Improving the Research Environment of High Performance Computing for Non-Cluster Experts Based on Knoppix Instant Computing Technology

Fumikazu KONISHI[1], Manabu ISHII[2], Shingo OHKI[1], Yusuke HAMANO[3], Shuichi FUKUDA[2], and Akihiko KONAGAYA[1]

[1] RIKEN Genomic Science Center (GSC)
Advanced Genome Information Technology Research Group
Bioknowledge Federation Research Team
{fumikazu,ohki,konagaya}@gsc.riken.jp
[2] Tokyo Metropolitan Institute of Technology
[3] VSN Inc.

Abstract. We have designed and implemented a new portable system that can rapidly construct a computer environment where high-throughput research applications can be performed instantly. One challenge in the instant computing area is constructing a cluster system instantly, and then readily restoring it to its former state. This paper presents an approach for instant computing using Knoppix technology that can allow even a non-computer specialist to easily construct and operate a Beowulf cluster . In the present bio-research field, there is now an urgent need to address the nagging problem posed by having high-performance computers. Therefore, we were assigned the task of proposing a way to build an environment where a cluster computer system can be instantly set up. Through such research, we believe that the technology can be expected to accelerate scientific research. However, when employing this technology in bio-research, a capacity barrier exists when selecting a clustered Knoppix system for a data-driven bioinformatics application. We have approached ways to overcome said barrier by using a virtual integrated RAM-DISK to adapt to a parallel file system. To show an actual example using a reference application, we have chosen InterProScan, which is an integrated application prepared by the European Bioinformatics Institute (EBI) that utilizes many database and scan methods. InterProScan is capable of scaling workload with local computational resources, though biology researchers and even bioinformatics researchers find such extensions difficult to set up. We have achieved the purpose of allowing even researchers who are non-cluster experts to easily build a system of "Knoppix for the InterProScan4.1 High Throughput Computing Edition." The system we developed is capable of not only constructing a cluster computer environment composed of 32 computers in about ten minutes (as opposed to six hours when done manually), but also restoring the original environment by rebooting the pre-existing operating system. The goal of our instant cluster computing is to provide

an environment in which any target application can be built instantly from anywhere.

1 Introduction

Over the last decade, high performance computing has become a fundamental technology essential for large-scale scientific research. The Beowulf[1–3] parallel workstation that consists of commercial PC components achieves a balanced low-cost architecture for an environment of single-user scientific workstations. However, as Philip Papadopoulos points out, the economics of clusters have changed due to additional and ongoing personnel costs related to the "care and feed" of the machine.[4–6]

This paper presents an image-based approach for a light-load deploying system using Linux-based Live CD technology. The image-based system adapts well to temporary usage. The original environment can be easily rolled back as well. We have designed and implemented a new portable system that can rapidly construct a computer environment where high-throughput search applications for protein analysis can be performed instantly. One challenge in this instant computing area is making a target system with a reasonable configuration to enable instant construction, and then easy restoration to the former state. The advantage of instant computing is its demonstrated available technology to solve a given problem without the need for special technical knowledge in order to build a system that can perform the intended application. Consequently, end users have practical needs for instant computing.

2 Related Work

Related work in instant computing technology can be divided into two groups: install-based systems and image-based systems.

2.1 Install-Based Deploying System

NPACI Rocks toolkit The NPACI Rocks toolkit developed by the University of California at San Diego (UCSD) is designed to address a large-scale, clusterwork support infrastructure for applications that scientists can build and manage by themselves. Rocks has achieved the setup of a system consisting of hardware and software, and which is unified by prescribed system configurations. The Rocks cluster architecture inherited by the Beowulf project was defined as consisting of minimal components for which there are large mean-time-to-failure specifications. [4] The conventional Beowulf parallel workstation intended for scientific applications requiring the repetitive use of large data sets and large applications is composed of a front-end node and work-nodes with dual channel Ethernet networks. Therefore, the presumed Rocks hardware system offers both simplicity and a high degree of practicality. UCSD also developed a robust set

of OS installation tools known as NPACI Rocks with RedHat Kickstart. UCSD had deployed all software by using RPM-based automatic configuration technology, and also supported a reinstallation mechanism for forcing the base OS on the computing nodes as well. As far as possible in the Rocks world, such technologies may facilitate the easy building of cluster systems by end users. In addition, similar efforts have been made to realize a light-speed deploying system for cluster computing the Real World Computing Partnership, Scyld Beowulf, Scalable Cluster Environment, Open Cluster Group, VA Linux, and Extreme Linux. These install-based management systems are fitted into a uniform cluster system that can be used for a long time. For weekend computing, end users may want to temporarily construct a cluster computer system, and will not accept a destructive reinstallation because the PC must be restored to its original condition.

2.2 Image-Based Deploying System

Knoppix Knoppix is a collection of GNU/Linux and features one CD live file system (iso9660) that can be customized as a full-featured portable computer system. [7] The key technologies of Knoppix are automatic hardware detection and configuration, and a compressed loop-back device. The loop-back device allows us to mount a file as a block device, thus reducing the system file image and enabling the file to be read on the-fly decompression. After the CD has been mounted on the loop-back device, additional memory disks (tmpfs) are mounted with a writable ext2 file system for an application program as a normal Linux distribution system. The tmpfs size is adapted from the available size of real memory.

ClusterKnoppix ClusterKnoppix is a Linux kernel extension for single-system image clustering that adopts Knoppix distribution using the OpenMosix kernel, and is designed to activate a cluster without having to install it on the hard disk. The MOSIX multi-computer system, which is improved by kernel algorithms for sharing the scalability of PC cluster resources, has a feature of preemptive process migration for dynamic load-balancing and memory-ushering, due to the management mechanism required for cluster-wide dynamically distributed resources in a time-sharing parallel execution environment for multiple users. MOSIX offers a general-purpose environment infrastructure for executing large scale, demanding sequential and parallel applications.[8] The MOSIX infrastructure includes such MOSIX File Systems as the Global File System (GFS)[10] and Parallel Virtual File System (PVFS)[11] that provide a unified view of all files on all mounted file systems in all nodes of the MOSIX cluster. The system consists of several functional components for easily building a cluster system. For booting up the work nodes via a network, ClusterKnoppix has the OpenMosix terminal server that integrates the Pre-Boot Execution Environment (PXE), Dynamic Host Configuration Protocol (DHCP), and tftp. Moreover, the system allows new nodes to join the cluster automatically by using the auto discovery feature for improved convenience.[12]

3 Application

In the present bio-research, bioinformatics applications that typically represent a data-oriented approach are utilized with a public and/or in-house database from which a researcher can obtain new findings about topics of interest. To prepare a research environment, bioinformatics applications usually require a long time to perform, and are not easy for researchers who are not experts on information technology. Furthermore, it is very troublesome to build and maintain a system for large-scale computation. We were assigned the task of proposing a way to build an environment where a cluster computer system can be instantly set up. This technology is expected to accelerate the pace of bioinformatics research. Building such a temporary system is not very appealing in view of existing research and development systems. Therefore, there is an urgent need to address this nagging problem.

We have chosen InterProScan [13] as a reference application, which represents data-oriented characteristics. InterProScan is an integrated application prepared by the European Bioinformatics Institute (EBI) that utilizes many databases and scan methods for protein signatures. These well-maintained databases include protein families, domains, and functional sites in which identifiable features found in known proteins can be applied to unknown protein sequences. InterProScan allows a protein science researcher to simultaneously scan several member databases, such as PROSITE patterns, PROSITE profile, PRINTS, PFAM, PRODOM, SMART, and TIGRFAMs. InterProScan is capable of scaling workload with local computational resources, though biology researchers and even bioinformatics researchers find such extensions difficult to set up.

4 System Design and Implementation

We have been addressing the difficulties of instantly constructing a high-performance cluster computing system, and improving the research environment to make it easy for non-cluster experts to do so. Our approach entails two main domains in the deploying environment. First, we will decide on a target application. Secondly, our second domain is remastering a specific service. This section describes the system design and implementation for remastering typical bioinformatics applications on Live-OS.

InterProScan consist of several functional scripts: system configure, pre-procedure, job submitting, status checker, post-procedure, and member database. The database contains 11,972 entries, representing 3079 domains, 8597 families, 228 repeats, 27 active sites, 21 binding sites, and 20 post-translational modification sites. Overall, there are 7,521,179 InterPro hits from 1,466,570 UniProt protein sequences in release 10.0. [14] Thus, the database contains 5.3G-byte file sets comprising 38,391 directories and 38,433 files. InterProScan is a well-known application with abundant directories.

As for why there are so many directories, there is an issue regarding how a protein family model file of the HMMER program is stored in each directory.

Thus, the file system must store much structure information as metadata. Moreover, InterProScan will also submit 12 jobs per sequence file. Thus, storage space greater than the file size of a member database is required when adding the file size of meta-data and the results.

A Live-OS offers the advantage of easily setting up the most suitable environment, but the technical issue of creating more than 6G bytes of data storage space for member databases and results must be addressed. Specifically, a single Live-OS node without a hard disk drive cannot be expected to obtain RAM-DISK space exceeding 2G byte. Therefore, to make our proposed method complementary, a parallel file system has been chosen to integrate RAM-DISK storage with a clustered Live-OS computer. To realize our method of an integrated RAM-DISK, we have designed KnoppixCluster using a traditional architecture for high-performance computing environments such as the Beowulf parallel workstation, which has been defined for single-user multiple computers. In order to develop InterProScan service on KnoppixCluster, we have developed a series of setup scripts: htc_hop, htc_step, and htc_jump. The htc_hop script executes a setup procedure to deploy the back-end image. First, the front-end is booted from a local CD-image. Then the front-end executes htc_hop to activate the features with our configuration, which includes the network card settings, DHCP IP ranges, and client-side NIC drivers. To enable these features, we have chosen the Knoppix terminal server, which allows thin clients such as diskless workstations. [15] After htc_hop is completed, the system is ready to start a back-end-node booting sequence. The back-end nodes must support a PXE prepared by the front-end node for network booting from the terminal server .

The htc_step script can then be executed on the front-end node, provided that the necessary number of back-end-nodes are booted up, thus allowing us to automatically create a configuration file for PVFS2 and Condor [16]. Our configurations that focus on instant computing can instantly create an on-memory-parallel-file system using PVFS2 to integrate a specified memory disk (tempfs) that is mounted with a writable ext2 file system on the back-end nodes. In order to build a service environment for InterProScan4.1 with database release 10.0, additional capacity of 1.2G bytes is necessary for that data structure, although a capacity of 5G bytes should be sufficient to store a database. Thus, the quantity necessary for this structure information can be obtained through experimental observation beforehand. Therefore, this system must use PVFS2 to create a total capacity greater than 6.2G bytes. This capacity thus becomes the condition on which to maintain the minimum system configuration. This condition is evaluated based on the run-time system capacity on the back-end nodes, then the possibility of said system configuration is evaluated, and the script provides information for the end-user. The back-end node serves an important role in providing the on-memory-parallel-filesystem. The back-end node also functions as a work node to perform a given task at the same time. In order to utilize the back-end nodes, the condor scheduling system allows us to deal with parallel jobs involving large-volume processing. The condor is set up by running a condor setup program (condor_configure) on each back-end node in

the on-memory-parallel-filesystem through serial processing. In order to collect the software and hardware of "headth" on all nodes, we have chosen Ganglia, which is a lightweight, distributed, multicast-based monitoring system. Ganglia allows us to indicate the number and speed of the CPUs, the kernel version, the amount of RAM installed, and more useful information about all nodes. After all setups are completed, a test job is performed in the htc_jump script to confirm whether all setups can be properly executed, and with the results being verified.

5 Performance

5.1 The evaluation of application performance by the boot memory model

We have evaluated the effects of dividing main memory in the Knoppix Instant Computing System on an application because rewritable space is important for practical use. This problem dictates how much main memory size can be assigned to RAM-DISK to execute typical bioinformatics applications on KNOPPIX.

For example, a computer with 1G byte of memory is able to assigned 400M bytes of memory for operating system use; thus, the remaining 600M bytes of memory can be assigned for RAM-DISK use. When consisting of ten cluster nodes, this system can create a rewritable capacity of 6G bytes . The capacity that can be used for the on-memory-parallel-file system is reduced when too much quantity is allocated to the operating system. The system thus assigns the necessary and sufficient conditions of system memory for a bioinformatics application.

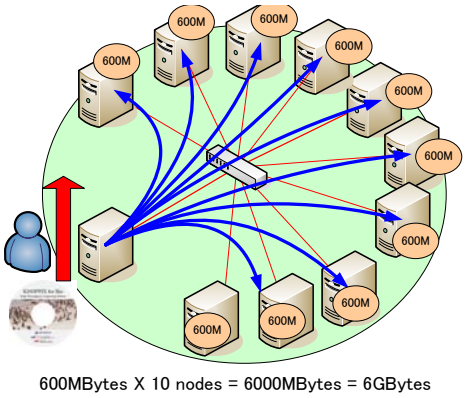

Fig. 1. A boot model for Knoppix Cluster

Table 1 shows the test equipment that we used to build Knoppix Cluster for this performance evaluation. The front-end node has 4G bytes of memory

to ensure the stability of system operation, and the back-end nodes have 2G bytes of memory. Table 2 shows the experimental conditions for the different boot memory models. As for the experiment, we measured the execution time of parallel BLAST extended from NCBI BLAST [18] with a combination of query and database size on the on-memory-parallel-filesystem in both High mode and Low mode. The experiment was repeated five times to evaluate repeatedly, and a total trial experiment count of 1350 times was enforced.

Figure 2 shows the homology search throughput, which is the capability of each memory model in units of time. There were no differences between the two memory models based on the results of the wide-range parameter sweep experiment. Therefore, it was shown that similar performance could be expected for a special configuration of the on-memory-parallel-filesystem.

Table 1. Test Equipment

	Front-end node	Back-end node
CPU	Pentium4 2.4 GHz	Pentium4 2 GHz
Main Memory	4G bytes	2G bytes
NIC	1000 Base-T	100 Base-T
node	1	10

Table 2. Experiment Conditions for boot memory models

item	parameter
Memory Model	high (800 MB) low (400 MB)
File System	PVFS ver.1
application	Parallel Blast
Database Size (sequences)	1000,10000,10000
Query (sequences)	1,2,4,8,10,100
Number of Proc. (CPU)	1,2,4,8,10
Rep.	5

5.2 The evaluation of instant system setup performance

Knoppix-Cluster allows us to instantly reproduce a final system configuration by embedding a construction script in a boot image. An end user without special expertise can use this mechanism to build a cluster system. Table 3 shows the setup time for each step in our instant Knoppix Cluster. It took about ten minutes (on a 30-node scale) to build a cluster computer on the instant system. Even when compared with the reinstallation time stated in the reference paper about ROCKS [4], performance equivalent to that above is realized for this setup time.

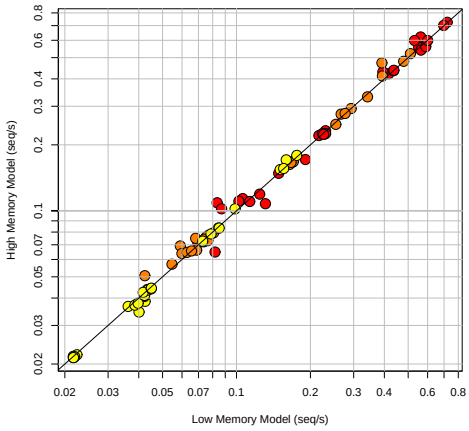

Fig. 2. Homology Search Throughput based on differences in boot memory proportion

Table 3. The setup time for an instant cluster

Script		Time (sec)		
Work nodes size		**10**	**20**	**30**
HTC_hop		*22*	*21*	*19*
HTC_Step		*214*	*419*	*619*
File System	(PVFS2)	46	82	118
Scheuler	(Condor)	156	299	445
Monitor	(Ganglia)	12	21	32
Total Time		**234**	**440**	**638**

5.3 The evaluation of InterProScan4.1 performance for instant computing

An experiment was conducted to verify practical use of the high-throughput application environment instantly provided for non-cluster experts. InterProScan4.1 is a well-known integrated application that can perform a search using 12 programs and a database. Each application has a different executive time distribution. Figure 3 shows the total execution time difference between RAM-DISK and Hard DISK. Since Knoppix Cluster is built by using the on-memory-parallel-filesystem (which is a case of special use),we had to observe the difference in practical execution time.

6 Conclusion

We have presented and implemented a new approach to image-based instant computing technology on Knoppix Cluster for improving the research environ-

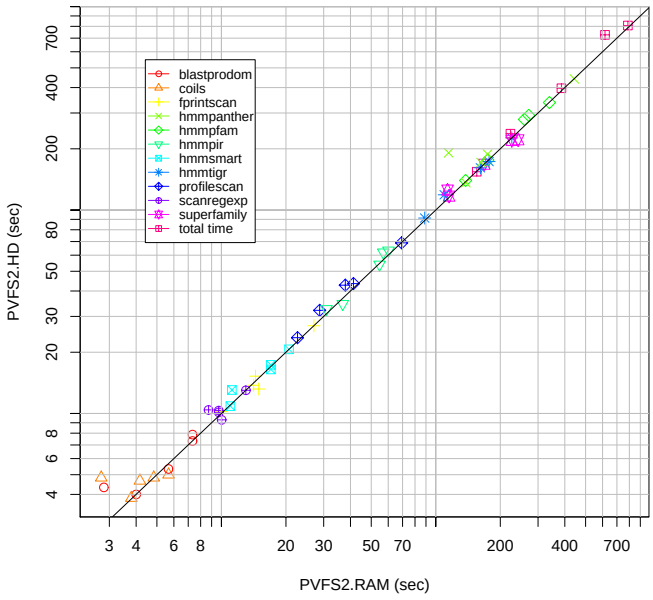

Fig. 3. The execution time of InterProScan versus RAM-DISK (RD) and Hard Disk (HD)

ment of high performance computing for non-cluster experts. Our work represents the first step in exploring design and implementation issues regarding instant computing technology. We have been very particular about restoring the original condition as held before. We expanded instant computing by using the on-memory-parallel-filesystem for image-based technology from install-based technology, and this technology enabled us to handily build a cluster system. Consequently, we considered its adaptation to practical bioinformatics applications and succeeded in building an InterProScan4.1 environment and distributing images for Knoppix using the InterProScan4.1 High Throughput Computing Edition. [19] The results can then be used as one infrastructure for deploying application service.

Acknowledgments

The authors would like to thank Kuniyasu Suzaki for his valuable contributions regarding Knoppix.

References

1. Donald J. Becker, Thomas Sterling, Daniel Savarese, John E. Dorband, Udaya A. Ranawak, Charles V. Packer, "BEOWULF: A PARALLEL WORKSTATION FOR

SCIENTIFIC COMPUTATION", Proceedings, International Conference on Parallel Processing, (1995)
2. Thomas Sterling, Daniel Savarese, Donald J. Becker, Bruce Fryxell, Kevin Olson, "Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation", Proceedings,High Performance and Distributed Computing, (1995) 23–30
3. Thomas Sterling, Donald J. Becker, Daniel Savarese, Michael R. Berry, and Chance Res. ,"Achieving a Balanced Low-Cost Architecture for Mass Storage Management through Multiple Fast Ethernet Channels on the Beowulf Parallel Workstation", Proceedings, International Parallel Processing Symposium, (1996) 104–108
4. Philip M. Papadopoulos, Mason J. Katz, and Greg Bruno, "NPACI Rocks: Tools and Techniques for Easily Deploying Manageable Linux Clusters", Cluster 2001: IEEE International Conference on Cluster Computing, Oct. (2001)
5. Mason J. Katz, Philip M. Papadopoulos, and Greg Bruno, "Leveraging Standard Core Technologies to Programmatically Build Linux Cluster Appliances", Cluster 2002: IEEE International Conference on Cluster Computing, Apr. (2002)
6. Philip M. Papadopoulos, Caroline A. Papadopoulos, Mason J. Katz, William J. Link, and Greg Bruno, "Configuring Large High-Performance Clusters at Lightspeed: A Case Study", Clusters and Computational Grids for Scientific Computing 2002, Dec (2002)
7. Klaus Knopper, "Building a self-contained autoconfigurarion Linux system on an iso9660 file system", 4th Annual Linux Showcase & Conference Atlanta, (2000)
8. Barak A. and La'adan O., "The MOSIX Multicomputer Operating System for High Performance Cluster Computing", Journal of Future Generation Computer Systems (13) 4-5, pp. 361-372, March (1998)
9. Amar L., Barak A. and Shiloh A., "The MOSIX Parallel I/O System for Scalable I/O Performance", Proc. 14-th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2002), pp. 495-500, Cambridge, MA, Nov. (2002)
10. Global File System, "http://www.redhat.com/en_us/USA/home/solutions/gfs/"
11. P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur, "PVFS: A Parallel File System For Linux Clusters", Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA, pp. 317-327, Oct (2000)
12. ClusterKnoppix, "http://bofh.be/clusterknoppix/"
13. Evgeni M. Zdobnov and Rolf Apweiler, "InterProScan–an integration platform for the signature-recognition methods in InterPro. Bioinformatics", 17(9):847-8, Sep (2001)
14. Mulder NJ, et al. , "InterPro, progress and status in 2005", Nucleic Acids Res. 33, Database Issue:D201-5 (2005)
15. Linux Terminal Server Project, http://www.ltsp.org/
16. Michael Litzkow, Miron Livny, and Matt Mutka, "Condor - A Hunter of Idle Workstations", Proceedings of the 8th International Conference of Distributed Computing Systems, pages 104-111, June, (1988)
17. Todd Tannenbaum, Derek Wright, Karen Miller, and Miron Livny, "Condor - A Distributed Job Scheduler", in Thomas Sterling, editor, Beowulf Cluster Computing with Linux, The MIT Press, (2002)
18. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. ,"Basic local alignment search tool." J. Mol. Biol. 215:403-410 (1990)
19. Knoppix for InterProScan4.1 High Throughput Computing Editon http://big.gsc.riken.jp/index_html/Members/fumikazu/htc/