Journal of the Korean
Data & Information Science Society
2006, Vol. 17, No. 2, pp. 569 ~ 579

PC-Based Hybrid Grid Computing for Huge Biological Data Processing[1]

Wan-Sup Cho[2] · Tae-Kyung Kim[3] · Jong-Hwa Na[4]

Abstract

Recently, the amount of genome sequence is increasing rapidly due to advanced computational techniques and experimental tools in the biological area. Sequence comparisons are very useful operations to predict the functions of the genes or proteins. However, it takes too much time to compare long sequence data and there are many research results for fast sequence comparisons. In this paper, we propose a hybrid grid system to improve the performance of the sequence comparisons based on the LanLinux system. Compared with conventional approaches, hybrid grid is easy to construct, maintain, and manage because there is no need to install SWs for every node. As a real experiment, we constructed an orthologous database for 89 prokaryotes just in a week under hybrid grid; note that it requires 33 weeks on a single computer.

Keywords : Bioinformatics, Grid Computing, LanLinux, Sequence Comparison

1. Introduction

Recently, the amount of biological data is increasing rapidly with the improvement of the experimental and computational techniques. GenBank, a

[1] This work was supported by the research grant of the Chungbuk National University in 2004.

[2] Professor, Dept. of Management Information Systems, Chungbuk National University, Cheong-ju, Chungbuk, 361-763, Korea
E-mail : wscho@cbnu.ac.kr

[3] Dept. of Industrial Information Engineering, Chungbuk National University, Cheong-ju, Chungbuk, 361-763, Korea

[4] Corresponding Author : Professor, Dept. of Information and Statistics & Institute for Basic Science Research, Chungbuk National University, Cheong-ju, Chungbuk, 361-763, Korea
E-mail : cherin@cbnu.ac.kr

representative database for DNA sequence data, continues to grow at an exponential rate [30]. Historically, it has been doubling in size per 18 months, but the rate has accelerated to doubling every 15 months recently. That is, the growth speed of the data overtakes the improvement speed of the microchip (Moore's Law). Therefore, it takes too much time to process massive biological data such as genome sequences in a single server and therefore we need distributed or parallel computing approaches.

As a solution of the performance problem, computing grid, which is a collection of geographically distributed computational resources (PCs) connected through Internet [1,2,3,4], has been utilized in the bioinformatics area. Although supercomputers or cluster systems are alternatives, they are expensive and not scalable.

We propose a hybrid grid system with combination of existing grid middleware such as Globus [29] with LanLinux system [25], and PCs are only used to process huge amount of biological data. We have implemented well-known sequence alignment tool BLAST [1, 2, 3], and evaluated performance of hybrid grid for real data. LanLinux System is originated from EtherBoot Project [22]. It enables PCs to boot Linux, DOS, and other open source OSs such as FreeBSD from server without using local hard disks. After booting, each PC does a role as a complete system and a member of the grid. Originally, it was developed to provide the environment for Linux education and management of the PC rooms, but we are adopting it as a grid system construction tool.

LanLinux is used as an auxiliary method to make up for the weak points of the current grid systems. The most challenging problems in constructing grid with PCs come from the heterogeneity and changeable environments. There are various operating systems such as several versions of windows, Linux, UNIX and so on. Furthermore, the environment of the PCs changes frequently; users usually reset-up the PCs at least once or more a year and install several kinds of the programs which can affect the operation of the grid middleware.

However, LanLinux provides a unified grid environment for all member nodes. In addition, we don't need to install grid middleware for all nodes individually and don't worry about changing the local systems.

As a real experiment, we constructed an orthologous database for 89 prokaryotes just in a week by using the hybrid grid system; note that it requires 33 weeks on a single computer.

The paper is organized as follows. In Section 2, we present related work. In Section 3, we describe the hybrid grid environment in detail. In Section 4, we show the experiments for sequence comparisons and present the experimental results on BioGrid. In Section 5, we conclude the paper.

2. Related Work

In this section, we introduce grid computing and its applications to bioinformatics.

2.1 Grid Computing

Grid is the computing environment which uses the inter-connected resources (PCs) safely to overcome the limitations of performance and scalability. Although the parallel and cluster computers have been developed to enhance performance, they are generally expensive and un-scalable. Grid has been introduced to overcome these weak points.

There are several well-known grid middlewares such as Globus, Condor, and so on. Globus Toolkit [9] is a famous grid construction tool, and admitted as the de-facto standard of the grid. Globus toolkit has been developed to mainly connect the super-computers since 1996. Globus Toolkit offers software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. They can be used independently or together to develop grid applications.

Many of the grid projects are mainly dealing with high-performance resources such as clusters or supercomputers that are not changeable and more stable than PCs; HGP (Human Genome Project), NASA IPG [31], European data grid [32], and Euro grid projects [35] are the representative grid projects. SETI@HOME[33] and KOREA@HOME [34] projects are focusing on the PCs and dealing with scientific problems such as the climate prediction and drug discovery. Users should individually install grid middlewares to participate in @HOME project. As environments of the user PCs are changeable, dynamic and heterogeneous, the construction and maintenance of grid are not persistent. Furthermore, grid middewares may have trouble with other softwares. To address these problems, we apply the EtherBoot technology [22] in the grid construction.

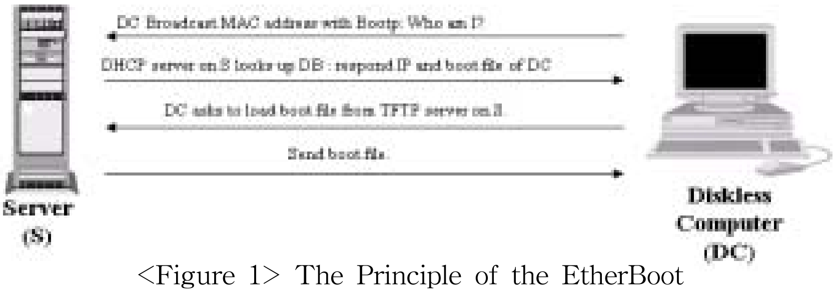
2.2 The Application of Grid in Bioinformatics

The amount of the genome sequences is dramatically increasing because of the advanced experimental tools and computers in biology. Huge amount of data requires high performance computing system. For example, construction of the orthologous database for more than hundred species requires more than 33 weeks by using a single computer.

The main applications of the grid are sequence alignment [14] and protein secondary structure prediction [15] in which large scale sequence data is involved. In these applications, there are so many works to be processed, and each of them takes long time to be completed. A lot of researchers are now trying to find the solution from grid computing. Wang, et al. (2002) and Soojin, et al. (2004) applied grid computing in protein secondary structure prediction and protein sequence alignment, respectively.

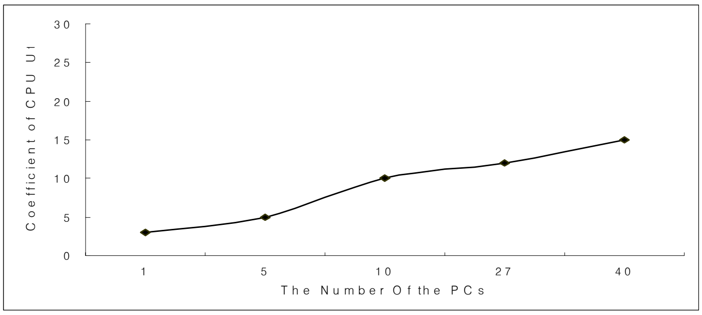
2.3 LanLinux System

LanLinux supports PCs to boot Linux system from the server without utilizing the local disk by attaching a special hardware card to the PCs. This is originated from EtherBoot, which is "a software package for creating ROM images that can download code over an Ethernet network to be executed on an x86 computer"[22]. EtherBoot is used in various applications; X-terminal, clusters of computer servers, and so on. Figure 1 shows the principle of the EtherBoot.


DC Broadcast MAC address with Bootp: Who am I?
DHCP server on S looks up DB : respond IP and boot file of DC
DC asks to load boot file from TFTP server on S
Send boot file.
Server
(S)
Diskless
Computer
(DC)

<Figure 1> The Principle of the EtherBoot

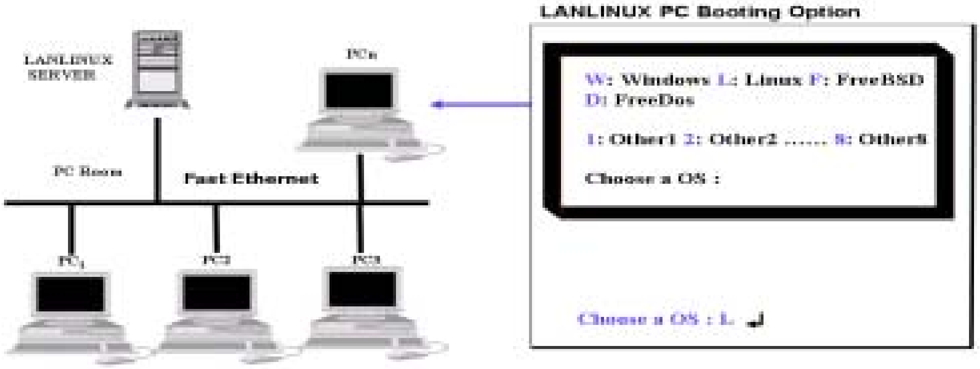
When Diskless Computer (DC) boots, it broadcasts MAC address with Bootp. Then DHCP of Server (S) searches the databases, responses the IP and boot file (kernel) of DC. DC requests to load boot file from TFTP server on S. S sends boot files and DC starts to boot. After booting the system, DC maintains the system through the communication with S. EtherBoot technology is widely used to construct X-Terminal, which shares the resources and the software of the main frame computer. However, LanLinux enables PCs to utilize their own resources and share the software of the server after booting. Note that even large number of the PCs, overload of S is not serious as shown in Figure 2 (less than 15%).



<Figure 2> CPU Utilization of server node as increasing the number of the DCs

3. Grid Environment

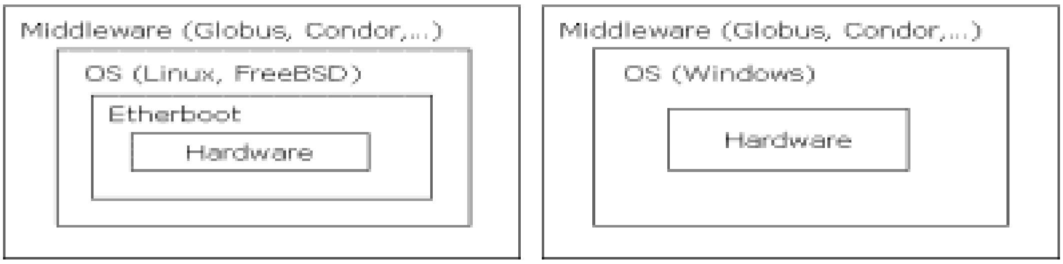
In this Section, we introduce the hybrid grid environment in detail. LanLinux basically provides PCs with options about whether they boot from the LanLinux server by network or local disk as shown in Figure 3.



```
W: Windows L: Linux F: FreeBSD
D: FreeDos

1: Other1 2: Other2 ...... 8: Other8

Choose a OS :


Choose a OS : L ↲
```

<Figure 3> The initial step of the booting: provides the booting options to each PC

In Figure 3, if the user selects 'W', then the PCs boots from the local disk; if the user chooses 'L', then the PC boots from the LanLinux server by the steps of the Figure 1 and then the PC becomes a member of the grid. System architecture of the node is shown in Figure 4(a).


Middleware (Globus, Condor,...)
OS (Linux, FreeBSD)
Etherboot
Hardware
Middleware (Globus, Condor,...)
OS (Windows)
Hardware

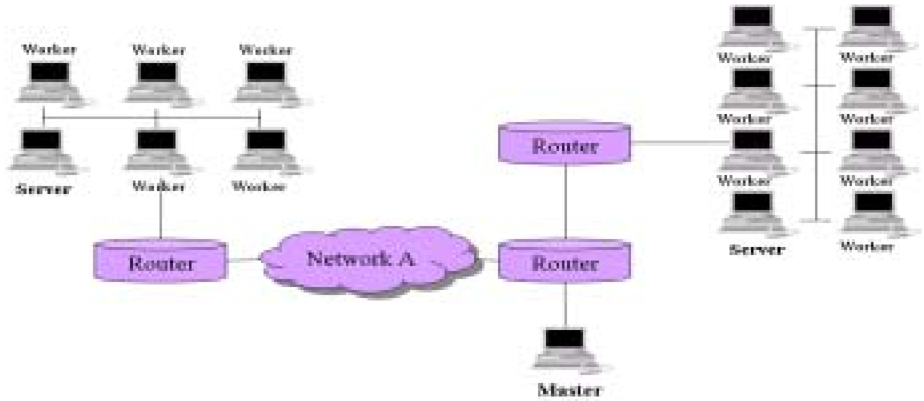
(a) Hybrid nodes (b) General node

<Figure 4> The differences of the structure between Hybrid (a) and general nodes (b)

As shown in Figure 4(a), hybrid nodes have the EtherBoot layer unlike the general nodes in Figure 4(b). EtherBoot layer enables the PC to run Linux system regardless of the local system, which may be heterogeneous and changeable continuously. Grid middleware is operated on the Linux system. Therefore, it guarantees that the nodes on grid have the unified environment (Linux) and are maintained safely for a long time.

In case of the general nodes, various programs may be installed with intention or secrecy like adware, spyware, and other Active-X programs. These programs can be conflicted with grid middleware and have a bad influence. Although grid middlewares such as several @HOME projects offer the comfortable environment to make grid, nobody can guarantee the security and the stability to the PCs.

Hybrid grid consists of the three types of the nodes as shown in Figure 5; a master, servers, and worker nodes.



<Figure 5> The structure of the LanLinux-base grid

The master node has all jobs to be performed and the information for the server nodes such as basic information of IP, resource specification, status whether it operate or not, and the number of the worker nodes. Master node delivers the jobs as the number of the worker nodes as to each server node. The server node is the LanLinux server and provides kernel and the system files to the worker nodes. The server node identifies the information of the worker nodes and gives the jobs according to the status of them. Each worker node performs the jobs given by the server node repeatedly.

The LanLinux-base grid environment has several advantages in aspects of the construction, management, and maintenance of the grid.

First, each PC doesn't need to install the grid middlewares because all information of the worker node is contained in the server node. After attaching the BootROM card to PCs, we only concentrate on the server nodes. Therefore, the construction of the grid is easy and comfortable. Note that we should install grid program to the PCs individually and PCs are re-setup periodically.

Second, each node may be used either Linux system or local system depending on the user's selection. User can change the role of his PC from a member of the grid to the local system anytime by rebooting his PC.

Third, hybrid grid provides a unified platform with limited changes. This is very beneficial to develop grid-enabled applications. Note that the most challenging problems in @HOME are heterogeneous platform with changes of the member PCs.

4. Performance

In this Section, we perform the high-throughput sequence comparisons on the hybrid grid and evaluate its performance.

4.1 Sequence Comparisons for Orthologous Database Construction

We first define notations to be used in the sequence comparisons. Table 1 shows notation in the performance analysis.

<Table 1> Notification used sequence comparisons

Definition	Description
G	A set of genomes to be used in the orthologous clustering
G	The genome of the species i ($G_i \in G$).
N	The number of species in the orthologous clustering.
NG_i	The number of genes in the genome G_i.
$SeqAlign(G_i,G_i)$	Genome comparison operation between two genomes G_i and G_j
$T(G_i,G_j)$	Comparison time between G_i and G_j (20minutes ~ 4 hours)

We performed the sequence comparisons to construct the orthologous database [16, 11] for 89 prokaryotes, which contains the functionally similar gene groups based on the sequence similarity. Constructing an orthologous database basically requires a lot of sequence comparisons. If we process N organisms, we have to perform ${}_NC_2$ genome-genome comparisons (3916 comparisons for 89 organisms) as in (4.1).

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} SeqAlign(G_i, G_j) \qquad (4.1)$$

Furthermore, each genome comparison needs as many operations as the formula (4.2), each of which takes at least from 20 minuets to 4 hours in a server.

$$SeqAlign(G_i, G_j) = N_{G_i} \times N_{G_j} \qquad (4.2)$$

The total time to process all comparisons for 89 organisms becomes (4.3).

$$\sum_{i=1}^{88} \sum_{j=i+1}^{89} T(G_i, G_j) \cong 2 months \sim 21 months \qquad (4.3)$$

4.2 Experimental Results

In the experiment, we compared 89C2 genome-genome sequence comparisons. In the sequence comparison, each comparison can be processed independently. Therefore, we solve the problem by using hybrid grid.

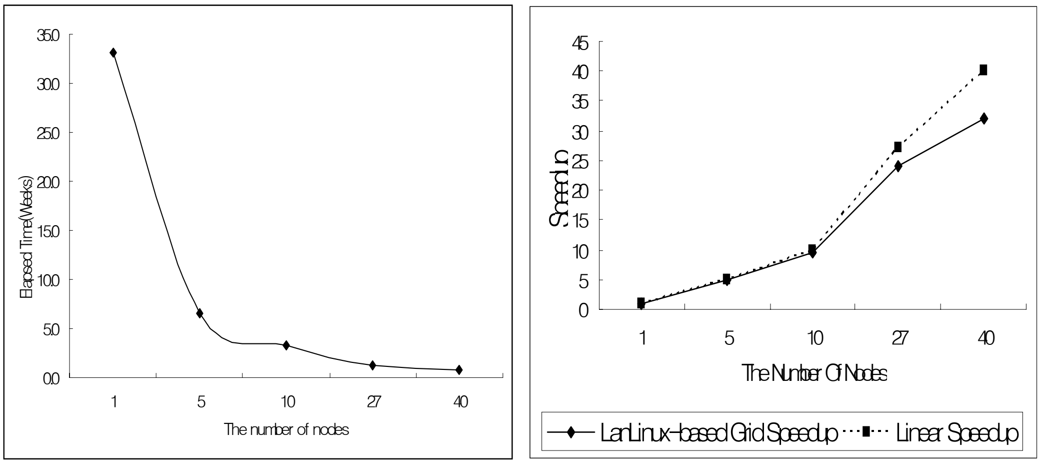
Each server node makes worker nodes to process their jobs, assembles the results from the worker nodes, and gives them to master repeatedly. Table 2 shows grid environments: 27 PCs with a server. Each PC has Pentium 4 3GHz CPU.

<Table 2> Experimental Environment

Location	The Number of the Server	The Number of the PCs
Class A	1	13
Class B		14

Theoretically, each PC has the power of 6G Flops and overall performance is near 162G Flops as hybrid grid can utilize the full power of PCs. Note that most of the general grid uses just a small portion of the resources. Now, Korea@Home provides 465G Flops computing power for 937 PCs because it uses only small portion of the PCs.

Figure 6 shows reduction of the computing time as the number of PCs increases.



(a) The elapsed time for the number of the PCs

(b) The speedup of the Hybrid grid.

<Figure 6> The performance evaluation of the Hybrid grid

Note that the performance of the hybrid grid is proportional to the number of the PCs as shown in Figure 6. This guarantees the efficiency of the 90%.

5. Conclusion

We proposed a hybrid grid and applied it in the analysis of huge biological sequences. Hybrid grid uses the EtherBoot technology to overcome the heterogeneity and change of the member PCs which are the main challenges of the existing grid. This approach is very efficient to construct, maintain and manage the grid because LanLinux provides a unified platform regardless of the local system. Member PCs may be used either local system or a member of grid depending on the user's selection at the booting time. Furthermore, hybrid grid supports high-performance with cheap expenses. In the experiment, we completed the orthologous database construction for 89 prokaryotes just in a week. Note that it requires 33 weeks when we use a single computer. In future, we will apply various applications to the grid system.

References

1. Altschul, S. F., and Gish, W. (1996). Local alignment statistics, Methods in Enzymology, 266, 460-480.
2. Altschul, S. F., Madden T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, and W., Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, Nucleic Acids Research, 25, 3389-3402.
3. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic Local Alignment Search Tool, Journal of Molecular Biology, 215, 403-410.
4. Baker, A. M. and Fox, G. C. (1999). Metacomputing: Harnessing Informal Supercomputers, In High Performance Cluster Computing, Prentice-Hall.
5. Brewer, E. (1997). Clustering: Multiply and Conque, Data Communications.
6. Chi, E., Shoop, E., Carlis, J., Retzel, E. and Riedl, R. (1997). Efficiency of shared-memory multiprocessors for a genetic sequence similarity search algorithm, Technical Report, Computer Science Dept., University of Minnesota.
7. Foster, I. (2002). The Grid: A New Infrastructure for 21st Century Science, Physics Today, 55(2), 42-47.
8. Foster, I. and Kesselman, C. (1999). The Grid: Blueprint for a New

Computing Infrastructure, Morgan Kaufmann, 1st edition.

9. Foster, I. and Kesselman, C. (1997). Globus: A Metacomputing Infrastructure Toolkit, International Journal of Supercomputer Applications, 11(2), 115-128.
10. Foster, I., Kesselman, C. and Tuecke, S. (2001). The Anatomy of the Grid : Enabling Scalable Virtual Organizations, International Journal of Supercomputer Application, 15(3), 1-24.
11. Gish, W. and States, D. (1993). Identification of protein coding regions by database similarity search, Nature Genetics, 3, 266-272.
12. James, H. A. (1999). Scheduling in Metacomputing Systems, BSc(Ma&Comp Sc) (Hons)
13. Kuo, Y. and Yang, C. (2003). Apply Parallel Bioinformatics Applications on Linux PC Clusters, Tunghai Science, 125-141.
14. Kuo, Y. L., Yang, C.T., Lai, C.L., Tseng, and T.M. (2004). Construct a Grid Computing Environment for Bioinformatics, In Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks(ISPAN'04), 1087-4089.
15. Soojin, L., Min-Kyu, C., Jin-Won, Jung, Jai-Hoon, Kim., and Weontae, Lee (2004). Exploring protein fold space by secondary structure prediction using data distribution method on Grid platform, Bioinformatics, Advance Access published on July 29, 20(18) 3500-3507.
16. Tatusov, R. L., Galperin, M. Y., Natale, D. A., and Koonin, E. V. (1999). The COG Database: A Tool for Genomic-Scale Analysis of Protein Function and Evolution, Nucleic Acids Res. 28, 33-36.
17. Teo, Y., Wang, X., and Ng, Y. K. (2004). GLAD: a system for developing and deploying large-scale bioinformatics Grid, Bioinformatics, Advance Access published on September 23.
18. Wang, L. (2002). Biogrid Computing Platform: Parallel computing for protein alignment analysis, HPC Asia'02, Bangalore, India.
19. Yamanishi, Y., Yoshizawa, A. C., and Itoh, M. (2003). Extraction of Organism Groups from Whole Genome Comparisons, Genomic Information, 14, 438-439.
20. Yang, C. and Hung, C. (2001). High-performance computing on low-cost PCs based SMPs clusters, In Proc. of the 2001 National Computer Symposium (NCS 2001), Taipei, Taiwan, 149-156.
21. Biology Workbench Portal, http://workbench.dsc.edu/
22. EtherBoot Project, http://etherboot.sourceforge.net/
23. Global Grid Forum, http://www.ggf.org/
24. KISTI Grid Testbed, http://gridtest.hpcnet.ne.kr/
25. LanLinux, http://www.lanlinux.com/
26. LHC - The Large Hadron Collider Home Page, http://lhc-new-homepage.web.cern.ch/

27. My Grid, http://www.mygrid.org.uk/
28. NASA Launchpad Portal, http://portal.ipg.nasa.gov/
29. The Globus Project, http://www.globus.org/
30. NCBI, http://www.ncbi.nlm.nih.gov/
31. NASA IPG, http://www.ipg.nasa.gov/
32. EU DataGrid, http://www.ipg.nasa.gov/
33. SETI@HOME, http://setiathome.ssl.berkeley.edu/
34. KOREA@HOME, http://www.koreaathome.org/
35. EUROGRID, http://www.eurogrid.org/

[received date : Mar. 2006, accepted date : May. 2006]