMMA NIC) and Advanced Sparse Flat Network Neighborhood (SFNN) protocols on the remaining NICs. The latter package allows us to bring devices up and down on demand and designate channel-bonds without the use of expensive switches.

We have run the standard NAS benchmark suite and found remarkable performance gains in the initial tests. We are using inexpensive 8 port GigE switches that support jumbo frames on our FNN. Initial testing indicates that we are getting better performance than we expected from this configuration across the full range of NAS tests.

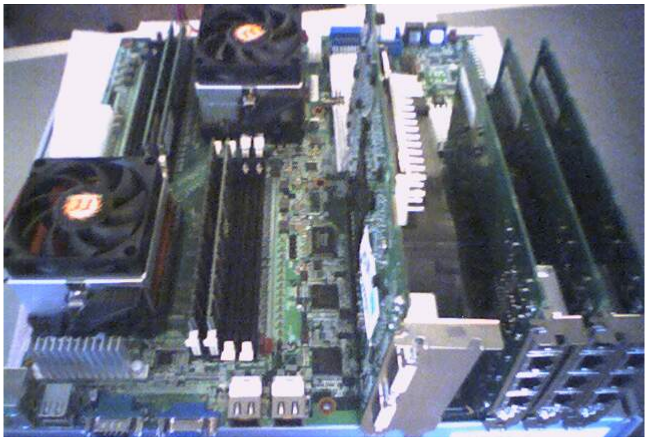

Figure 1: A measure polytope node has some number of NICs appropriate to the cluster scale with the first NIC being used for the most latency sensitive traffic.

Cluster Design

The 8 nodes are currently mounted on shelving that allows us easy access to the cards and connectors. After testing, this will be further compressed into a dense embedded cluster. At that point, we will have a case approximately 18 inches cubed that will represent a stackable 64 PE microcluster that you can use as a basis for a larger modular cluster. The cluster currently uses 5 8-port GigE switches, 1 for the latency sensitive traffic and 4 for the SFNN traffic. We have tested that path extensively and can guarantee full pipe usage (4 GBps) over a range of scenarios. Our current coprocessor solution uses our NV40-based GPU. In particular, we are using the GPUFFTW libraries with it. We anticipate replacing this solution with the HyperTransportcompatible FPGAs from DRC that plug directly into an Opteron Socket (940).

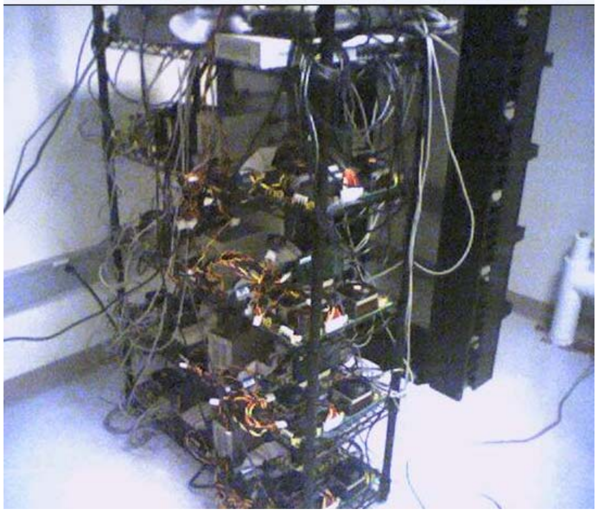

Figure 2: The measure polytope cluster currently uses shelving instead of PC racks since the evaporative cooling solution uses the existing Data Center A/C for the totality of it's cooling.

Conclusions

The full range of benchmarks will be run on this configuration this summer. In particular, the HPC Challenge benchmarks will be run to demonstrate the value of the GAMMA protocol in conjunction with FNN. We will publish these results here.

References

References were not included as part of this abstract.

http://www.disi.unige.it/project/gamma/

http://aggregate.org/SFNN/

http://gamma.cs.unc.edu/GPUFFTW/

http://drccomputer.com/