2001 년 한국정보처리학회 추계 학술발표논문집 제 8 권 제 2 호

# Tera-scale cluster 개발을 위한 실험

홍정우, 박형우, 이상산

슈퍼컴퓨팅개발실/슈퍼컴퓨팅센터
한국과학기술정보연구원
e-mail : jwhong@hpcnet.ne.kr

# Experiments to build tera-scale cluster

Jeong Woo Hong, Hyung Woo Park, Sang San Lee
Dept. of Supercomputing development/Supercomputing Center/
Korea Institute of Science and Technology Information

**Summary**

At the end of 1999, the TeraCluster project in the KISTI Supercomputing Center was initiated to explore the possibility of PC clusters as a scientific computing platform to replace the Cray T3E system in KISTI by the year 2002. In order to understand whether an application is scalable to tera-flops sized cluster system, running test is inevitable. Extensive performance tests using well-known benchmarking codes with real applications' characteristics in them were carried out with different combinations of CPUs, system boards, network devices. The lessen learned shows the relationships between system performances and varied applications' different needs resulting in promises of How-Tos in building large scale cluster system. The 64/16 node clusters with Alpha EV6(466MHz), Pentium III(667 MHz) i inter-node network of Fast Ethernet, SCI[1] and Myrinet[2] were evaluated. More detailed specifications of the Linux clusters are described in Table 1†.

## 1. Introduction

Because the economical facets of cluster computing appeal broad audience, it has become one of the easy-to-find methods of high performance computing. However, building one's own small to medium scale cluster is one thing and building larger scale cluster is another for a few reasons including maintenance, optimization, etc. In addition to that even its advantageous side of application specific hardware reconfigurability can put some burden on system planner who's task is providing a cluster system whose scale goes up to a few hundred computing nodes because of the sheer numbers of components that determine system performance and also errors.

KISTI Supercomputing Center's plan to replace CrayT3E system with cluster systems demands acquiring specific requirements from applications for the systems if KISTI supercomputing center will support the users with just what their applications need. Our needs has driven us to perform extensive benchmarking tests using a numbers of known benchmark codes and our own major applications on different scale of varied system configurations.

We assumed that there would be relaships between basic system performance numbers and practical application's. Based on that assumption, benchmarks were tested to earn the understandings of typical cluster systems performance. With this we expect to set some principles to help our users of implementing their codes to larger scale cluster system and fair system requirements for future cluster computing related researches.

One can find systems tested from table 1. The configurations are for each configuration's inherent scalability tests and comparisons.

† These tests have been generously supported by Compaq-Korea, LinuxOne, Samsung Electronics, Samsung Corp, Zion.

Table 1. Properties of System Architecture

| | CPU | OS | Network | | | No. CPU | Cooperation Company |
|---|---|---|---|---|---|---|---|
| | | | Types | Latency* (PingPong,ms) | Bandwidth* (Peak, byte/sec) | | |
| DS10 | 466 MHz EV6 Alpha | Linux (Kernel 2.2.0) | Fast Ethernet | 140 | 8.9 | 64 | LinuxOne, Compaq Korea |
| UP2000 | 466 MHz EV6 Alpha | Linux (Kernel 2.2.0) | Myrinet | 13 | 139.7 | 64 | ZionLinuxSystems, Samsung Electronics, Samsung Corp. |
| Intel(s) | 667 MHz Pentium III | Linux (Kernel 2.2.0) | SCI | 6 | 24.93 | 16 | Scali/Dolphin |
| Intel(m) | 667 MHz Pentium III | Linux (Kernel 2.2.0) | Myrinet | 11 | 141.7 | 16 | Samsung Corp |
| CrayT3E | 450 MHz EV5 Alpha | UNICOS/mk | 3D Torus | 13 | 157 | 128 | |

* Obtained by PMB[3] test in the KISTI supercomputing Center.

## 2. Benchmarks and performance Numbers

The benchmarks chosen for essential system performance tasks are in table 2 with rationales of choice.

Table 2. Known benchmarks

| Name | Description | Rationale |
|---|---|---|
| Linpack [4] | Linear algebra computing library performance test | • Standard benchmark problem<br>• Top 500 ranking rating |
| NPB2.3 [5] | Application benchmark set from computational fluid dynamics | • Selected sets of problems with different characteristics<br>• CFD application's view |
| PMB(Pallas MPI Benchmark) [3] | MPI primitives performance test | • An application benchmark<br>• Network performance from the view of application<br>• Communication primitives can provide programming direction |

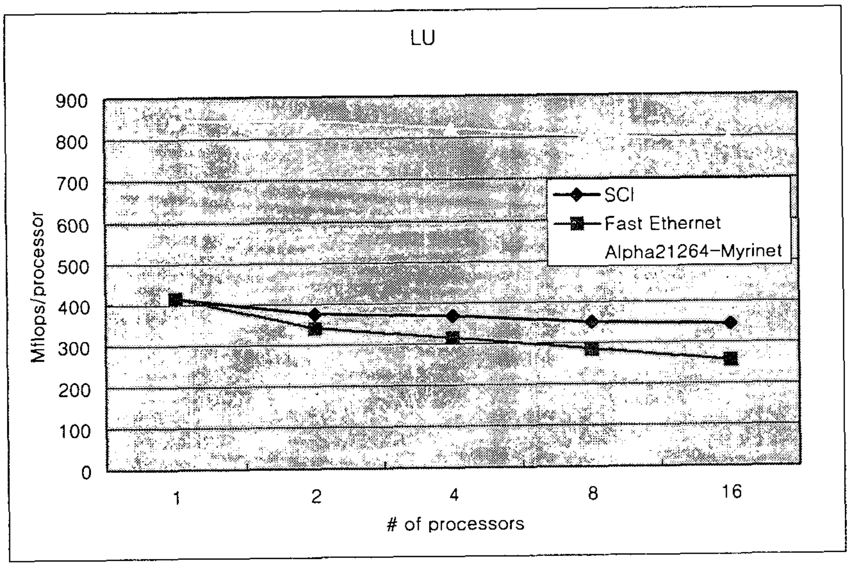


Fig. 1. Linpack performance graph

Among three of the benchmarks MPI primitives performance test is most noticeable. Since other benchmarks show limited facets of system performance, elemental numbers are more appealing. For the most of the real applications, network performance is most noticed factor. However ours benchmark tests find that the Linpack performance numbers, which are still used to evaluate cluster systems, are not much influenced by network performances as shown in Fig.1. This encourages our interests to find more realistic measures to estimate cluster system performances to properly satisfy our computational needs. NPB benchmarks can suggests more realistic performance numbers. However, their predefined problems sizes and computational characteristics whose computation occurs within physical memory spaces does not reflect real needs of resource hogging demands. One example is a structural analysis code[2] in which the problem chosen as a good candidates for cluster computing application demands far much memory spaces that its program code does memory swapping which could be noticed in Fig.10 from structural analysis parts of their report[2].

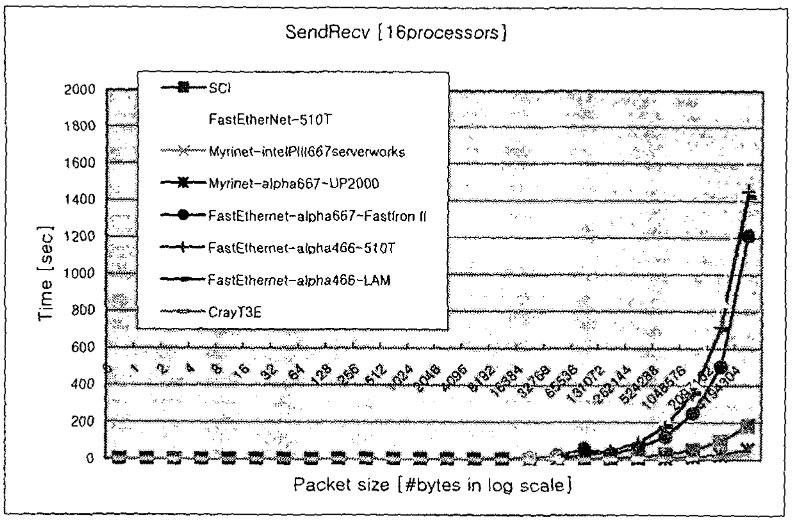


Fig.2. SendRecv Time

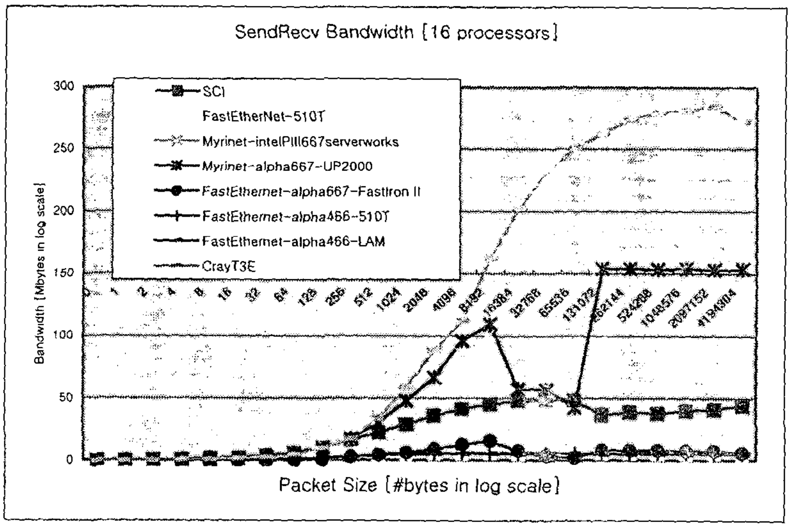
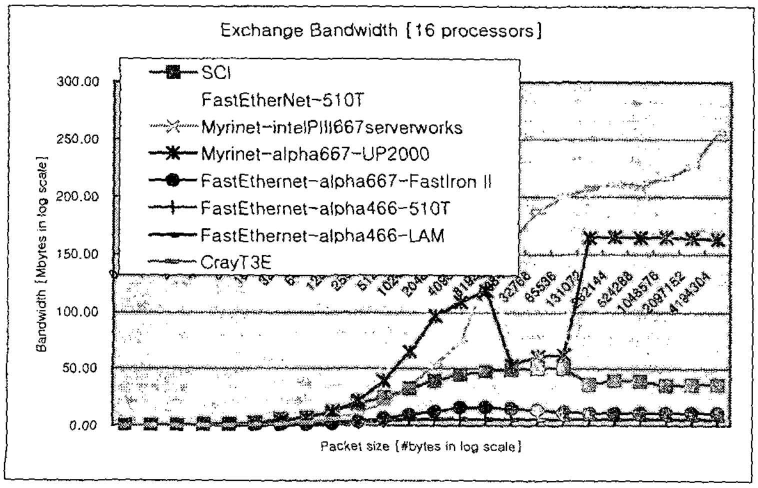


Fig.3. SendRecv bandwidth



Another usefulness of elemental performance measuring is their usage as optimization guidelines and scalability enhancing techniques. In figures of 2 to 6, MPI primitives show non-regular variations. This leaves marginal expectations of system dependent optimization techniques to be effective on grand challenge scale computing problems.

Physics *ab initio* problems [2] using global communication primitives can achieve bigger scale computations with localization technique for their communications such as substituting one global communication with two local group communications according to the performance guidelines from communication performances graphs which is shown in fig.10(a), and fig.10(b)

Fig.4. Exchange time

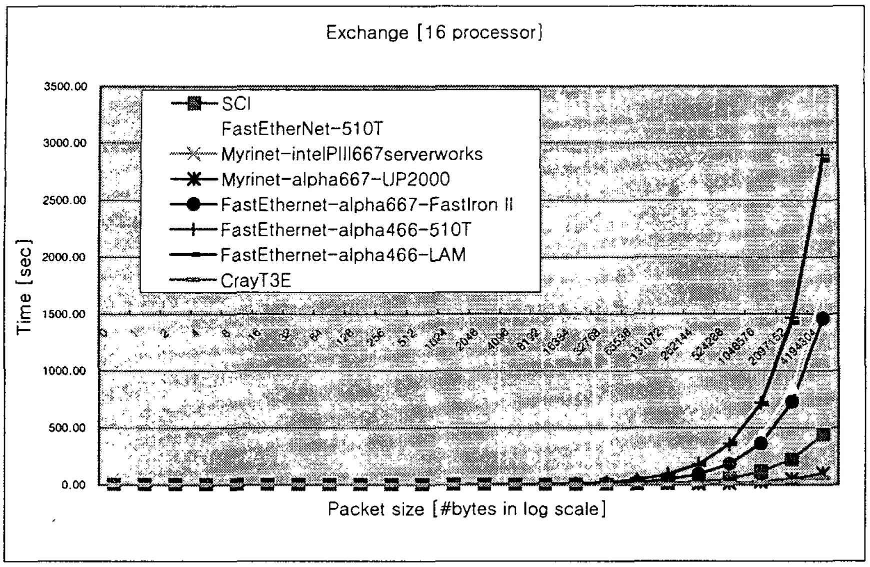


Fig..5 Exchange bandwidth

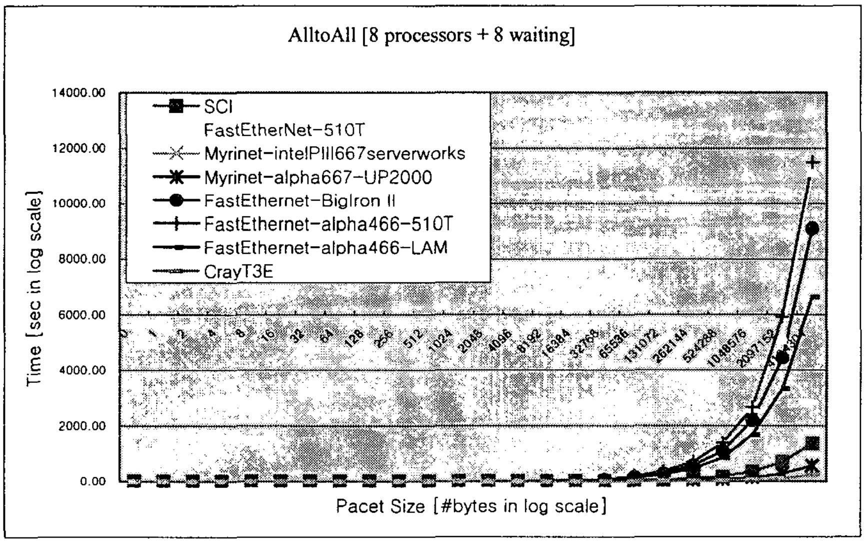


Fig.6. AlltoAll time

## 3. Conclusion

Well known benchmarks' lack of reflecting proper computational demands encourages to seek another guide lines to be used for cluster system building. And rather basic performance measurements such as communication primitives' performance and e CPU performance only, or disk input/output numbers can give better insights into cluster system performances.

By reviewing real applications' performance benchmarking results on cluster systems with various configurations in comparison with well known elemental benchmarking codes can give guide lines to build application specifically optimized cluster system. However, the rather basic performance numbers can still be useful in planning bigger computing from the points of SI(system integration) which is another noticed characteristics of the cluster system concepts.

These observations can also be used in building teraflops-scale cluster system that demands the system planners to give their customers of how to optimize their code to achieve best from ever lacking system resources.

## Reference

[1] Roger S. Pressman. "Software Engineering, A Practitioner's Approach", 3rd Ed. McGraw Hill, 1997

[2] Sangsan Lee, Minsu Joh, Jeongho Kim, Sangju lee, Hongsuk Lee, Kumwon cho, Kyungsu Kim, Jeongwoo Hong , "Building and Benchmakring of 64 node Alpha/Myrinet Cluster", KISTI Supercomputing Center, Technical Report TR00-0410-004.

[3] Pallas MPI Benchmarks – PMB, Part MPI-1.

[4] Linpack Benchmark, http://www.top500.org/lists/linpack.html

[5] NAS Parallel Benchmarks (NPB), http://www.nas.nasa.gov/Research/Software/swdescription.html#NPB