Res. Lett. Inf. Math. Sci., 2003, Vol.**5**, pp 1-10
Available online at http://iims.massey.ac.nz/research/letters/

Performance Characteristics of a Cost-Effective Medium-Sized Beowulf Cluster Supercomputer

A.L.C. Barczak, C.H. Messom & M.J. Johnson

Institute of Information & Mathematical Sciences
Massey University at Albany, Auckland, New Zealand.

This paper presents some performance results obtained from a new Beowulf cluster, the Helix, built at Massey University, Auckland funded by the Allan Wilson Center for Evolutionary Ecology. Issues concerning network latency and the effect of the switching fabric and network topology on performance are discussed. In order to assess how the system performed using the message passing interface (MPI), two test suites (mpptest and jumpshot) were used to provide a comprehensive network performance analysis. The performance of an older fast-ethernet/single processor based cluster is compared to the new Gigabit/SMP cluster. The Linpack performance of Helix is investigated. The Linpack Rmax rating of 234.8 Gflops puts the cluster at third place in the Australia/ New Zealand sublist of the Top500 supercomputers, an extremely good performance considering the commodity parts and its low cost (US$125000).

---

1 Introduction

The Helix is the newest and most powerful supercomputer (Beowulf cluster) in New Zealand. Its hardware was funded by Alan Wilson Center for Molecular Ecology and Massey University a centre of research excellence established in 2002. The primarily purpose of the Helix is to run Bioinformatics programs such as BLAST for whole genome searches, however secondary users span the fundamental sciences, mathematics and computer science. Having built an experimental cluster three years ago (1), the Center for Parallel Computing at the Institute of Information and Mathematical Sciences (IIMS) was given the opportunity to design and build a significantly larger cluster. Several research groups in mathematics, computer science and bioinformatics within New Zealand would benefit from such a machine. The budget was extremely limited and it was necessary to choose hardware that would give the best price/performance possible. To evaluate the price/ performance ratio both the floating point and integer performance were considered. Many benchmarks are available today (see (2)) but few are able to characterize a cluster in a useful and reproducible way. Tests are often misleading, Gropp and Lusk discuss many of the traps found in performance measurement (3). The Linpack benchmark was chosen as the floating point benchmark as it would test the scalability of the Helix rather than just the raw peak performance. The integer benchmark was not required to scale as many bioinformatics applications that primarily make use of integer operations are "embarrassingly parallel", that is, they require little interprocess communication, meaning that they scale well. As well as the macro performance of the machine as illustrated by the Linpack benchmark, we were also interested in the micro performance of the Helix system. In order to do this we used the 'mpptest' test suite developed by Gropp and Lusk (3). Using these programs to measure different characteristics of the cluster has lead to some interesting observations. A ring test was also carried out, showing that the topology has virtually no effect on point to point operations.

1.1 Background

With the availability of cheap commodity processors and network interconnects, the cluster architecture for supercomputers has become popular. As commodity processors have increased in cock speed and are able to simultaneous execute multiple floating point instructions they have started to rival the performance of dedicated scientific processor architectures. The programming techniques suitable for networked computers are fully discussed in (4) For several years researchers have been investigating the clustering of legacy hardware using discarded pcs, servers. and network cards(5), (6). These systems can produce good performance figures for embarrassingly parallel applications, but suffer from scalability problems due to small memory, low bandwidth and high latency interconnects. An interesting study of the effects on applications and the sensitivity to differences in bandwith and latency is found in (7). With recently released processors such as the Intel Xeons and AMD Athlons, cheap RAM and commodity gigabit Ethernet with reasonable latency it has become possible to build scalable high performance computer system.

2 A brief description of the clusters

2.1 The Sisters

The Sisters was the first Beowulf cluster built at Massey University. It consists of 16 nodes connected by a fast Ethernet network. The nodes use Pentium III CPUs running at 667MHz with 256MB Ram. The server is a dual Pentium III with 1GB Ram. There is a single switch linking the main server and the nodes. Key bottlenecks in the system include the high latency and low bandwidth network interconnects.

2.2 The Helix

Given the requirement of a homogeneous cluster for the Helix, the main decision was whether to use Xeon or Athlon processors. For similarly rated cpus Xeon 2.2GHz and Athlon MP2200, the Xeon's clock speeds are higher, but the floating point performance are about equivalent while the Athlon has a significant advantage on integer operations. Bioinformatics applications rely heavily on integer operations, therefore even with equivalent costs the applications would favour the choice of the Athlon processor. In terms of price/ performance the floating point performance favours the Athlon. Considering all options the nodes were built with dual Athlon MP2100+ CPUs, 1GB memory and dual 3com 64bit gigabit network interface cards. The dual processor configuration was favoured so that the network interconnect costs would be contained. Standard unmanaged gigabit switches with two network interface cards in each node were chosen. The two network interface cards will theoretically increase each nodes available bandwidth, but due to bus contention issues, the main advantage of the two cards are the increased connectivity, namely the number of nodes that are directly connected via a single switch. The nodes are arranged in rows and columns on the same switch and each node will statically choose the best path (either located in the same switch or with the smallest number of hops) to the next node. Typically, if they are within the same switch they will have a low latency and high bandwidth. Given the constraints, i.e., the number of nodes, availability of ports on the switches etc, the final design is shown in figure 1. In this design there is either a direct connection to another node via a single switch, or at most up to 3 switched.

3 Performance

Tests were conducted with just two nodes, network cards and one network switch from each vendor. The selection of the nodes and switches was determined by the performance on the Linpack benchmark for four processors. Two dual AMD 2100 resulted in a 8.45 Gflops,(Rmax)

Table 1: Gigabit Ethernet real bandwith versus PCI bus

PCI bus	Bandwith
64 bit (133 MHz)	920 Mbps
64 bit (100 MHz)	850 Mbps
32 bit (100 MHz)	600 Mbps

while two dual Xeon 2.0 resulted in 8.36 Gflops. The bandwidth and latency characteristics of the network are shown in table 1.

Having selected and assembled the cluster, the testing of the Helix performance on the Linpack benchmark and specific issues regarding the network infrastructure and its effects on overall performance (using mmptest and jumpshot) were investigated.

3.1 Linpack performance results

The Linpack benchmark was run as soon as the Helix's construction was completed. The Linpack benchmark measures the performance of a distributed memory computer while solving a dense linear system in double precision. Several factors determine the performance achieved on this benchmark, but the bandwidth and latency are a significant component. High bandwidth and low latency ensure that the communication is not a bottleneck in the algorithm. The algorithm tends to be more efficient for larger problem sizes and since the optimal performance is when the problem fits into available system memory, the larger the memory of the machine the better. Each of the nodes in the Helix contains 1GB of RAM, so this provides a limit to the maximum problem size. Since we require 8 bytes per double precision floating point number, the maximum theoretical matrix size is 94,118 by 94,118. However since there is operating system overhead, a more realistic size using 80% of the memory gives a figure of 80,000 by 80,000. The theoretical peak performance (Rpeak) of the Helix (the maximum Gflop rating that the machine is guaranteed never to exceed) is 448.8 Gflops (9), however this is not a useful measure since it will not be achieved even if every machine is working independently. The peak Linpack performance (Rmax) using a problem size of 82080 was 234.8 Gflops (10).

The performance of the Helix on the Linpack benchmark vs processors (and memory) shows that the system scales almost linearly (see figure 2), this is due to the high bandwidth, reasonable latency switching and the grid layout of the nodes. The switches each have 24 ports and as seen in figure 1 the vertically connections connect up to 22 nodes on a single switch. This means that up to 44 processors can be used with a single high bandwidth hop between processors. When a larger number of processors are required for the Linpack benchmark, then the nodes must be selected to make optimal use of the grid layout. The almost linear increase in performance versus number of processors for the Helix indicates that the architecture is scalable and so additional nodes can be added to the Helix to increase the size of problems solved as well as the speed at which they will be solved. The grid layout can be expanded to accommodate row and column sizes of 23 nodes which will be equivalent to 1058 processors with an expected Linpack rating of 1.88 Tflops putting it in the 24th position in the November 2002 Top 500 list at an estimate cost of approximately US$ 1 million.

3.2 An analysis of performance using mpptests

There are two main reasons we chose 'mpptests' for the performance assessments. There was a need to assess performance when using standard libraries in a fairly standard way, offering average numbers for the programmers to check their own applications performance. This set of applications allows evaluation of how efficiently the hardware is being used by the whole software structure,

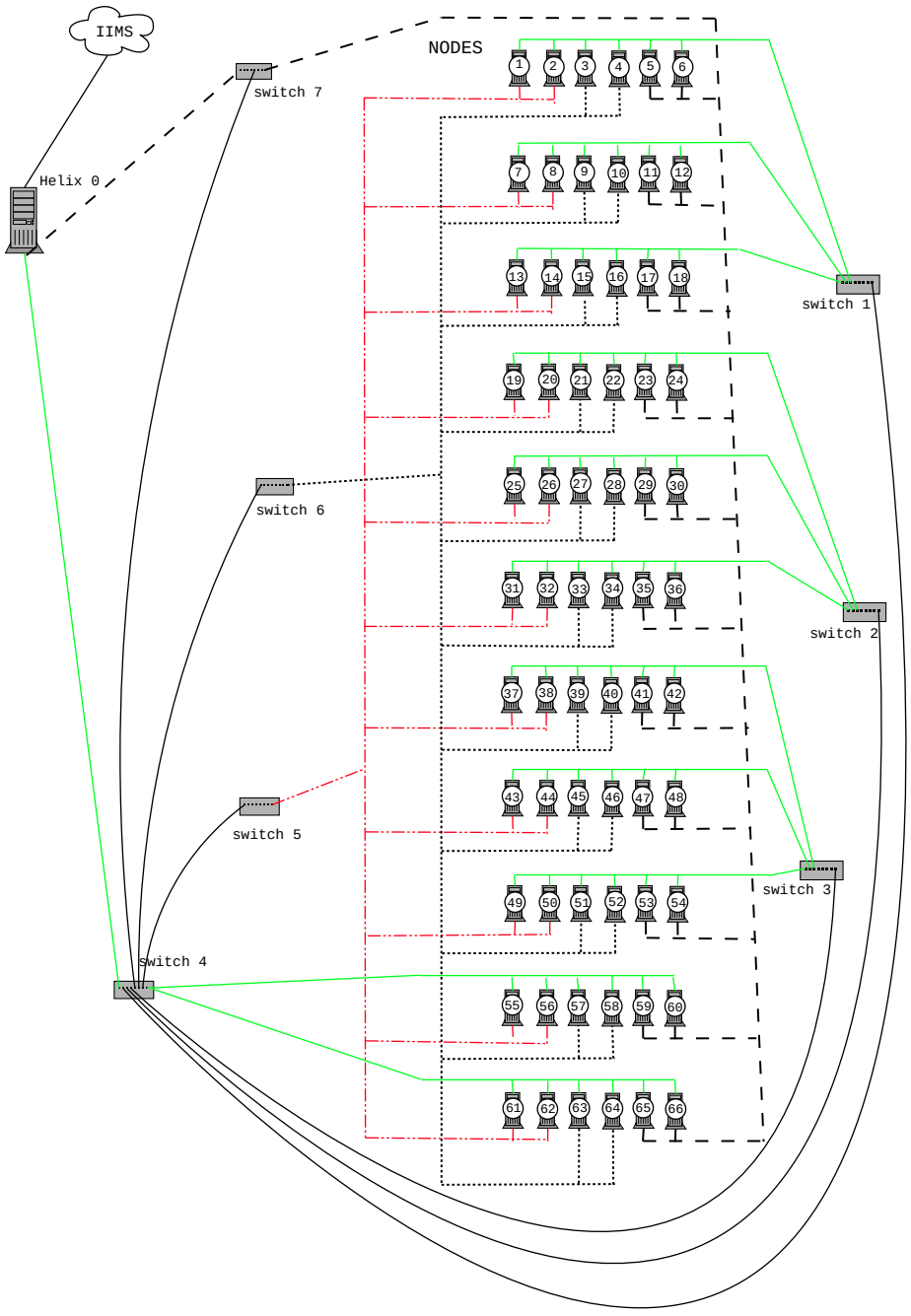

Figure 1: Helix's topology.

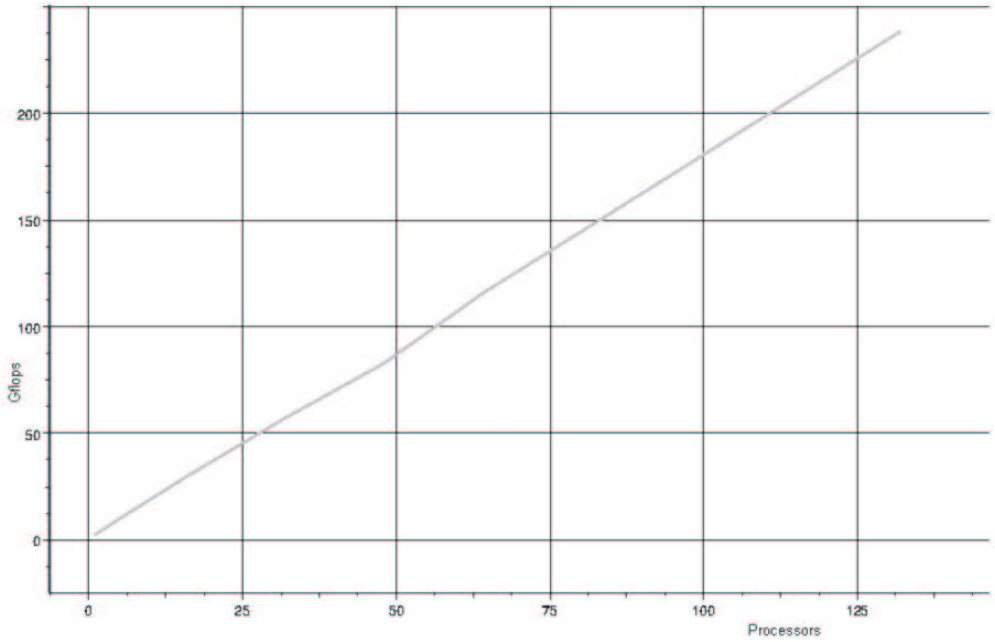

Figure 2: Linpack Rmax rating versus number of processors.

including the operating system and MPI. The second reason for using 'mpptests' is that their results are reproducible despite the simplicity of the tests. There are other benchmarks available that will test a variety of the MPI calls, for example SkaMPI (8). It has been noted before that a single factor can drag performance down significantly. For example, it is well known that certain device drivers do not use the full bandwidth or work poorly with certain Gigabit adapters. For this reason a careful selection of cards and drivers was done.

A point to point test shows that for short messages in a Gigabit Ethernet the times vary very little up to 1000 bytesand their values are around 35 microseconds. This may be considered as an indication of the latency time. This compares favourably with the same test run in the older cluster (the sisters) with fast Ethernet. Due to the low bandwidth as the message size increases the times build up quickly and it goes to 250 microseconds for a message size of 1000 bytes.

It is interesting to note that theoretically the gigabit Ethernet should be 10 times faster but due to latency effects, the speedup is not as high for small messages. The latency of less than 40ms is reasonable for the gigabit Ethernet system. The times for even very short messages are about 75% when compared to the fast Ethernet and there is a gain in speed for any message size. The bigger the messages, the more advantageous becomes the use of Gigabit Ethernet.

With longer messages, the picture changes dramatically. Now the bandwidth on the Gigabit Ethernet shows the full advantage of its use. Figure 3 shows that the times grows linearly up to 65KB in the Helix. The same tests running on the Sisters are shown on figure 4. For 65KB the transfer times are approximately 6.7 times the ones obtained on Helix. The same trends were observed when using non-blocking/point to point messages. The maximum theoretical bandwidth in the fast Ethernet and on a Gigabit Ethernet are 12.5MB/s and 125MB/s respectively. For long messages (figures 3 and 4) Fast Ethernet is approximately 87% as fast as the maximum. A similar analysis for the Gigabit Ethernet shows that it is only about 59% as fast as the theoretical maximum.

The other common tests are the ones with collective operations. Both clusters were measured using "goptest" with -bcast option operations. An interesting effect of the smp machines using

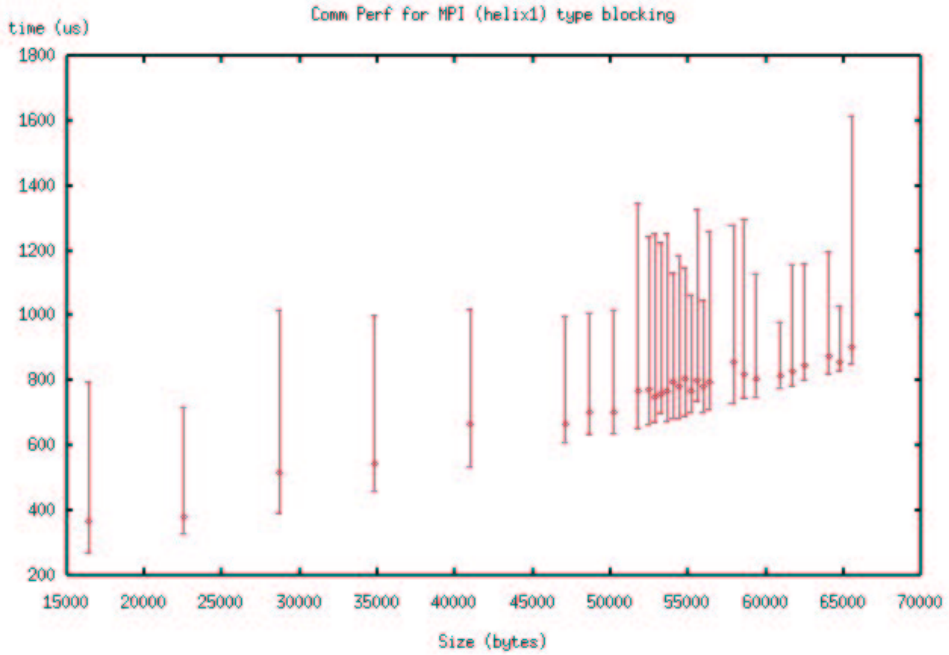

Figure 3: Point to point messages in Helix (long, blocking)

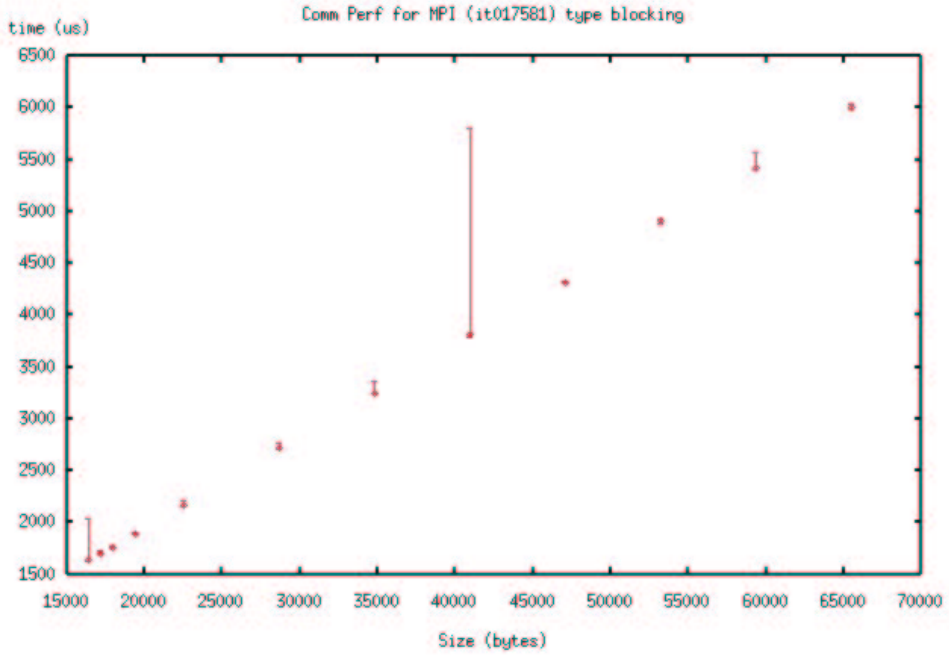

Figure 4: Point to point messages in Sisters (long, blocking)

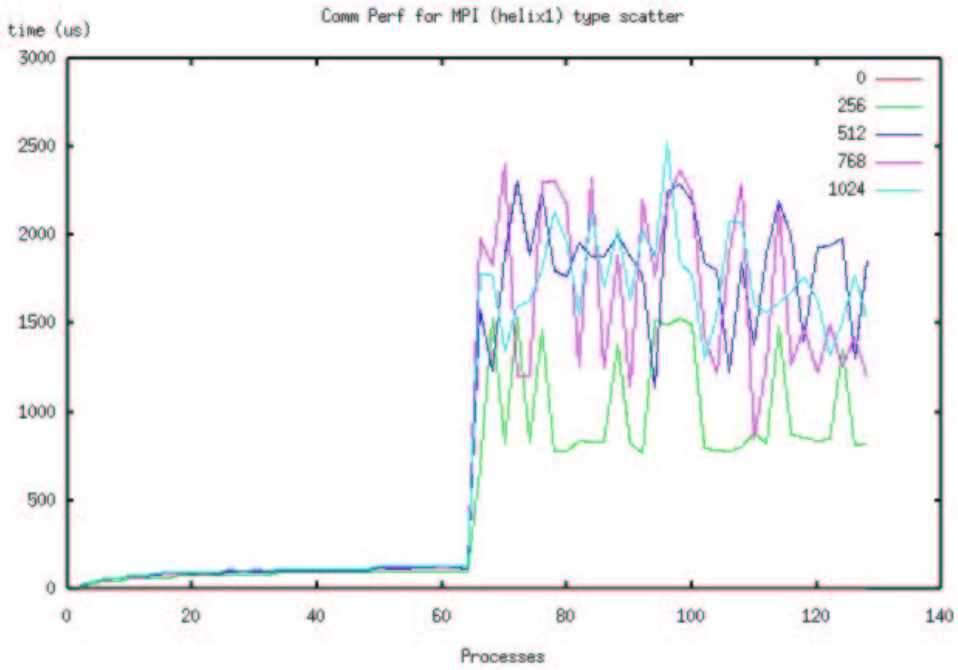

Figure 5: Goptest running in 128 processors with the default order

Allreduce operations can be noticed in figure 5. The nodes were ordered in such a way that only one processor was used up to 64 nodes, then the list restarts from node 1 and went in order up to the 64th node (the order was 1-2-3-4-...64-1-2-3-4...64). When the Allreduce operation was done over more then 64 processors, the time suddenly increases and it seems to be independent of the message sizes. The test was repeated several times to make sure the effect was not due to some specific state of the networking structure. It was noticed that the same test running with the nodes arranged in a different order had a different impact on the times. Many combinations were tried with varying results. Ordering the nodes properly the effect on the Allreduce operations can be different. In figure 6 the order was 1-2-3-4...63-64-64-63-...4-3-2-1. This effect seems to be due to the way that the AllReduce operator is implemented in mpich. Using the new ordering the number of processes that communicate with each other on the same smp node are reduced, eliminating this detrimental effect.

3.3 Ring test using analysis jumpshot

An interesting observation was made by Zaki et. al. (11).They found that the Beowulf network structure can sometimes be revealed by running a ring test application in MPI and observing the CLOG file. Applying this same concept, ring tests were run in the Helix cluster. When two neighbouring nodes are not located in the same switch, it would be expected that a slightly longer message passing time would be measured. For example, there are two hops between nodes 36 and 37 (figure 1). Running a ring test with a 127KB message size on the cluster the following CLOG file was obtained and it is shown in figure 7.

The figure shows that there is no noticeable effect of the topology, even when using relatively long messages. Several tests using the processors in different order were carried out to confirm the results. This figure also shows that the point to point messages are being delivered with the same speed as the single point to point messages in figure 3 with little degradation when making multiple hops over switches. This could be explained by the fact that the additional delay introduced by

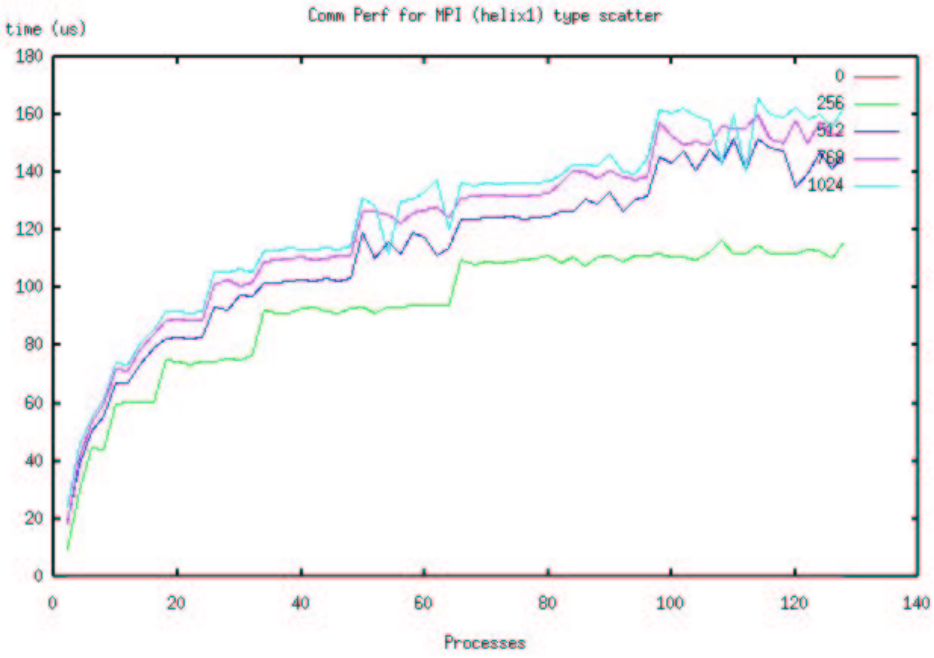

Figure 6: Goptest running in 128 processors using a different node ordering

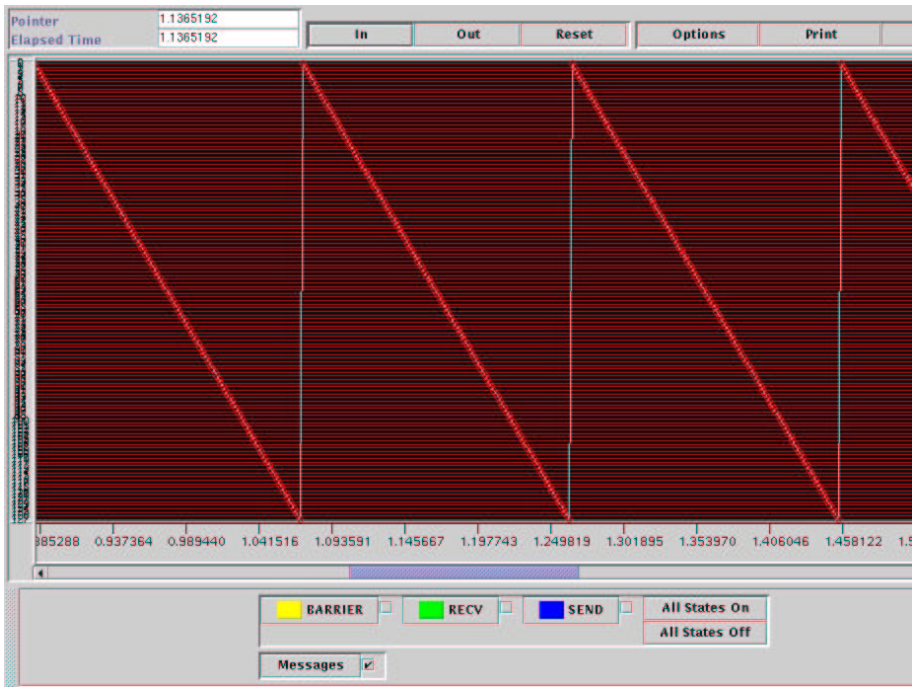

Figure 7: A ring test using 127KB message and 128 processors

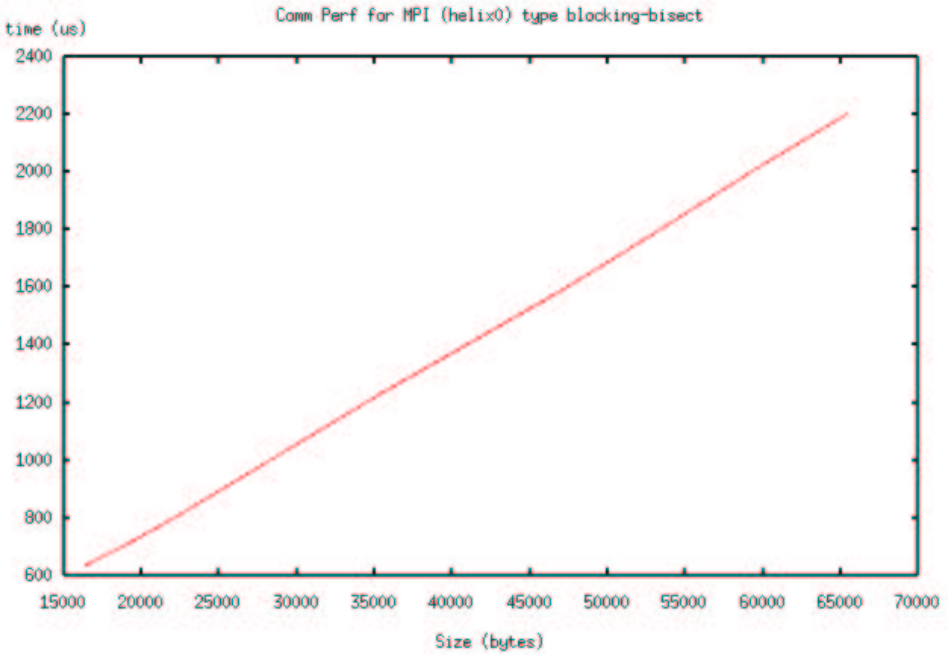

Figure 8: Bisectional bandwidth of the Helix

the switches is small compared to the communication overhead of MPI.

3.4 Bisectional bandwidth

Figure 8 shows that the bisectional bandwidth of the Helix is approx 220Mbps for a broad range of message sizes. These results were obtained using just one process at each node as reliable reproducible results could not be obtained running two processes in each node. These results are consistent with a theoretical analysis of the network topology where we section the network in two halves at the node 32-33 boundary. This means that we have about 110Mbps of full duplex bisectional bandwidth available for each processor which means that the architecture will give reasonable performance for algorithms that have not necessarily been optimised for the Helix. The reasonable bisectional bandwidth is a result of the large number of nodes (almost half of the cluster) that are connected directly to each node by a Gigabit Ethernet connection. As the network size increases to the proposed maximum of 23 x 23 nodes, or 1058 processors, the bisectional bandwidth will not scale well and so the algorithms will have to be modified to suit the grid topology.

4 Conclusions

The use of gigabit Ethernet has proved to be a good choice, even considering that the gain in performance for short messages is not 10 times. Although the relative latency is high, especially with very short messages, there are gains for any message size and the absolute latency is reasonable. The bigger the message the better the gains when compared to fast Ethernet. The effect of using SMP machines with two Gigabit Ethernet adapters each proved to be economical and performs well. Even considering that the two cards cannot achieve their maximum bandwidth when communicating simultaneously due to motherboard constraints it still plays an important role of fairly distributing communication loads across the network. The collective tests showed that by carefully

ordering the way the processors work with certain MPI collective operations a better result can be achieved.

References

[1] Grosz, L. and Barczak, A.: Building an Inexpensive Parallel Computer. Res. Lett. Inf. Math. Sci.**1** (2000) 113–118

[2] Benchweb: www.netlig.org/benchweb/

[3] Gropp, and Lusk, : Reproducible measurements of MPI performace characteristics. In: Dongarra,J., Luque, E., and Margalef, T., (editors): Recent Advances in Parallel Virtual Machine and Message Passing Interface, volume 1697 Lecture Notes in Computer Science, pages 11–18. Springer Verlag (1999).

[4] Wilkinson, B. and Allen, M.: Parallel Programming. Prentice Hall, New Jersey (1999)

[5] Savarese, D. F. and Sterling, T.: Beowulf. In: Buyya, R. (ed.): High Performance Cluster Computing, Vol.1. Prentice Hall PTR, New Jersey (1999)

[6] Ridge, D., Becker, D., Merkey, P., Sterling, T.: Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs Proceedings, IEEE Aerospace, 1997

[7] Plaat, A., Bal, H.E., Hofman, R.F.H., Kielmann, T.: Sensitivity of parallel applications to large differences in bandwidth and latency in two-layer interconnects Future Generation Computer Systems 17 (6): 769-782 APR 2001

[8] Reussner, R.: Recent Advances in SKaMPI In: Krause, E. and Jäger, W. (eds): High Performance Computing in Science and Engineering 2000 Transactions of the High Performance Computing Center Stuttgart (HLRS) 520–530, SPRINGER (2001)

[9] clusters.top500.org

[10] www.top500.org

[11] Zaki,O. and Lusk E. and Gropp,L. and Swider, D: Toward Scalable Performance Visualization with Jumpshot. High Performance Computing Applications vol 13 number 2 (1999) 277–288