Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

New Experiments Towards to Performance Comparison Between CISC and RISC Architecture Cluster Environments

Hsun-Chang Chang* Li-Jen Chang* Yaw-Ling Lin* Kuan-Ching Li* Chao-Tung Yang‡ Liang-Teh Lee†

*Dept. of Computer Science and Information Management
Providence University, Taichung 43301, Taiwan
Email: {hcchang,ljchang,yawlin,kuancli}@cs.pu.edu.tw
‡High Performance Computing Laboratory, Dept. of Computer Science and Information Engineering
Tunghai University, Taichung 40744, Taiwan
E-mail: ctyang@mail.thu.edu.tw
†Dept. of Computer Science and Engineering, Tatung University, Taipei 10451, Taiwan
E-mail: ltlee@cse.ttu.edu.tw

***Abstract*— This research paper discusses performance issues of RISC and CISC architecture based cluster systems and one grid computing system built based on former two cluster systems. A number of benchmark programs were executed in these computing systems, in order to proceed further analysis of experimental results of same benchmark programs and applications. We could observe that, depending on the nature of the given problem, best results can be shown for a given class of architecture cluster computing platform. As matter of comparison of two cluster systems, analysis from the experimental results indicates that x86 based architecture machines can be the inexpensive workstation clusters, but Apple's PowerMac G5 workstation based cluster systems is potentially recommended for a high performance scientific computations. Additionally, we also shown that grid technology is viable by increasing total system performance at no additional cost for its implementation.**

Keywords- **Network of Workstations, Performance Evaluation, CPU Architecture, Grid computing.**

I. INTRODUCTION

To answer the need to access computing power and to solve large scale computational problems, distributed network computing environment has become a cost-effective and popular choice to attend such high demand computations [4]. Unlike supercomputers of yesterday, a cluster or grid system can be used as multi-purpose platform, to run diverse high-performance applications. As result, growing interest in the scientific community to move numerical computations and simulations from traditional large scale mainframes and supercomputers to distributed parallel computing on low-cost PC cluster platforms.

Nowadays, most cluster systems are built using Intel or AMD CISC architecture computers running a variant of open source operating systems. This cluster environment consists of workstations which are interconnected among themselves using high speed local area network, e.g., Gigabit Ethernet, SCI, Myrinet and Infiniband. Continuous improvements and investigations has been proceeded recently in the performance of networks, both in reducing latency and in increasing bandwidth [13]. In this 21st century, a new technology called Grid is developed, by enabling the coupled and coordinated use of geographically distributed resources for purposes such as large-scale computation and distributed data analysis [2], [8]. Grid technology is not a new idea. The concept of using multiple distributed resources to cooperatively work on a single application has been around for several decades. Together with the development of Globus [7] and MPICH-G2 [6], [11], re-structuring and executing parallel applications in grid computing environments is not another challenge, since it is automatic the process of executing those parallel applications already developed for cluster technology.

The "Message Passing Interface" (MPI) has become a dominant industry standard for writing portable message passing parallel programs. It feature a range of functionality, including point-to-point, with synchronous and asynchronous communication modes, and collective communication. This concept applies to such computer systems capable of exchanging information and communicating through a "Message Passing Interface" (MPI) standard [14].

MPICH-G2 is a grid-enabled implementation of the MPI v1.1 standard, it uses Globus Toolkit [7] services to support efficient and transparent execution in heterogeneous Grid computing environments. In addition, it allows you to couple multiple machines, potentially of different architectures and topologies, to execute MPI parallel applications. It automatically converts data in messages sent among computing nodes of different architectures. Furthermore, MPICH-G2 can be used to execute your application using vendor-supplied implementations of MPI for intra-machine communication and TCP for inter-machine communication.

At Virginia Tech, researchers developed a distributed parallel computing system based on the MacOS X operating system, built by using 1,100 Apple's PowerMac G5 computers. It is the "Big Apple" cluster. In November 2003, it quickly was

ranked as third fastest in Top500 list and it achieved a total performance over 10 Teraflops (peak speed of 17.6 Teraflops). Several other projects have been built using these computers, such as Mach5, it is being built with 1,566 G5 Xserve units, with theoretical speed of over 25 Teraflops. Thus, there are potential gains in computational performance and low-cost in such systems.

We wonder which CPU architecture platform is most suitable to solve problems with massively sequences in computational biology and bioinformatics and aerodynamics applications. In addition, We would like to know the influence in parallel application by assigning computing jobs to available computing nodes in cluster systems through grid technology. In this way, we proceeded to perform performance evaluation between two cluster systems of different CPU architecture and a grid computing environment.

The first cluster system is built with AMD Athlon processors running Fedora Core1, a RedHat and community supported open source project. The second computing system is built using Apple PowerPC processors running MacOS X 10.3.4. The grid computing environment is built basically by interconnecting selected computing nodes of both cluster computing systems.

Three sets of benchmark programs were performed in these two computing platforms, as goal in our experiments for performance evaluation:

- NAS Parallel Benchmarks (NPB programs): a set of eight problems consisting of five kernels, which highlight specific areas of machine performance, and three pseudo-applications, that simulate computational fluid dynamics [1],
- mpptest: a program that measures the performance of some of the basic MPI message passing routines in a variety of situations, implemented by ANL (Argonne National Laboratory, USA)
- ClustalW-MPI: a well-known paralllel version tool of bioinformatics to align multiple protein or nucleotide sequences [12]. One main goal is to evaluate the scalability and performance of clusters of workstations. Multiple sequence alignment is one of the cornerstones of modern bioinformatics. With the aid of this tool, biologists are capable of analyzing structural and functional similarities and differences. Furthermore, they are able to study the sequences patterns conserved and predict their evolutionary relationships. Several researches have focused on the development of fast and accurate multiple sequence alignments algorithms [5], [12], [16]–[18].

This research paper is organized as follows. Section 2 presents the benchmark applications and computing environments for experiments performed in this research. In section 3, the experimental results are presented and discussed. Finally, in section 4, a conclusion and future works.

II. Benchmark Programs and Computing Platforms

Our experiments using benchmark programs were performed on two cluster computing environments and one grid computing environment, which is a combination of two cluster environments.

The first computing environment is built using nine PCs (1 master node + 8 computing nodes), where each one of nodes contains one AMD Athlon 2400+ processor, 1Gb DDR333 memory, 60Gb ATA HD, interconnected via Gigabit Ethernet network.

The second computing environment contains nine workstations (1 master node + 8 computing nodes), each one of nodes contains one 1.6GHz Apple PowerPC G5 processor, 1 Gb DDR400 memory, 80Gb S-ATA HD, frontside bus (FSB) speed 600MHz, interconnected via Gigabit Ethernet network.

The parallel benchmark programs used in our experiments are mainly based on MPI [14]. There are several versions of MPI available, and MPICH [9] is chosen to be installed in our computing system because of its wide availability. MPICH is implemented using P4 device layer, so all communications are performed on top of TCP sockets.

Parallel programs can be started using `mpirun` command in the frontend node shell script, which takes the name of the program and the numbers of the computer nodes to be involved. It remotely spawns the processes on selected cluster computing system. As for our experiments, we used the GNU Compiler Collection (GCC), which has a complete set of compilers for C, C++ and Fortran 77 [9]. As matter of comparison, we executed the same code optimization level `-O3` provided by GNU compiler. Furthermore, the benchmark programs were appropriately installed in these systems, the source data is accessed through the Network File System (NFS) present in these systems.

The NPB benchmark programs are well-known problems for testing the capabilities of parallel computers and parallelization tools. The NPB (NAS Parallel Benchmarks) 2.4 are a set of eight problems consisting of five kernels, which highlight specific areas of machine performance, and three pseudo-applications, that simulate computational fluid dynamics. Problem sizes and verification values are given for its problem sizes as classes S, W, A, B, C, and D.

We briefly describe the NPB benchmark programs which we used to perform our experiments. Kernel `IS` performs ranking an unsorted sequence of integer keys that are uniformly distributed among processors. This benchmark requires frequent communication and the pattern of communication is fully connected graph. Application `LU` solves a finite different discretization of 3-D compressible Navier-Stokes equations by using a Symmetric Successive Over Relaxation (SSOR) numerical scheme. Application `SP` solves 3 uncoupled systems of non-diagonally dominant, scalar, pentad-diagonal equations using the multi-partition algorithm. Basically, it approximates factorization which decouples the *x*,*y*, and *z* dimensions resulting in three sets of narrow-banded, regularly structured systems of linear equations.

ClustalW-MPI [17] is the most widely used mean for globally aligning multiple protein or nucleotide sequences. It incorporated a number of improvements to the alignment algorithm, including sequence weighting, position-specific gap penalties and the automatic choice of a suitable residue comparison matrix at each stage in the multiple alignment. The alignment is achieved via three steps. Recently, Li and Justin Ebedes has reported a parallel version of the ClustalW for cluster environments using MPI interface. The first stage of ClustalW-MPI is to process pairwise alignment. It requires the calculation of a distance matrix. For *n* sequences, the execution time complexity of this stage is $O(n^2)$. The second stage determines the topology of the progressive alignment and produce the guide-tree that determinate order of alignments. The algorithm for generating the guide tree is the neighbor-joining method, by Saitou and Nei [15]. Finally, the last stage obtains the multiple sequence alignment progressively. A series of pairwise alignments are computed using full dynamic programming to align large group of sequences. The sequences are aligned starting from the leaves of the tree toward the root.

The third set of benchmark program we selected to be executed in our computing environments is `mpptest` [10], developed by ANL/Argonne National Laboratory, USA. Basically, it is a program that measures the performance of some of the basic MPI message passing routines in a variety of situations. In addition to the classic ping pong test, mpptest can measure performance with many participating processes (exposing contention and scalability problems) and can adaptively choose the message sizes in order to isolate sudden changes in performance. In addition, mpptest includes a simple "halo" or "ghost cell" exchange test; such tests are often more representative of real performance in applications.

In our experiment, three parallel programs of NAS Parallel Benchmarks Suite has been selected because of the memory limitation and issues with the compiler. The problem size in these benchmark programs grows from class A to class D.

Regarding to application ClustalW-MPI, we would like to know the performance and what proportion of work in each stage is performed when compared with the total work performed. The source sequences randomly select 500 sequences of the genome of *P. aeruginosa* strain PAO1 from *Pseudomonas aeruginosa Community Annotation Project* [3]. The length of the sequences ranged from 50 to 3535 characters.

The options blocking and non-blocking options of mpptest [10] program has been selected in our set of experiments to be performed with three computing platforms described. We would like to know what happens with computational execution time when we overlap or not intensive computations and communication among selected computing nodes when using these options.

During the experimentation using grid computing platform built, all three benchmarks were performed with MPICH-G2 [11] and Globus Toolkit 3.0.2.

Computer Nodes Specification		
	AMD Athlon	Apple PowerPC
Switch	SMC 8598T	SMC 8598T
CPU speed	2.0GHz	1.6GHz
Network speed	1GB/s	1GB/s
Topology	2 switches	2 switches
Middleware	MPICH 1.2.6	MPICH 1.2.6
L2 cache	256Kb	512Kb
Operating system	Fedora Core 1 (RedHat)	MacOS X

TABLE I

SUMMARY OF COMPUTER NODE HARDWARE SPECIFICATIONS.

III. EXPERIMENTS AND RESULTS

In our experiments, we did not performed any type of performance tuning in all parallel benchmark programs with comments supplied by the compiler. The execution time of selected applications was averaged with at least 6 executions of the same application.

The applications IS, LU and SP of NPB (NAS Parallel Benchmarks) were selected to be executed in CISC and RISC architecture based cluster computing platforms. The problem size, types and its sequential execution times are in Table II and Table III. In Table II, "N/A" means that error message had been displayed in that specific experiment during our experimentation. This may be due to the fact that computing nodes involved did not have enough memory for that specific computation. Figure 3 shows the processing time of application IS under different problem sizes.

Still in this Figure 3, it shows that Apple PowerMac cluster computing platform is ahead of other computing platforms for the execution of LU application. Different result is shown the execution of SP problem, where we could see better results in AMD based cluster and grid computing platforms, when compared to Apple based cluster.

Figure 4 shows the execution time of application ClustalW-MPI, in both RISC and CISC architecture based cluster platforms. In order to fairly perform this experiment, we did not pick any specific sequence that may produce the well balanced guide tree. Additionally, the speed of network interface card did not affect the experiment in our observation itself, because the amount of data transmitted is far under the theoretical speed with our tested problem sizes.

Figures 1 and 2 show experiments with mpptest program. The figure 1 shows experimental results with computation and communication *not* overlapping at any time during experimentation, while we perform experiments with MPI point-to-point communication send/receive, whether it is blocking or non-blocking.

The figure 2 shows experimental results when computation and communication were able to be overlapped, in situations when MPI point-to-point communication send/receive is blocking or non-blocking.

We can observe from figures 1 and 2 that both point-to-point communication send/receive is blocking or non-blocking, Apple based cluster platform shows better performance than AMD based cluster and grid platforms. We varied

Application Sequential Times			
Program	Program Size	Sequential Time(s)	Platform
IS	2^{25},Iterations=10	N/A	Apple
LU	102x102x102,Iterations=250	1491.31	Apple
SP	102x102x102,Iterations=400	2134.33	Apple

TABLE II

SEQUENTIAL EXECUTION TIMES USING APPLE PROCESSOR BASED CLUSTER SYSTEM.

Sequential Execution Times			
Program	Program Size	Sequential Time(s)	Platform
IS	2^{25},Iterations=10	34.99	AMD
LU	102x102x102,Iterations=250	1854.45	AMD
SP	102x102x102,Iterations=400	2112.50	AMD

TABLE III

SEQUENTIAL EXECUTION TIMES USING AMD PROCESSOR BASED PC CLUSTER SYSTEM.

our message size from a few bytes to some megabytes, and as the message size increases, Apple based cluster platform could handle communication better than other two platforms. For instance, for a message size close to 1xE+07 bytes, Apple based cluster could handle in either blocking and non-blocking modes by using around 5000μs, while in other two platforms we need more than two-fold of transmission time of the same message, using around 11000μs.

We can observe in figure 2 results when we overlap computation with communication, either in blocking and non-blocking modes. Again, for transmission of the same message close to 1xE+07 bytes, in Apple based cluster it is needed around 3000μs, while in other two computing platforms, it is needed around 4500μs.

IV. CONCLUSIONS AND FUTURE WORK

In this research paper, we show some results using well known parallel benchmarks and bioinformatics sequences alignment application under three different computing platforms, being the first platform a CISC architecture based cluster, the second platform is a RISC architecture based cluster, while the third platform is built using grid technology.

From the experimental results, it may indicate that CISC based architecture computers can be the inexpensive workstation clusters. We would like to emphasize that the results shown in this paper **did not** take advantages of code optimization using Apple PowerPC G5 processors' Velocity Engine. The AltiVec instruction set allows operation on multiple bits within the 128-bit wide registers. This combination of new instructions, operated in parallel on multiple bits, and wider registers, provide speed enhancements of up to 30x on operations that are common in media processing.

Although softwares developed for RISC processors must handle more operations than traditional CISC processors, RISC processors have advantages in applications that benefit from faster instruction execution, such as engineering and graphics workstations and parallel-processing systems. Additionally, they are less costly to design, test, and manufacture. RISC technology processors are cost-effective, low power, small die size and high performance micro-controlled. Nowadays, RISC 16/32-bit technology include INTEL StrongARM and XScale family processors, Samsung's S3C2410X01 ARM processor and ARM's microprocessors boost most handheld devices and general applications available.

Thus, we believe that systems based on PowerPC processors has the potential and actually suitable to be a high performance scientific computing platform, either using cluster or grid technologies. RISC technology processors is promise

As next step in our research, we would like to optimize the bioinformatics application ClustalW-MPI's parallel code for PowerPC processors, by taking instruction-level advantages with Apple's Altivec technology. Additionally, in order to improve the speedup of the ClustalW-MPI, we need to split up a single alignment task among multiple processors. This will need a new algorithm for decrease the costs of communications between each processor, speeding up the total performance of this application.

REFERENCES

[1] D. Bailey, T. Harris, W. Saphir, R. Van der Wijngaart, A. Woo, and M. Yarrow. The nas parallel benchmarks 2.0. *Technical Report NAS-95-020*, NASA Ames Research Center, 1995.

[2] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, 23:187–200, 2001.

[3] Stover CK and et al. Complete genome sequence of pseudomonas aeruginosa pa01, an opportunistic pathogen. *Nature*, 406(6799):947–948, 2000.

[4] L. M. Ni D. K. Panda. Special issue on workstation clusters and network-based computing: Guest editors' introduction. *Journal of Parallel and Distributed Computing*, 43:63–64, 1997.

[5] Justin Ebedes and Amitava Datta. Multiple sequence alignment in parallel on a workstation cluster. *Bioinformatics*, 20:1193–1195, 2004.

[6] I. Foster and N. Karonis. A grid-enabled mpi: Message passing in heterogeneous distributed computing systems. *Proc. Supercomputing 98 (SC98),*, 1998.

[7] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Supercomputer Applications*, 11(2):115–128, 1997.

[8] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann Publishers, 1999.

[9] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6):789–828, 1996.

[10] William Gropp and Ewing Lusk. Reproducible measurements of mpi performance characteristics. *PVM/MPI*, pages 11–18, 1999.

[11] N. Karonis, B. Toonen, and I. Foster. Mpich-g2: A grid-enabled implementation of the message passing interface. *Journal of Parallel and Distributed Computing*, 63(5):551–563, 2003.

[12] Kuo-Bin Li. Clustalw-mpi: Clustalw analysis using distributed and parallel computing. *Bioinformatics*, 19(12):1585–1586, 2003.

[13] M. Lobosco, V. S. Costa, and C. L. de Amorim. Performance evaluation of fast ethernet,giganet and myrinet on a cluster. *Computational Science - ICCS 2002, International Conference, Amsterdam, The Netherlands, April 21-24, 2002. Proceedings, Part I.*, 2329:296–305, 2002.

[14] M.Snir, S.W.Otto, S.Huss-Lederman, D.W.Walker, and J.Dongarra. *MPI: The Complete Reference.* MIT Press, 1996.

[15] N. Saitou and M.Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425, 1987.

[16] Jens Stoye, Vincent Moulton, and Andreas W. M. Dress. Dca: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Computer Applications in the Biosciences*, 13(6):625–626, 1997.

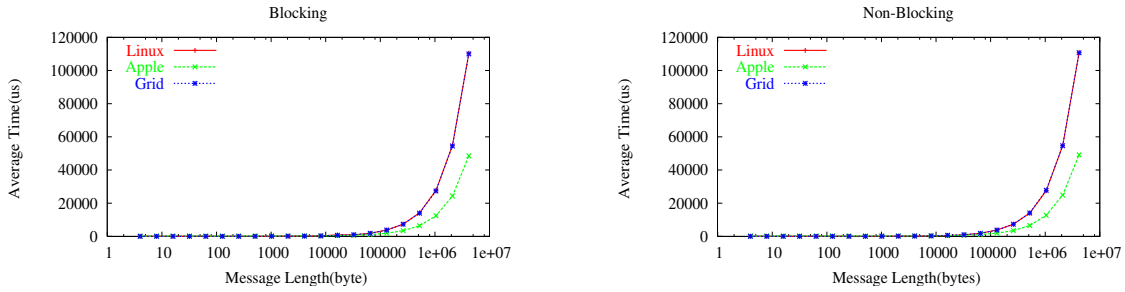

Fig. 1. Performance evaluation of Mpptest.

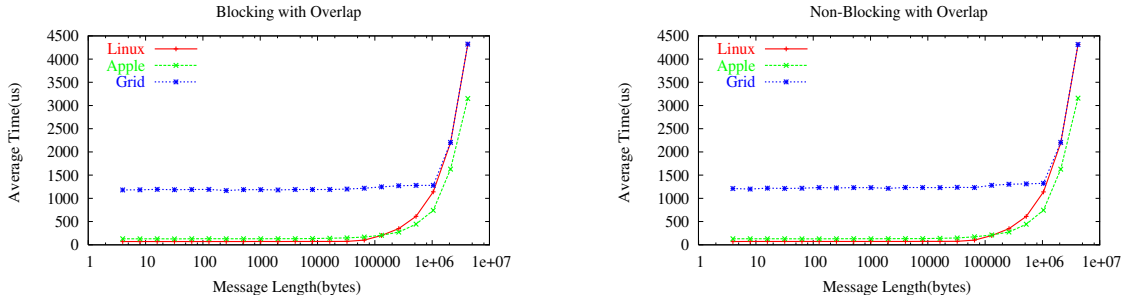

Fig. 2. Performance evaluation of Mpptest.

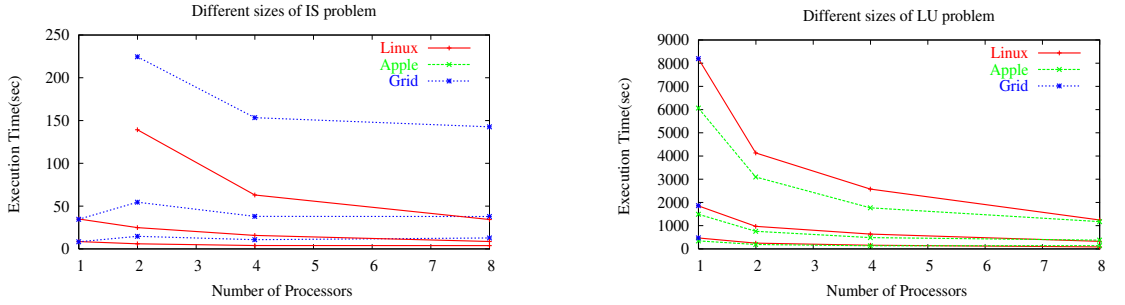

Fig. 3. Performance evaluation of NAS/IS and LU.

[17] J. D. Thompson, D. G. Higgs, and T. J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22(22):4673–4680, 1994.

[18] Tieng K. Yap, Ophir Frieder, and Robert L. Martino. Parallel computation in biological sequence analysis. *IEEE Trans. Parallel Distrib. Syst.*, 9(3):283–294, 1998.

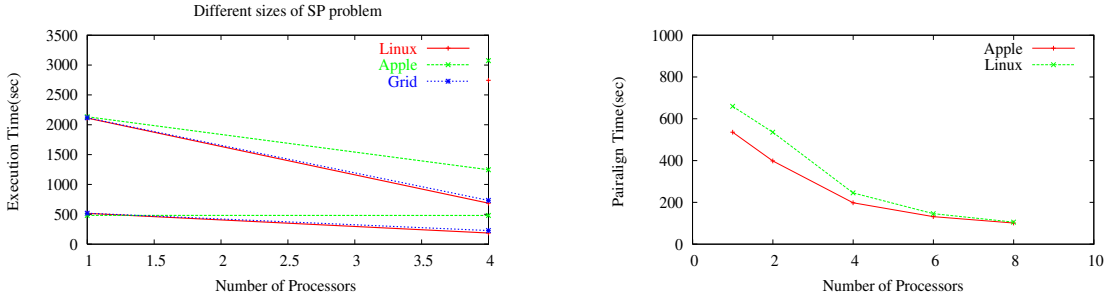

Fig. 4. Performance evaluation of NAS/SP and ClustalW-MPI.