

Early Experiences with the Myricom 2000 Switch on an SMP Beowulf-Class Cluster for Unstructured Adaptive Meshing

Charles D. Norton and Thomas A. Cwik
Jet Propulsion Laboratory
California Institute of Technology
MS 168-522, 4800 Oak Grove Drive
Pasadena, CA 91109-8099 USA
Charles.D.Norton@jpl.nasa.gov

Abstract

We explore the current capabilities of the recently released Myricom 2000 switch, using MPICH-GM for communication, on a 2-way SMP Pentium III Beowulf-Class cluster. Performance measurements indicate that data transfer rates of approximately 225 Mbytes/s with 9.3 μ seconds latency for ping-pong tests can be achieved for messages as large as 32 Mbytes. When shared-memory communication is used approximately 130 MBytes/s with 1.5 μ seconds latency for long messages (250 MBytes/s peak) is possible. The performance varies depending on how processors communicate; either within an SMP node or across nodes. Performance for parallel unstructured adaptive refinement of 3D tetrahedral meshes shows noticeable improvement when compared to 100BaseT Ethernet. Furthermore, when compared to traditional systems such as the SGI Origin 2000, the combination of this fast network with high performance SMP processors demonstrate that Beowulf-Clusters compare favorably with such systems—even for communication intensive applications.

1. Introduction

Beowulf-Class clusters have demonstrated that scalable parallel computing can be achieved, at low cost, through the use of commodity off-the-shelf (COTS) parts. The COTS approach allows one to configure a system using the latest hardware and software that is available, and affordable, at the time. As time goes by, one has the option to reconfigure a system as new products are announced and released. Many cluster components may be freely available, such as system software, while others, including advanced microprocessors, are constantly under competitive pricing pressures to remain affordable.

The network interconnect, however, allows great flexibility regarding the communication performance one expects versus the amount one is willing to pay. Although 100BaseT Ethernet cards are not expensive the approximate 11 MBytes/s upper bound can hinder the performance of clusters containing many fast SMP processors with large amounts of main memory. Nowadays, improvements in system software and microprocessor performance bring added pressure to consider fast networking to maintain a balanced system.

The JPL High Performance Computing Group maintains a series of clusters. Our most powerful cluster contains 26 compute nodes, and one front end node, of dual-processor 800 MHz Pentium III's—a total of 54 processors in all. Each node has 2 GBytes of RAM available giving a system with 104 GBytes of main memory with 41.6 GFlops of computation. We recently replaced our 3COM SuperStackII switch and 100BaseT Ethernet network with Myricom's new 32-port Myricom 2000 networking hardware. More details about Myricom's technology including architecture specifications, software, algorithms, and products are available at their web site and elsewhere [1, 6].

We will explore our experiences and evaluate the performance impact this hardware introduces for our system. The results will be compared to experiences before the new network was introduced, and to a more traditional system (the SGI Origin 2000) for adaptive meshing simulations that typically stress CPU, memory, and communication performance.

Please note that we have worked closely with Myricom to resolve hardware and software issues that affect performance. Our most recent results are given, but further updates may be presented at the conference. Please contact the authors for more information.

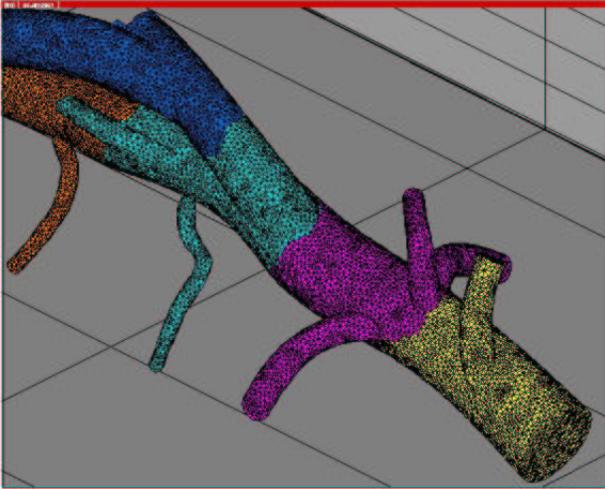


Figure 1. Repartitioning and migration of artery mesh segment using PYRAMID AMR Library.

2. Characterizing Communication-Intensive Applications

Many physics-based numerical applications are fundamentally irregular, meaning that the solution process is largely determined at run-time and the communication requirements are non-uniform, although generally predictable. Parallel adaptive methods fall into this category and software for these techniques require very high performance systems for large problems.

We have developed software to handle parallel unstructured adaptive mesh refinement for finite element applications [5, 7]. This tool allows large triangular and tetrahedral meshes to be loaded, adaptively refined with automatic mesh quality control, load balanced, and migrated among the processors using high level object-based library commands. Since parallel adaptive mesh refinement potentially involves working with many millions of elements great effort is used to minimize communication and to ensure that transmitted messages are as large as possible. On a cluster, managing communication becomes even more important since applications may use networks that are significantly slower than those found on most traditional super-computing systems.

Figure 1 shows a segment of a large tetrahedral artery blood flow mesh segment. The original geometry was provided by Taylor et. al [8] and the initial mesh was generated by the Scientific Computation Research Center at Rensselaer Polytechnic Institute [2]. The mesh contains 1.1 million elements where our PYRAMID adaptive mesh refinement library was used for repartitioning, load balancing, and

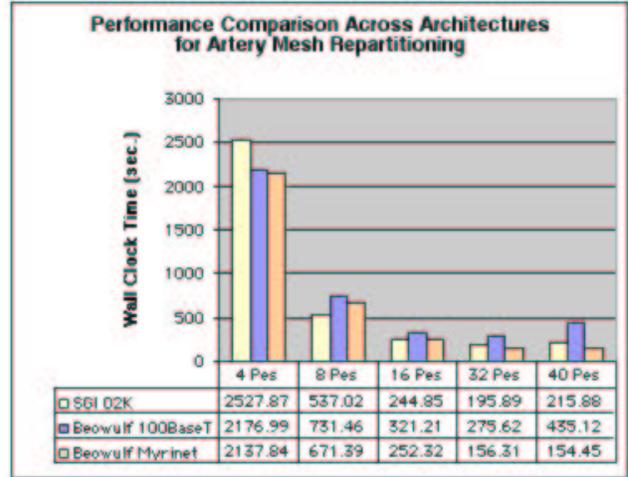


Figure 2. Performance for communication bound mesh loading, repartitioning, and migration of the artery mesh.

mesh migration. Processor partitioning is also indicated.

Mesh migration performance comparisons for this problem, using 100BaseT Ethernet and Myrinet on the cluster as well as the NUMA architecture of the SGI O2K, indicate the benefit of Myrinet as shown in figure 2. This is largely a communication-based benchmark. As we will see, however, the performance tradeoffs vary based on the problem solved. Nevertheless, this initial benchmark indicates that the Beowulf cluster can compete on par with traditional systems even for communication intensive applications.

3. MPICH-GM and Linux

Although cluster technology has generally stabilized integrating new components into an existing system can cause problems. At this writing, our Beowulf cluster is based on Redhat Linux 6.2 with Kernel version 2.2.19 SMP using MPICH-GM version 1.2.1..4 released in July of 2001. Previous versions of MPICH-GM had problems, but the most recent release has good stability with improved performance. Both the low level GM communication subsystem and Myricom's version of MPICH, called MPICH-GM, have been improved. Although work continues Myricom support has been very active in working with us to address and correct problems.

More specifically, we are using gm-1.4.1pre14 with the Myrinet M3E32 Switch, PCI 64B, Lanai 9 with 4 MB SDRAM. Our SuperMicro SUPER 370 DLE Motherboards use the ServerWorks ServerSet LE rev 5 chipset. There have been reports of memory corruption problems with the ServerWorks chipset and Myricom discovered this when ex-

aming our system. The memory itself is fine, but when running beyond physical memory the data returning from the disk due to swapping can get corrupted. As a temporary measure Myricom disabled DMA usage for all of the IDE controllers on all of the nodes. We have reported this problem to the motherboard vendor.

Our processors were installed and rack mounted by the same vendor that configured and installed the RedHat Linux kernel. Unfortunately, we did not immediately realize that certain configuration options required by our system were not specified correctly. In fact, the highmem region of memory was not addressable so we experienced unnecessary memory swapping that also interfered with the memory requirements of the GM communication layer. Most of the defaults for kernel version 2.4.x satisfied our system requirements, but the recommended kernel for MPICH-GM (version 2.2.x) did not. Extra details required in the kernel configuration included specifying that we had SMP processors, that the amount of real memory was 2GBytes, and that the highmem region of memory should be addressable.

Previous versions of MPICH-GM required users to specify a number of performance related parameters regarding memory registration, shared-memory support, and so on. The new version hides these details and has other useful features, such as the capability to decide at run time if shared-memory communication should be used. The ability to have shared-memory communication as an option was important for PCI chipsets that exhibited poor performance since this option often improved performance. For our system, the peak PCI bandwidth is 455 MBytes/s which is very good. On older systems this was often a “hidden” bottleneck to network performance since the PCI could be as low as 125 MBytes/s. We experience better performance for large messages when shared-memory is not used.

Figure 3 shows the results of a network ping-pong test on two processors for the Myrinet installation compared to previous results for 100BaseT and the SGI Origin 2000. Incidentally, the MPI implementation on the Origin uses global shared memory to implement message passing. When a processor requires data the packets are sent over the CrayLink so the latency to access memory becomes a critical part of a performance metric on this machine in addition to good cache management.

The improvement for our cluster is significant where neighbor processors that are not on the same board are used. This certainly had an impact on the artery mesh migration problem in figure 2. While we are very close to peak speed for 100BaseT we also achieve near peak speed for MPICH-GM on the Myricom 2000 hardware (rated at 2 Gbits/s) for ping-pong tests. The average latency is still quite low, about 1.5μ sec. for processors that share a CPU board and 9.3μ sec. for processors on separate CPU boards. The Myrinet result in figure 3 uses processors on separate boards

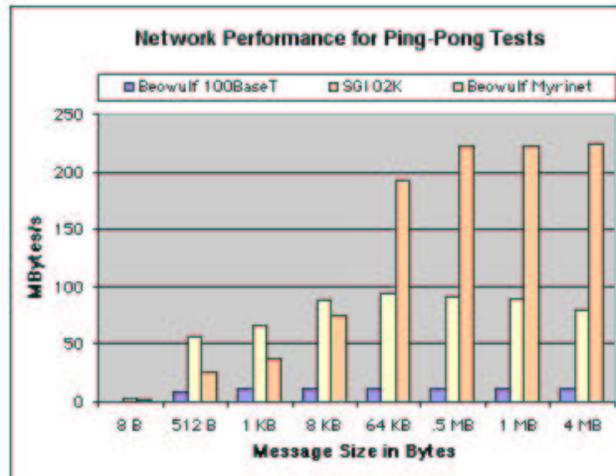


Figure 3. MBytes/s transmitted in ping-pong tests between two processors.

so shared-memory communication is not an issue.

Earlier systems have reported an overhead associated with MPICH over GM, and that the MPI implementation can take away as much as 23% of the peak bandwidth of Myrinet [3]. Other benchmarks also indicated that 149 MBytes/s with 10μ sec. latency have been measured with older versions of the Myrinet hardware [6] so our new hardware and software does show a big improvement over these past results.

4. Evaluating Functionality and Performance

Our primary interest is to examine the effect of a network upgrade for a communication intensive application. However, before examining how the Myricom 2000 hardware performs on our adaptive meshing simulations we should take a closer look at network performance in an SMP environment.

The MPICH-GM results in figure 4 show that good performance is possible, particularly for large messages. There is a cross-over point where communication between processors on the same CPU board (node) falls behind processor communication across boards for messages larger than about 32 KBytes. This is primarily due to cache effects since the receiving processor can access near by data directly from the cache, instead of from the shared-memory region, but only up to a point. Nevertheless, for our adaptive meshing problem we regularly send messages as large as 35 MBytes in size and we do not use shared-memory communication in MPICH-GM so our simulations will select processors one at a time, one board at a time.

Regarding the `ch_p4` device, commonly associated with

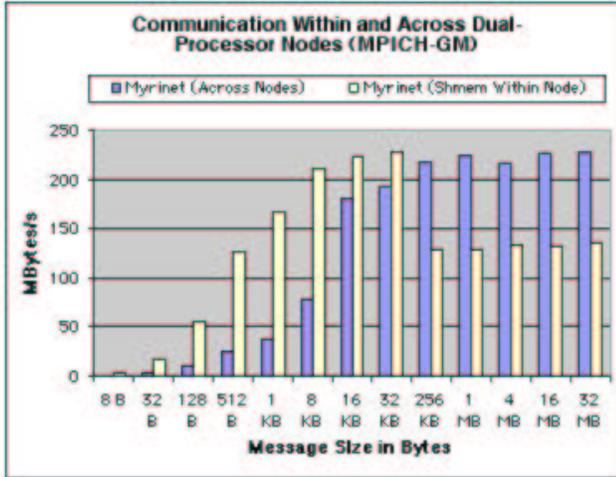


Figure 4. MBytes/s transmitted in ping-pong tests between two processors on the same node (using shared-memory communication) and across nodes using MPICH-GM.

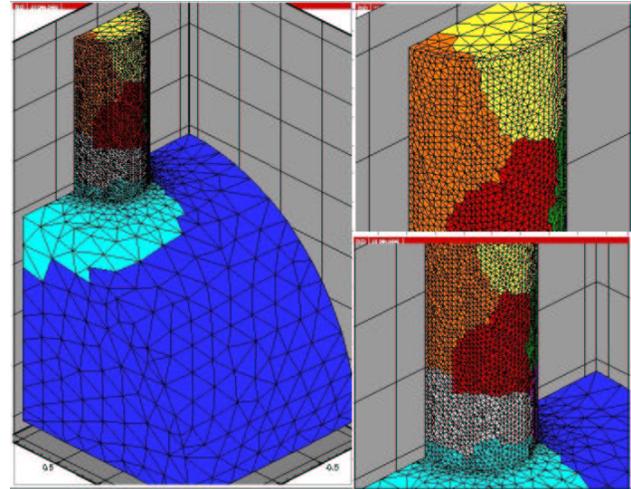


Figure 5. Muzzle-brake shock tube mesh with initial mesh partitioning and redistribution among eight processors.

the Ethernet IP protocol, one can actually send such traffic over Myrinet if desired. This is generally a matter of building MPI with the `ch_p4` device and ensuring that the Myrinet configuration files and system host tables are set up properly. Although this gives an improvement over the Ethernet cables, and the performance is essentially independent of the processor configuration chosen, the main advantage of setting up the Ethernet IP is for system maintenance. We have found it very useful to have a backup network in such circumstances.

4.1 Analyzing the Muzzle-Brake Mesh

Figure 5 shows a muzzle-brake shock tube mesh with its initial partitioning and redistribution among processors. We repeated the simulations from [7] using the new Myricom 2000 and have included the combined results in figure 6. The initial mesh contains just 34,214 elements, but after three adaptive refinements it contains 1,264,443 elements. Figure 7 shows this mesh after the first refinement.

What is clear is that for a small number of processors the Beowulf cluster performs much better than the Origin for this problem. As the number of processors is increased all configurations show some scalability, but the Origin outperforms the cluster. One would suspect that a fast Myrinet network would show an even greater improvement over the 100BaseT Ethernet. This is misleading, however, because for this specific mesh the communication performance is not dominant.

A breakdown of the time spent in mesh migration, which

includes partitioning and load balancing, compared to creating new elements by adaptive refinement shows that much more time is spent in the AMR process. On the Beowulf cluster using Myrinet $\sim 182s$ and $\sim 46s$ are spent in creating new elements and migrating them respectively. Similarly, $\sim 98s$ and $\sim 22s$ are respectively spent in these stages on the Origin. In fact, once the coarse elements have been redistributed the new elements are created locally so no communication is required. This implies that improvements in the network will not significantly impact overall performance for this specific problem.

4.2 Analyzing the Earthquake Mesh

Figure 8 shows the performance for an earthquake mesh generation where communication is more dominant. These results show a large performance improvement for the cluster under Myrinet. The initial mesh only contains 1,316 elements where 554,141 elements are created after three refinements. This example shows how a change in the problem description can impact performance for applications that have irregular characteristics.

Another important point, however, regards improvements in algorithm design for communication on clusters. This is also shown in figure 8 where the migration time is measured for 8 processors based on old and new communication algorithm techniques. The original algorithms set up communication schedules based on processors sending and receiving messages directly as needed. That approach assumed that good performance can be achieved by matching communication operations exactly. Since the communica-

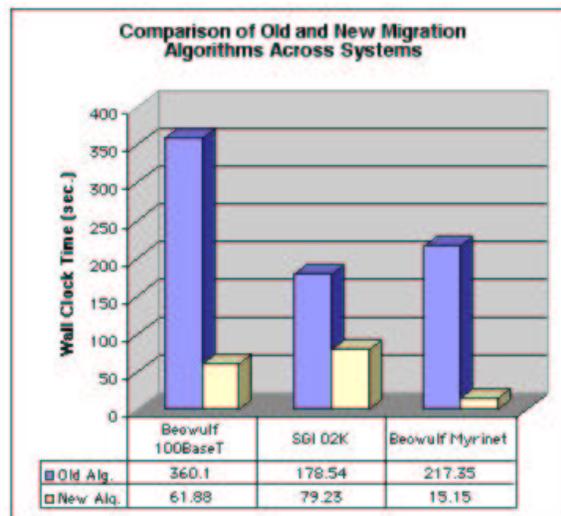
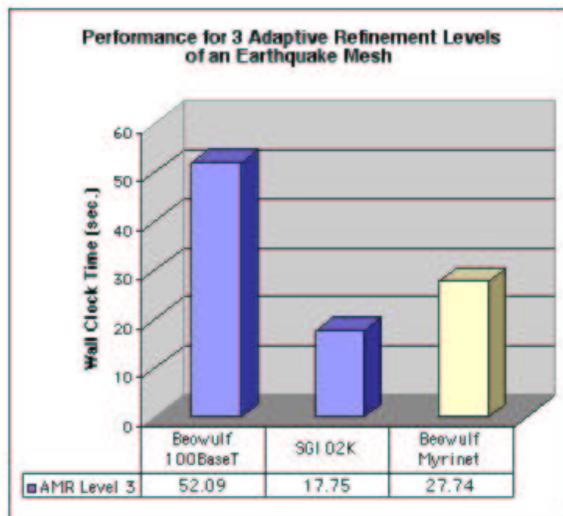


Figure 8. Performance after three adaptive refinements of an earthquake mesh among 32 processors where the Myrinet network upgrade affects overall performance. Also shown is the migration stand-alone timing due to message passing algorithm improvements for 8 processors where various performance comparisons can be derived.

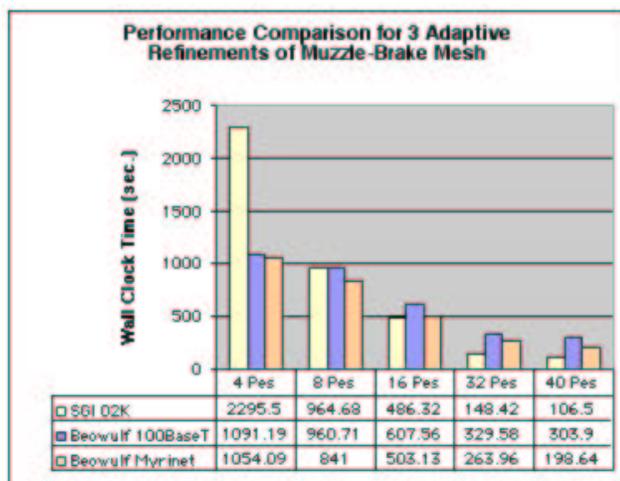


Figure 6. Performance after three adaptive refinements of the shaft section of the muzzle-brake mesh.

tion schedule is irregular processors managed non-uniform message passing activities, but the volume of data transferred was limited to only what was required among processors that communicate.

The new communication algorithm is much more scalable for large systems in that a tree-based pair-wise exchange algorithm is used. This algorithm works on any

number of processors, not just a power of two. It is unique in that it guarantees that a minimal number of exchanges will be performed—meaning that the algorithm can determine if a processor has already received the data it needs and skip communication operations as necessary. The volume of data tends to grow with such algorithms as exchanges are performed, but this algorithm can also determine when data need not be included in future pairwise exchanges and it will remove such data from future operations. It is designed to take advantage of networks supporting full-duplex communication.

Although for tightly-coupled networks this algorithm will work well, it is very well structured for clusters that have slower networks, as seen in figure 8 for the 100BaseT Beowulf network. When Myrinet is applied the results are even more dramatic. All of the performance results in this paper are based on using the new communication algorithms.

4.3 Analyzing the Artery Mesh

Returning to our artery mesh figure 9 shows the adaptive refinement of a small section that creates 1.8 million elements from the initial mesh of 1.1 million elements. Figure 10 shows again how performance can vary between the Myrinet-based Beowulf cluster and the SGI Origin 2000. In this example we used 50 processors since the GM messaging layer would occasionally detect errors in message send operations when all 52 processors were used.

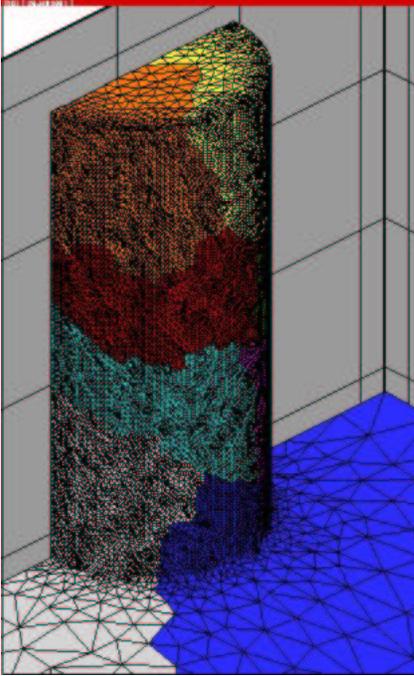


Figure 7. Mesh structure with partitioning after adaptive refinement of the muzzle-brake.

The first instance in figure 10 adaptively refined a small region and analysis showed that the time spent in migration and adaptive refinement were nearly equal across both machines. In the second instance the region refined was increased, creating about 4.5 million elements. For the cluster both the migration and adaptive refinement time increased significantly, but more time was spent in adaptive refinement than migration. This pattern was also true for the SGI Origin, but both migration and refinement were faster than on the cluster.

In the third instance the entire artery was refined creating 7.4 million elements. In this case, where one would expect the cluster to perform poorly, it actually outperformed the SGI Origin. Analysis showed that the migration time for the cluster was slower than for the Origin while the adaptive meshing time was faster for the cluster than for the Origin. Overall, this gave the cluster better performance in this case.

We also examined the message passing structure more closely for these problem instances. Although the new pairwise message exchange algorithm was used for mesh migration we decided to analyze the mapping for processor communication exactly. This included measuring, for each processor, the number of other processors that need to send data as well as the number of elements each processor must receive in order to achieve load balance during migration. Neither of these measurements allowed specific conclusions

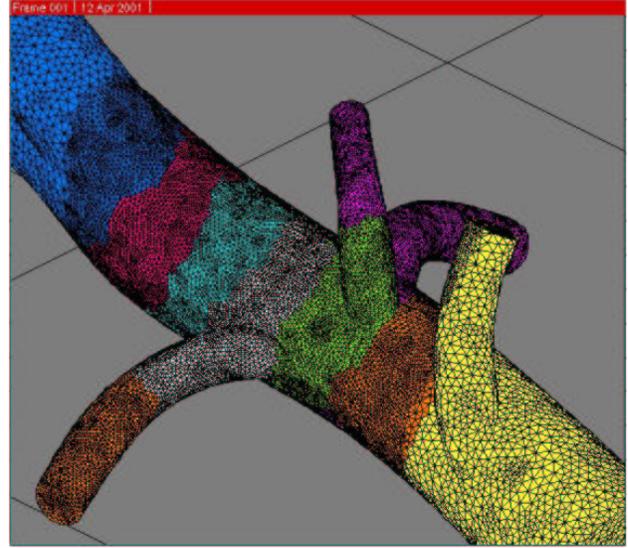


Figure 9. Adaptive refinement, repartitioning, and migration of the artery mesh segment containing 1.8 million elements.

to be drawn about the performance differences among these problems cases. In fact, the data interaction was fairly well balanced for all problem sizes under both architectures. There were instances, however, where the number of elements a small number of processors must receive for load balance was about 5 times larger than the average, but this was characteristic of all test cases.

This would suggest that characteristics of the problem may determine performance much more than features of the network, and that for irregular problems this is difficult to characterize.

Most people are aware that the balance between CPU speed and the time to fetch data from memory does play a role in performance. The pipelined super-scalar architecture of the R12000 processors on the Origin are only clocked at 300 Mhz, but the processor to memory communication interaction is very well balanced leading to good single-node performance. The Pentium architecture, historically, has not been balanced as well potentially causing memory accesses to fall behind high CPU clock rates. This has been seen in calculations involving highly structured matrix operations where the ability to control how data is accessed can be controlled. Although adaptive meshing is very irregular by nature, the possibility exists that for certain problems the CPU to memory interaction dominates performance more than the network on a cluster.

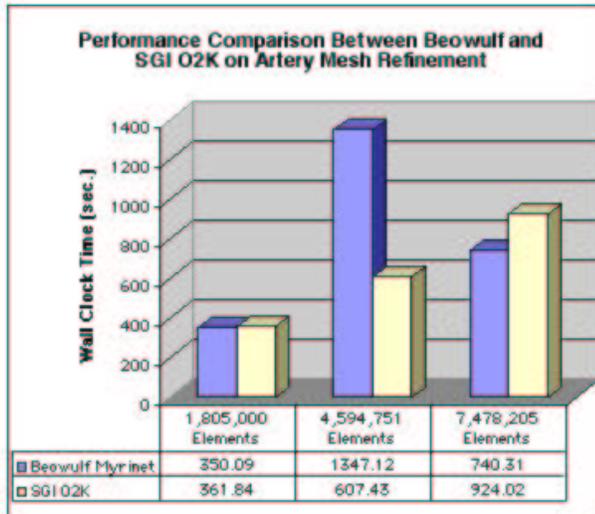


Figure 10. Performance comparison for adaptive refinement of artery mesh for various region sizes on 50 processors.

5. Conclusion

As one would expect, the upgrade of a network mainly benefits problems with large message passing requirements. On point-to-point ping-pong tests using Myricom's MPICH under the GM messaging layer we can achieve near peak performance, with low latency, for sufficiently large messages. On an SMP where shared-memory communication is possible, we also observed a crossover where communication on the dual-CPU board falls behind communication across boards for large messages. Ultimately the ability to sustain network performance for large messages is more important for users running large applications.

Another useful performance measurement is the performance of the network where multiple pairs of processors are exchanging messages simultaneously. This indicates network performance under a load. Figure 11 shows results of this test for 16 processors simultaneously using Myrinet and MPICH-GM for ping-pong and full-duplex message exchanges. Our new migration algorithms perform such pairwise exchanges.

Figure 11 shows a measurement of the bisection bandwidth for various message sizes across multiple numbers of processors. The data has been normalized by the number of processors in a partition. A range of performance variations can be seen based on the number of processors used in the simultaneous communication. For large messages the performance is good for a small subset of processors, but as the number of processors increases (implying more traffic on the network), the performance drops. This is likely a contribut-

ing factor to the performance of our adaptive meshing application which performs exchanges for large messages. In these tests shared-memory communication is not applied.

One caveat regarding performance comparisons between the Beowulf SMP Cluster and the SGI Origin 2000 for our adaptive mesh problem is that the simulation results are not precisely identical. In particular, the ParMetis partitioner applied in the dynamic load balancing stage produces different partitionings on each machine [4]. (This is a side-effect of the random numbers used in the partitioning algorithms.) This will affect the adaptive refinement process and may affect comparative timings, but it should be not too significant.

Finally, it is interesting to compare how well commodity networks, such as Myrinet, compare to highly rated machine specific networks such as the Cray T3E. Figure 12 gives this comparison which is reasonably good for a commodity cluster with Myrinet 2000. Although the Cray T3E bandwidth is higher the latency is also higher measured at 34.71μ sec. compared to 9.3μ sec. for Myrinet.

Installing updated MPICH-GM software had a large impact on our system reliability and stability. We still experience PCI-related data corruption on occasion so replacing the motherboards is an issue we must address. In the end, the network is just one contributor to the combination of factors that affect performance and usability of the cluster. Our experience is that good network performance can be achieved using Myricom 2000, but the effective benefit for applications depends largely on their characteristics.

6. Acknowledgment

We appreciate several helpful discussions with members of the JPL High Performance Computing Group, JPL Supercomputing Group, and the Myricom Technical Support team. This work was performed at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

References

- [1] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W. Su. Myrinet – A Gigabit-per-Second Local-Area Network. *IEEE Micro*, 15(1):29–36, February 1995.
- [2] Joseph E. Flaherty and James D. Teresco. Software for Parallel Adaptive Computation. In Michel Deville and Robert Owens, editors, *Proc. 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, Lausanne, 2000. IMACS. Paper 174–6.
- [3] J. Hsieh, T. Leng, V. Mashayekhi, and R. Rooholamini. Architectural and Performance Evaluation of GigaNet

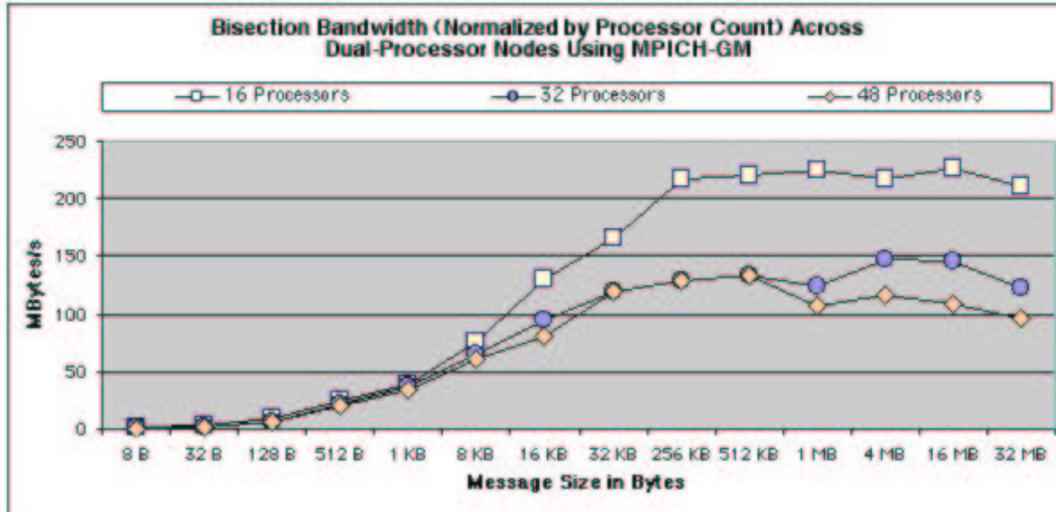


Figure 11. Bisection bandwidth measurements by processor where the aggregate bandwidth is normalized by the bisection partition size.

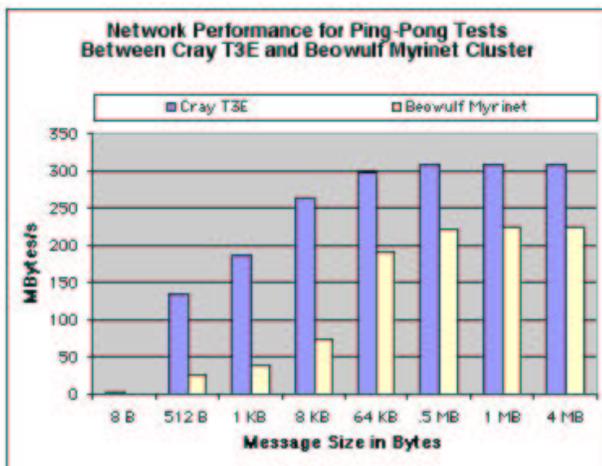


Figure 12. Performance comparison between Cray T3E and Pentium-III Beowulf cluster for ping-pong tests between two processors in MBytes/s.

and Myrinet Interconnects on Clusters of Small-Scale SMP Servers. In *Proc. SC'2000*, Dallas, Texas, November 04–10 2000. IEEE Computer Society. CD-ROM.

[4] G. Karypis, K. Schloegel, and V. Kumar. ParMetis: Parallel Graph Partitioning and Sparse Matrix Ordering Library. Technical report, Dept. of Computer Science, U. Minnesota, 1997.

[5] J. Z. Lou, C. D. Norton, and T. Cwik. A Robust Parallel Adaptive Mesh Refinement Software Package for Unstructured Meshes. In *Proc. Fifth Intl. Symp. on Solving Irregularly Structured Problems in Parallel*, 1998.

[6] Myricom, Inc. *Myricom Creators of Myrinet*, 2001. <http://www.myri.com>.

[7] C. D. Norton, J. Z. Lou, and T. A. Cwik. Status and Directions for the PYRAMID Parallel Unstructured AMR Library. In *15th International Parallel and Distributed Processing Symposium*, San Francisco, CA, April 23-27 2001. Irregular 2001 Workshop. CD-ROM.

[8] Charles A. Taylor, Thomas J. R. Hugues, and Christopher K. Zairns. Finite Element Modeling of Blood Flow in Arteries. To appear, *Comp. Meth. in Appl. Mech. and Engng.*, 1999.