

Performance Analysis of HP AlphaServer ES80 vs. SAN-based Clusters

B. Gordon, S. Oral, G. Li, H. Su and A. George
{gordon, oral, li, su, george}@hcs.ufl.edu

High-Performance Computing and Simulation (HCS) Research Lab
Dept. of Electrical and Computer Engineering, University of Florida
Gainesville, Florida 32611

Abstract

The last decade has introduced various affordable computing platforms to the parallel computing community. Distributed shared-memory systems and clusters built with commercial-off-the-shelf (COTS) parts and interconnected with high-performance networks have proven to be serious alternatives to expensive supercomputers in terms of both performance and cost. HP's new AlphaServer ES80 is an example of distributed shared-memory systems, while SCI and Myrinet are the two most widely used high-performance interconnects in building parallel-computing clusters. In this study, we experimentally compare the performance of these parallel computer systems. The emphasis is pointing out the strengths and weakness of the HP's AlphaServer ES80 in comparison with high-performance SCI and Myrinet clusters. We evaluated the systems in terms of sustainable memory bandwidth, interprocess communication and overall parallel computation performance using various widely-accepted benchmarks such as STREAM, PALLAS PMB-MPI, and NAS2.3 parallel suite. It was observed that the HP's AlphaServer ES80, executing Linux, provides remarkable computing power while its communication subsystem cannot handle heavy loads of small messages as effectively.

Keywords: Distributed Shared-Memory, System Area Network, EV7, Scalable Coherent Interface, Myrinet, Benchmarks.

1. Introduction

The need for powerful computing has led to the development of parallel computing. Not long ago, machines designed for substantial computing were limited to small numbers of institutions which could afford supercomputers. As technology prices dropped,

new alternatives to expensive supercomputers emerged. Among the methods explored, two architectures of parallel computing which provide substantial computing power are Distributed Shared-Memory (DSM) machines, and clusters with System Area Networks (SANs). These systems provide reliable, powerful computing and allow for some degree of scalability. However, the communication between processes can be a bottleneck for both kinds of systems [1].

SAN clusters consist of PCs connected together by a high-performance network [2]. This arrangement allows inexpensive COTS nodes to be combined to provide substantial computing power at a lower cost than traditional supercomputers. Each computing node has its own processor, memory, and storage. These nodes usually communicate through the high-performance network by various means of message passing. The networks are optimized to allow for good throughput with low latency for efficient passing of varying message sizes. As more computing power is needed, more PCs may be added. Scalable Coherent Interface (SCI) and Myrinet are two types of SAN technologies commonly used for clusters [3-4].

DSM computers are based on a number of processors controlling their own memory space, but also allowing other processors to read and write to their memory. Cache coherency schemes are used to maintain synchronized distributed information. Thus multiple memory banks are physically distributed in the system, but through cache coherency the memory appears to be uniformly shared. This architecture allows for a degree of scalability but issues of bandwidth and latency become more important as processors are added. The new AlphaServer HP ES80 (codenamed "Marvel"), based on the new EV7 processor, is a type of DSM computer [5].

This paper seeks to show the strengths and weaknesses associated primarily with the DSM architecture of the Marvel system. By analyzing the performance of the communication and computational aspects of the system and comparing the results with the

known architectures of SCI- and Myrinet-based clusters, these strengths and weaknesses are explored.

The organization of the paper is as follows. The next section will offer an architectural overview of the systems involved in this study. Section 3 offers an overview of the test-beds used in the study, and the selected benchmarks. The experimental results and analysis for each benchmark are presented in subsections of Section 3. Finally, Section 4 presents the conclusions and possible directions for further study.

2. Overview of Test-bed Architectures

The Scalable Coherent Interface (SCI) is an ANSI/ISO/IEEE standard (1596-1992) that describes a packet-based protocol [6]. The SCI implementation used in this study is Dolphin/Scali Wulfskit, which is a commercial high-performance implementation of the SCI standard. Dolphin/Scali SCI provides high-performance cluster computing by interconnecting clusters of computing nodes. SCI uses point-to-point links, maintaining low latency while achieving high data rates between nodes in a cluster that is scalable up to 64k processors. SCI meets the demands for high-performance cluster computing by allowing for distributed shared-memory computing and providing a message-passing interface [7].

Myricom's Myrinet is an ANSI standard (26-1998) that also describes a packet-based protocol [4]. Like SCI, Myrinet is also used to provide high-performance cluster computing. Nodes are connected using full-duplex, point-to-point links that offer low latency. Myrinet hosts usually have only one interface port. Unlike SCI, Myrinet hosts are interconnected using Myrinet switches that have multiple ports and switch packets depending on the routes defined in the packet headers. Switch-based networks are scalable up to tens of thousands of nodes, and the topology can contain multiple-path redundancy. Also setting itself apart from SCI, Myrinet only provides for message passing, and not shared memory [8].

The Marvel system is built around the new Compaq Alpha EV7 processor. The EV7 incorporates an EV68 core, two levels of cache, memory controllers, an Input/Output (I/O) port, and four interprocessor ports all within a single chip [5]. By having the memory controllers and I/O and inter-processor ports on chip, the EV7 is easily scalable while allowing individual processors quick and easy access to resources. There are four interprocessor ports on each processor labeled North, South, East and West (see Figure 1). All communication in or out of the chip, besides memory, is done through a single router. Memory integrity is maintained by directory-based cache coherency [5].

This approach allows for fewer connections and easier maintenance of chip resources. The basic building block of these servers is two EV7s connected by their North and South ports.

The three architectures evaluated represent points in a spectrum between purely distributed memory systems and purely message passing systems. Marvel is a DSM, the Myrinet cluster is a message-passing system, while SCI is a hybrid between the two.

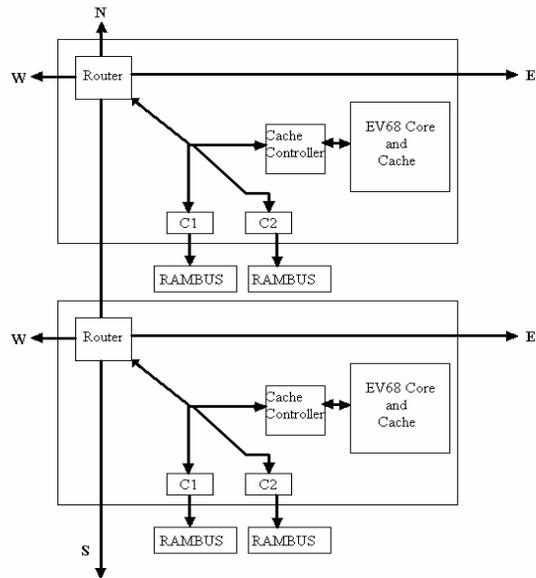


Figure 1: Basic block diagram of EV7-based systems

It is important to understand the various strengths and weaknesses that each of the above architectures have in order to know what type of architecture is best suited for the types of applications that are executed on parallel computer systems. To discover these characteristics, it is necessary that performance analysis is performed on various aspects of the architectures.

3. Experimental Performance Analysis

Three components involved in the performance of parallel computing architectures are the speed and bandwidth of the memory involved in each system, the interprocessor communication latency and throughput, and the overall computational power of each processor in the system. This study chooses the STREAM memory bandwidth, PALLAS PMB-MPI1, and the NAS NPB2.3 parallel benchmark suites to measure the performance of each component listed above as well as the combination of all three.

The SCI and Myrinet clusters involved in this study consist of 12 nodes, with the SCI cluster connected as a 4x3 torus network and the Myrinet cluster as a simple

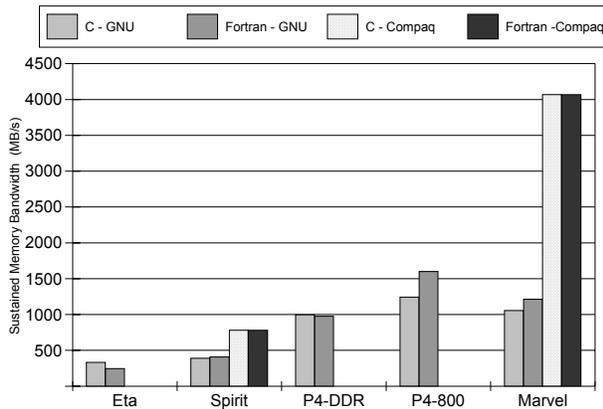
star topology with a switch in the center. The *Eta* computers are the hosts for both SCI and Myrinet. Each host is made up of dual 1GHz Intel Pentium-III processors with integrated 256KB L2 cache, ServerSet III LE chipset and 133MHz system bus, 256MB PC133 SDRAM, and two 64-bit, 66MHz PCI slots. The SCI interconnect cards are PCI-64/66/D330 operating at 5.33 Gb/s and providing one-way latencies as low as 2 μ s, and the Myrinet interconnect cards are M2M-PCI64A-2 operating at 1.28 Gb/s with one-way latencies as low as 10 μ s.

The Marvel system involved in this study consists of four EV7 chips which are housed by pairs in two boxes with proprietary interconnects between the two boxes. This arrangement creates a connection of four processors in a single ring where each EV7's South port is connected to another EV7's North port. Per processor there is 2GB of RAMBUS 800MHz memory, 128KB L1, and 1.5 MB L2 caches.

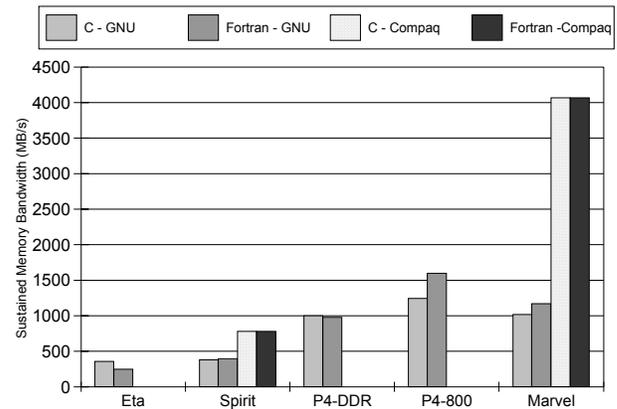
Each node of the SCI and Myrinet clusters uses Red Hat Linux 7.2 with mtrr-patched kernel 2.4.7-10smp. Marvel uses the generic Red Hat version available from Compaq with kernel 2.4.9-32.6jw2numasmp. This kernel was compiled for an EV4, so many of the newer features of the EV7 are not implemented. Marvel uses MPICH 1.2.4.6 compiled with the Compaq C compiler V6.4 to handle the parallel code. The SCI nodes use SSP 3.0.1 ScaMPI and the Myrinet nodes use MPICH-GM 1.2.4.8 to run the parallel processes.

3.1. STREAM benchmark

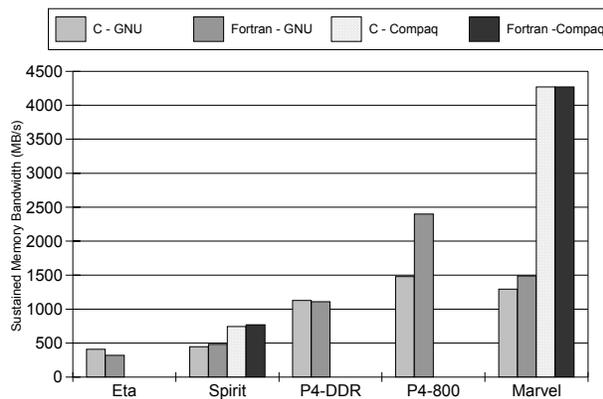
STREAM is a simple synthetic benchmark program that measures sustainable memory bandwidth in MB/s and the corresponding computation rate for simple vector kernels. The measurement is done through four separate functions on arrays. The first function, *Copy*,



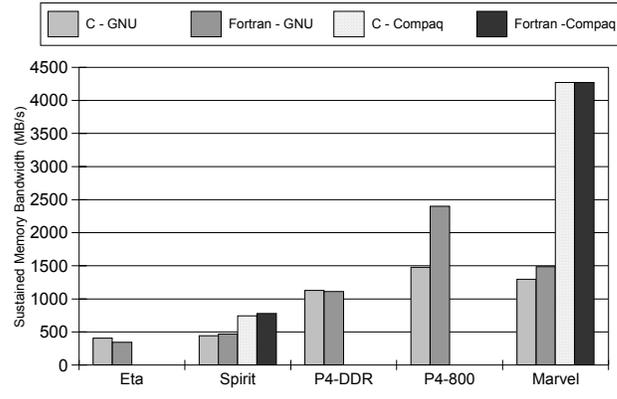
(a) Copy



(b) Scale



(c) Add



(d) Triad

Figure 2: STREAM results for Copy (a), Scale (b), Add (c), and Triad (d) on single processors

moves an array from one part of memory to another, $A[i]=B[i]$. The second function, *Scale*, scales an array by a constant and moves it to another array, $A[i]=q \times B[i]$. The third function, *Add*, adds two arrays together and places the result in a third array, $A[i]=B[i]+C[i]$. The fourth function, *Triad*, multiplies one array by a constant, adds it to a second array and places the result into a third array, $A[i]=q \times B[i]+C[i]$.

Separate timing is done for each function and the results are tabulated with the MB/s, minimum time, maximum time, and average time for each function [9]. All arrays reside in contiguous blocks in memory on single nodes, and should be at least four times the size of the largest cache on the system.

STREAM was executed on several different systems to show comparisons between the memory bandwidth of different memory architectures. The *Eta* computers are the nodes on which SCI and Myrinet are installed. In addition to these hosts and Marvel, several other host machines were included in the STREAM experiments for comparison. *Spirit* is a Compaq XP1000 workstation with a 667MHz EV67 processor, 1.2 GB 100MHz ECC SDRAM, 128KB L1 and 4MB L2 caches. Spirit uses Red Hat Linux 7.2 with kernel 2.4.9-32jw3. The *P4-DDR* is a Dell Dimension 4500 with a 2.0GHz Pentium-4, 256MB DDR-2100 SDRAM, 64KB L1 and 512KB L2 caches. The P4-800 has a 1.4GHz Pentium-4 with 512MB RAMBUS 800 RAM, 64KB L1 and 512KB L2 caches. Both Pentium-4 computers execute Red Hat Linux 7.3 with kernel 2.4.18-3.

The C and FORTRAN compilers used to compile STREAM are the GNU gcc V2.96 and f77 V0.5.26 compilers. In addition, Compaq C V6.4 and Compaq FORTRAN V1.2 were used to compile STREAM on Marvel and Spirit. The compile flag of `-O3` was used as optimization in the compilation of all benchmarks.

Because STREAM measures the amount of memory being accessed per second, we would expect to see that Add and Triad, which perform 3 memory operations (2 read, 1 write) would have a higher bandwidth usage than Copy and Scale, which perform 2 memory operations (1 read, 1 write). The graphs in Figure 2 show this trend. By comparing the performance for P4-800, P4-DDR and Eta, which have similar memory architectures, it is seen that the performance increases with bus speed. With the CPU running at a much faster clock rate than the memory bus, the additional CPU operations have little effect on the STREAM result. Since Spirit uses a different memory architecture than that of Eta and the P4 systems, no direct comparison can be made. However, we can see that Spirit with memory running at 100 MHz is actually running faster than the Eta with memory running at 133 MHz, which suggests that the memory architecture in Spirit is more efficient than that of PC-based memory systems.

One might expect to see a comparable performance between the Marvel and P4-800 systems because both feature the same RAMBUS memory technology and 800MHz memory bus speed. However, there are many other architectural factors that impact the level of performance for sustained memory bandwidth. The performance of Marvel is only slightly lower than that of P4-800 when STREAM was compiled with the GNU compilers. However, when using the optimized compilers from Compaq, the Marvel results achieve more than double the performance of the P4-800. This discrepancy exists because the GNU compiler is optimized for PC-based systems and not for the new Marvel architecture. This unoptimized compiler is the cause of lower memory bandwidth observed in the GNU compiler case since it is unable to take full advantage of the memory architecture in the Marvel machine. When a corresponding, optimized compiler is used, the advantages provided by the new memory architecture are realized.

In summary, from the STREAM benchmark, the Marvel memory system shows a dramatic increase in performance, when coupled with the correctly optimized compilers versus the memory systems of other computer architectures. Using correctly optimized compilers, the memory system provides high throughput and presents less of a bottleneck in Marvel as compared to other previous Alpha- and PC-based systems.

3.2. Pallas Exchange benchmark

The Pallas MPI benchmarks suite is a set of benchmarks developed by Pallas GmbH that measures the most important MPI functions [10]. Among these benchmarks is *Exchange*, named for the communications pattern that it uses, which measures sustainable bandwidth in MB/s between processes. The Exchange communications pattern occurs often in grid-splitting algorithms. A periodic chain of processes ($\dots, P_{n-1}, P_n, P_{n+1}, \dots$) exists where each process exchanges data with both its left and right neighbor in the chain. The Exchange pattern for each process n consists of five steps: MPI_Isend sends a message to process $n+1$, MPI_Isend sends a message to process $n-1$, MPI_Recv receives a message from process $n-1$, MPI_Recv receives a message from process $n+1$, MPI_Waitall waits until all the other processes have synchronized [10].

Exchange is tested with message sizes ranging from 1B to 4MB. All interprocessor message passing is timed and the results are given in a table with average throughput achieved in MB/s versus the message size. The results are plotted in Figure 3.

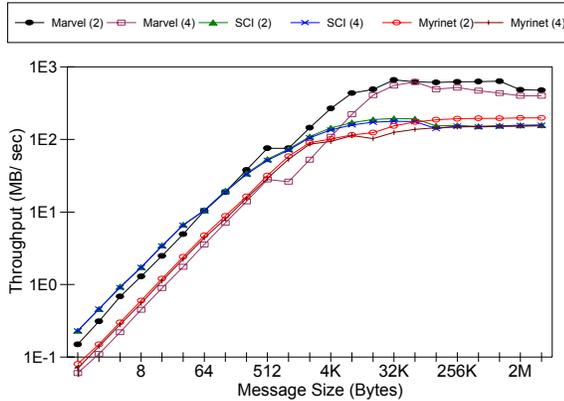


Figure 3: Throughput results from the Exchange benchmark, plotted on a logarithmic scale

Each of the Exchange processes ran on a separate processor, so the interprocessor communication power of several systems could be observed. The SCI and Myrinet tests were executed on Eta computers with one process per computer. Exchange was compiled with gcc V2.96 on Eta and with Compaq C V6.4 on Marvel; the `-O3` compilation flag was used for all builds for optimization. All systems were tested with two and four Exchange processes. For Marvel, the 2-process tests are done with both processes existing on the same 2-processor box, while the 4-process tests are done using two 2-processor boxes requiring inter-box communication.

As expected, all systems showed a general increase in bandwidth as message sizes increase. For small message sizes (128 bytes and below), SCI displayed the highest throughput followed by 2-process Marvel, Myrinet, and 4-process Marvel. The 2-process Marvel surpasses SCI and Myrinet in throughput when the message size reaches above 512 bytes. By comparison, the 4-process Marvel surpasses SCI and Myrinet when the message size reaches 8192 bytes. So, Marvel is observed to have a clear advantage over SCI and Myrinet for large message sizes.

Although having smaller throughput than Marvel, both SCI and Myrinet have 2-process and 4-process results that are relatively similar (when compared to Marvel). For SCI, the average percentage difference in throughput between 2-process and 4-process does not exceed 10%. For small message sizes (1024 bytes and below), the average percentage difference does not exceed 3.5%. Myrinet also displays small differences in results between 2-process and 4-process throughput for messages up to 16KB.

By contrast, Marvel shows large differences between 2-process and 4-process throughput results at nearly all message sizes. For message sizes of and below 4096 bytes, 2-process throughput results are more

than twice the throughput achieved by the 4-process setup. Only after messages sizes larger than 8KB does the throughput of 2-process and 4-process results from Marvel begin to converge. Still, the 2-process Marvel results show slightly better throughput than the 4-process Marvel.

Latency results for the systems were very similar to the throughput performance. For small message sizes (128 bytes and below), SCI displayed the lowest latency followed by 2-process Marvel, Myrinet, and 4-process Marvel. The results of the benchmark running on 2 processors in Marvel show better latency times than SCI as the message size reaches beyond 256 bytes. The results from 4 processors on Marvel running Exchange shows better latency times than SCI and Myrinet as the message size exceeds 8192 bytes. Figure 4 summarizes the latency results for the Exchange benchmark

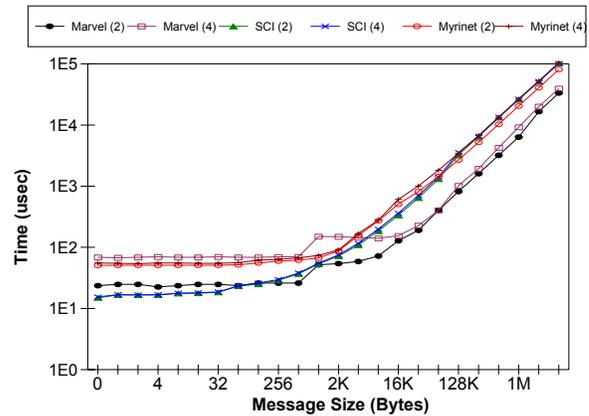


Figure 4: Latency results from the Exchange benchmark, plotted on a logarithmic scale

Just as in the throughput analysis, the same drop-off in performance for 4-process versus 2-process in Marvel could be seen in the latency results. For SCI, the difference between 2-process and 4-process latency never exceeds 9%. For Myrinet, the difference never exceeds 9% for message sizes up to 8192 bytes. In the Marvel system, the latency for the 4-process setup is more than twice that of the latency of the 2-process setup for message sizes up to 4096 bytes. As in the throughput results, as message sizes increase past 8192 bytes, the 2-process latencies and 4-process latencies for Marvel begin to converge.

Marvel's 2-process communication outperforms its 4-process communication in both throughput and latency tests. Although when message sizes increase past 8192 bytes the performance of the two converges, this performance difference is significant for small message sizes at and below 4096 bytes. Thus, clearly a penalty is paid for inter-box communication in the Marvel system running Linux.

3.3. NAS 2.3 parallel benchmarks

The NAS NPB 2.3 Parallel Benchmarks (NPB) are used to test both computational power and process communication through simulated and real application benchmarks [11]. The suite contains eight separate benchmarks. The benchmarks fit into five categories: high computation with only large-sized communications done at the beginning and end; high computation with a few large-sized communications; high computation with many small-sized communications; low computation with random communication; and low computation with many small-sized communications.

Five of the eight benchmarks found in the NPB2.3 suite were chosen to show strengths and weaknesses of the SCI and Myrinet versus Marvel because they illustrate each of the categories mentioned above. All the benchmarks are available in three sizes, class-A, class-B, and class-C. Class-A is the smallest set while class-C is the largest. All benchmarks were executed with class-A-sized problems. For all benchmarks, one process was executed per node for SCI and Myrinet, and one process per processor on Marvel. Thus, processor to processor communication could be tested.

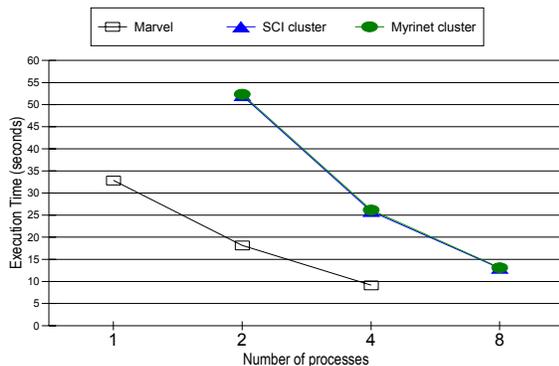


Figure 5: Results of NPB EP benchmark

The EP benchmark generates pairs of Gaussian random deviates [11]. This benchmark tests the pure combined computational power of each system. Separate sections of the uniform pseudorandom numbers can be independently computed on separate processors. The only requirement for communication is the initial distribution of the processes and the combination of the 10 sums from various processors at the end. Figure 5 shows that Marvel's computational power is far greater than that provided by the SCI and Myrinet nodes. Because of the low communication involved, Marvel's processors are well utilized as the speedup in execution time is almost linear as more processors are added.

The CG benchmark uses the inverse power method to find an estimate of the largest eigenvalue of a

symmetric positive definite sparse matrix with a random pattern of nonzeros [11]. With computation spread out by the large number of zero values and randomly placed non-zero values, CG is a good measurement of random communications between processes. The random communication will vary in size and number of messages passed between processes. The results from CG are shown in Figure 6.

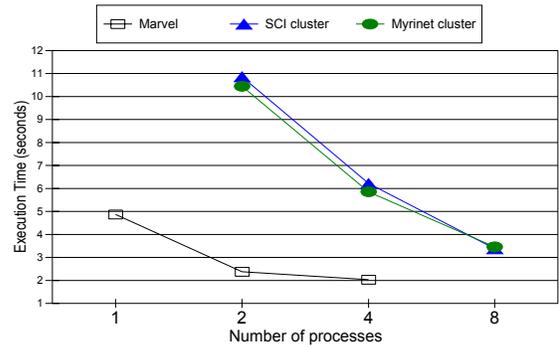


Figure 6: Results of NPB CG benchmark

The CG benchmark shows that while a single EV7 processor and two processors in a box do very well at computing and communicating, random communication between boxes is not as good as communication within one box. There is very little difference between the execution times of 2-processor and 4-processor setups. Meanwhile, SCI and Myrinet handle the random communication patterns well as the number of processes increases. Both systems experience a slight leveling in speedup as processes are added, but still retain a relatively linear speedup. Both SAN clusters show that eight processors are needed to equal the computational power of two EV7s for computationally intensive problems with random communication patterns. It is also seen that for such applications there is no large gain by adding more than the two EV7 processors.

The FT benchmark solves a partial differential equation (PDE) using forward and reverse 3-D Fast Fourier Transforms (FFTs) [10]. FT tests both CPU and message passing, as there are a number of array operations which require that each process share its portion of the array with the other processes. The implementation of the 3-D FFT PDE benchmark follows a fairly standard scheme. The 3-D array of data is distributed according to z planes of the array with one or more planes stored in each processor. The forward 3-D FFT is then performed as multiple 1-D FFTs in each dimension, first in the x and y dimensions, which can be done entirely within a single processor, with no interprocessor communication. An array transposition is then performed, which amounts to an all-to-all exchange, wherein each processor must send parts of its

data to every other processor. The final set of 1-D FFTs is then performed. A conventional Stockham-transpose-Stockham scheme is used for the 1-D complex FFTs. This procedure is reversed for inverse 3-D FFTs. The communication between processes is medium grained with larger message sizes for smaller number of processes. Figure 7 shows the results of this benchmark.

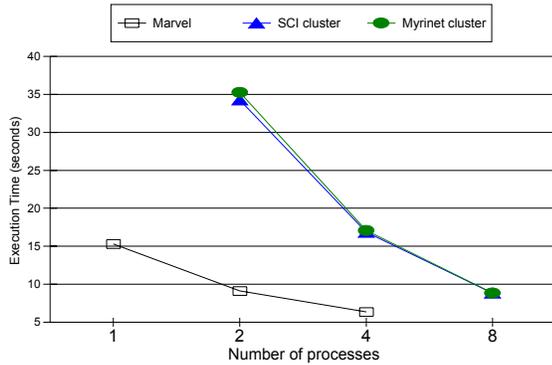


Figure 7: Results of NPB FT benchmark

FT shows that, again, EV7's computational power is substantial and, as long as message sizes are kept large, speedup is good as the number of processors increases. SCI and Myrinet show that they handle large message sizes well, as the execution time is almost halved when the number of processors is doubled.

LU is a simulated computational fluid dynamics application which uses symmetric successive over-relaxation (SSOR) to solve a block lower triangular-block upper triangular system of equations resulting from an unfactored implicit finite-difference discretization of the Navier-Stokes equations in three dimensions [12]. A 2-D partitioning of the grid onto processors occurs by halving the grid repeatedly in the first two dimensions, alternately x and then y , until all power-of-two processors are assigned, resulting in vertical pencil-like grid partitions on the individual processors. The ordering of point-based operations constituting the SSOR procedure proceeds on diagonals which progressively sweep from one corner on a given z plane to the opposite corner of the same z plane, thereupon proceeding to the next z plane. Communication of partition boundary data occurs after completion of computation on all diagonals that contact an adjacent partition. It results in a relatively large number of small- to medium-sized communications. The results of the LU benchmark are plotted in Figure 8.

The LU benchmark shows that the SAN systems handle well the scaling of computationally intensive processes with many small- to medium-sized communications. The decrease in execution time is consistent as processors are added. The Marvel shows that the speedup from 1 to 2 processors is about 3.5 due

to an additional tightly coupled processor adding to the processing capacity with a minimal communication overhead. Here, superlinear speedup is attributed to the increased cache capacity with two CPUs that reduces the frequency of memory access. By comparison, the speedup from 2 to 4 processors is only about 1.75. The processor interconnect becomes a bottleneck as processors are added past the initial 2-processor base.

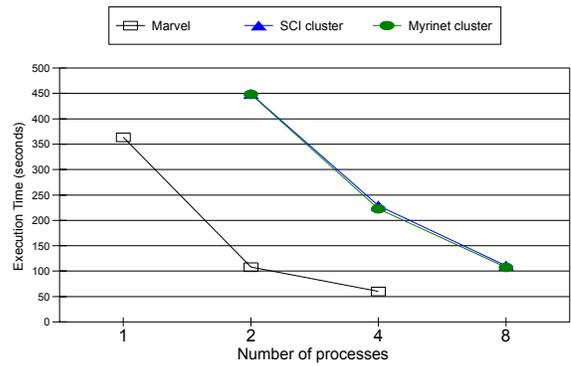


Figure 8: Results of NPB LU benchmark

The final benchmark considered is the IS benchmark. This benchmark sorts N keys in parallel [11]. The keys are generated sequentially and initially must be uniformly distributed in memory. In a distributed memory system with p distinct memory units, each memory unit initially must store Q keys in a contiguous address space, where $Q = N/p$. For our test, each processor places Q keys in the memory space associated with that processor. The benchmark requires little computational power with lots of small communications of a word a piece. The results for the IS benchmark are shown in Figure 9.

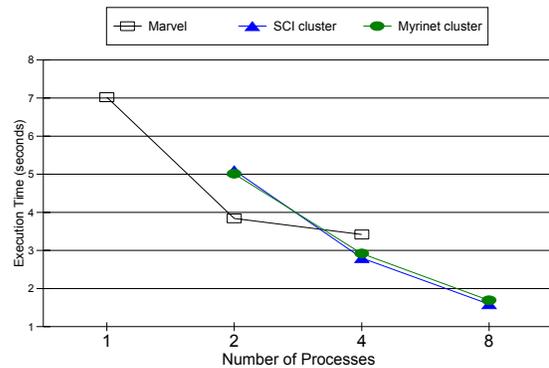


Figure 9: Results of NPB IS benchmark

Here, the communication bottleneck between two boxes plays a major role in the execution time of the benchmark. While execution time is almost halved by moving from 1 to 2 processors, there is little change in execution time by moving to 4 processors. SCI and

Myrinet pass small messages very well. Execution times almost halve as processors are doubled. The figure shows that 4 processors connected either by SCI or Myrinet actually execute the IS benchmark faster than 4 interconnected EV7s. SCI and Myrinet excel at passing many small messages over their respective interconnects. While such communications between two EV7 processors have shown to be comparable to SCI and Myrinet, these communications are less efficient when communication is done between boxes.

4. Conclusions

With the increasing need for powerful computing, many different ways of satisfying that need are emerging. Distributed Shared Memory (DSM) systems and clusters built with System Area Networks (SANs) are two approaches to providing reliable, scalable, and powerful computing. This study compared an EV7-based DSM system to two SAN-based architectures to analyze the strengths and weaknesses of the EV7 system. This analysis was accomplished by looking at the performance of the memory, interprocessor communication, and processor through established benchmarks which target those specific areas.

STREAM tests the memory bandwidth capability of systems. These tests showed that the Marvel memory system has high bandwidth when used with compilers targeted to its specific architecture.

The Pallas Exchange benchmark creates and measures interprocessor communications often seen in grid-splitting algorithms. The results of Exchange showed that there is significant overhead associated with passing message sizes of less than 4KB and that it is especially problematic for communication between 4 EV7s not residing in the same box.

The NAS benchmarks showed that the EV7's computing power is very good, but for communication-intensive applications, scalability was not as good as SAN systems using SCI and Myrinet. The Marvel system under Linux has shown that it excels in computational work that has a small number of communications of large message sizes. However, as message size decreases the overhead involved in communicating between processors often overshadows the message being passed. This overhead can be seen most clearly in the latency measurements as message size increases.

Through this study it has been shown that the Marvel EV7 communication architecture performs near the levels for high-performance SAN clusters. Communication overhead plays a large role in determining the latency and throughput for messages less than 8KB. This overhead is more pronounced for communication between two connected boxes of the

Marvel system than communication between processors located with the same box. The EV7 system architecture has the potential to provide higher throughput with lower latencies than SCI or Myrinet, but this potential is only realized for messages larger than 8KB.

Future research directions may include a similar look at the EV7 performance for systems using Tru64. Also, because the standard Red Hat Linux available from Compaq is compiled for an EV4 architecture, future studies may involve testing performance for Linux targeted for the EV7 architecture.

5. Acknowledgments

This work was supported in part by equipment donated or loaned by HP, Dolphin Interconnect Solutions Inc., and Scali Computer AS. Special thanks go to Harry Heinisch and Jeff Wiedemeier at HP for their technical support with the Marvel machine.

References

- [1] M. Oguchi, H. Aida, T. Saito, "A Proposal for a DSM Architecture Suitable for a Widely Distributed Environment and Its Evaluation." *Proc. of the 4th IEEE Int. Symp. on High Performance Distributed Computing*, August 1995.
- [2] G. F. Pfister, "In Search of Clusters," 2nd Edition, ISBN 0-13-899-709-8, Prentice Hall PTR, NJ, 1998.
- [3] D. Gustavson and Q. Li, "The Scalable Coherent Interface (SCI)," *IEEE Communications*, Vol. 34, No. 8, August 1996, pp. 52-63.
- [4] Myrinet-on-VME Protocol Specification, ANSI/VITA 26-1998, 1998.
- [5] K. Krewell, "Alpha EV7 Processor: A High-Performance Tradition Continues," White paper, http://www.compaq.com/hps/download/Compaq_EV7_Wp.pdf, April 2002.
- [6] IEEE, SCI: Scalable Coherent Interface, IEEE Approved Standard 1596-1992, 1993.
- [7] Scali Computer AS, Scali System Guide Version 2.1, White paper, Scali Computer AS, 2000.
- [8] N. Boden, D. Cohen, R. Felderman, A. Kulawik, C. Seitz, J. Seizovic, W. Su, "Myrinet: A Gigabit-per-Second Local Area Network," *IEEE Micro*, Vol. 15, No. 1, 1995, pp.26-36.
- [9] J. McCalpin, "Sustainable Memory Bandwidth in Current High Performance Computers," White paper, <http://home.austin.rr.com/mccalpin/papers/bandwidth/bandwidth.html>, 1995.
- [10] Pallas GmbH, Pallas MPI Benchmarks - PMB Part MPI-1, Pallas GmbH, March 2000.
- [11] D. Bailey, E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, S. Fineberg, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon, V. Venkatakrishnan, S. Weeratunga, "The NAS Parallel Benchmarks," *RNR Technical Report RNR-94-007*, <http://www.nas.nasa.gov/Software/NPB/Specs/RNR-94-007/node12.html>, March 1994.
- [12] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, M. Yarrow, "The NAS Parallel Benchmarks 2.0," *RNR Technical Report NAS-95-020*, http://www.nas.nasa.gov/Software/NPB/Specs/npb2_0/cember, 1995.