Original Paper

A Distributed-Processing System for Accelerating Biological Research Using Data-Staging

Yoshiyuki Kido,[†1] Shigeto Seno,[†1] Susumu Date,[†1]
Yoichi Takenaka[†1] and Hideo Matsuda[†1]

The number of biological databases has been increasing rapidly as a result of progress in biotechnology. As the amount and heterogeneity of biological data increase, it becomes more difficult to manage the data in a few centralized databases. Moreover, the number of sites storing these databases is getting larger, and the geographic distribution of these databases has become wider. In addition, biological research tends to require a large amount of computational resources, i.e., a large number of computing nodes. As such, the computational demand has been increasing with the rapid progress of biological research. Thus, the development of methods that enable computing nodes to use such widely-distributed database sites effectively is desired. In this paper, we propose a method for providing data from the database sites to computing nodes. Since it is difficult to decide which program runs on a node and which data are requested as their inputs in advance, we have introduced the notion of "data-staging" in the proposed method. Data-staging dynamically searches for the input data from the database sites and transfers the input data to the node where the program runs. We have developed a prototype system with data-staging using grid middleware. The effectiveness of the prototype system is demonstrated by measurement of the execution time of similarity search of several-hundred gene sequences against 527 prokaryotic genome data.

1. Introduction

Biological science has been rapidly expanding as a result of advances in information science and technology. Bioinformatics has solved biological science problems using information technology and has contributed to bio-related sciences, such as medical science, pharmaceutical science, chemical biology, and agricultural biology. In addition, bio-related databases are generally used in bioinformatics studies. For example, in drug discovery, docking simulations between drug-candidate compounds and target proteins use many bio-related databases, such as databases having protein structures and functions, compound structures, and drug data. These data are mainly stored in databases and are publicly available through the Internet. GenBank [1], EMBL [2] and DDBJ [3] are three such genome and gene sequence databases. Several other biological databases also contain specific biological data. For example, PDB [4] contains protein structure information, ChEBI [5] and PubChem [6] contain compound structures, OMIM [7] contains disease information, Gene Expression Omnibus (GEO) [8] contains gene expression profiles, KEGG [9] contains metabolic pathways, and PubMed [10] and MEDLINE [11] contain publication data in the area of life science.

Usage of bioinformatics tools using databases has been complicated by the fact that the number of databases and classes of databases has increased rapidly due to the progress of biotechnology. To tackle this problem, major database sites have developed keyword-based database-integration. Entrez [12] has been developed to search entire databases, such as genome and protein sequences, structures and their publication data via the web interface. Keyword-based integration is very powerful, but is insufficient for analysis with customizable parameters, such as sequence similarity search. Researchers often need to analyze the data on these databases with specific conditional parameters. Moreover, they often need to combine the public data and their own data, and often need to use the tool they developed. However, most database sites provide services not for analyses customizable by individual researchers but for general ones.

Web services also have been provided by major bio-related database sites. It is one of the network technologies that offer application program interface (API) as a method to access remote hosts. Researchers are able to analyze with customizable parameters via web services. However, the web services only handle a pre-

†1 Graduate School of Information Science and Technology, Osaka University

defined set of the databases and cannot handle their subset demanded by individual researches.

A possible solution to satisfy both customizable conditional parameters and desirable subset for individual researches is the mirroring of the whole databases from the database sites to individual users' environment. Mirroring is one of the distributed computing techniques that transfers database files to computational resources. However this approach also has a serious problem that is incremental update of the bio-related databases having increasing growth. Since the amount of bio-related data has been increasing rapidly now on, mirrored hosts require unrealistic storage volume (i.e., as much as or more than the amount of a database site requires).

To cope with the problem, we propose a distributed-processing system using data-staging technique. Data-staging is a distributed computing technology that provides a method of automatically transferring files before executing a job. Data-staging is technology to forward necessary files to a computing node when the files are needed, which allows a user to forgo file management. Data-staging does not cause a problem with respect to updating of the database because the latest database is forwarded directly from the database site.

In this paper, we have developed a prototype system with data-staging to handle a large number of databases, which allows researchers to access data of interest and to benefit from the computational resources of distributed technology. This paper is organized as follows. Section 2 presents background along with an example of protein-protein interaction network research, which requires heavy computational power and a significant amount of input data. We then describe the technical issues involved in the use of bioinformatics tools. Section 3 describes the construction of the prototype system with data-staging for a large number of databases. In Section 4, future issues and research directions to achieve the goal of the present research are discussed. Finally, Section 5 concludes our paper.

2. Background and Technical Issues

In this section, we first describe the background and motivation of our study. Then we describe the important technical issues that form the basis of the execution environment for bioinformatics tools with handling of data files.

2.1 Background and Motivation

Researchers need to use various bio-related databases. For example, some researchers investigate whether gene functions in certain organisms are also observed in other organisms. In the protein-protein interaction network analysis of *Escherichia coli* [13], a large number of BLAST [14] searches of 148 complete genomes were conducted using the sequences of the E. coli proteins corresponding to highly-connected nodes (hubs) in the network. The results of this analysis revealed that the hub proteins are highly conserved in more than 125 genomes and that the connectivity of proteins in the network is positively correlated with the number of genomes. Another example is that orthologue and paralogoue identification often need to conduct genome-by-genome sequence comparisons for the bidirectional best-hit analysis [15]. Thus, a general framework for a large number of analyses using several databases is needed.

The current services provided by genome database sites make such large-scale analyses difficult. For example, if BLAST is executed on a mixed database that includes several genome sequences, the results of the search may omit low-score sequences due to the cutoff setting. In general, such analyses must be conducted on individual sequences because the analysis may include a gene-finding process and the parameters of sequence searches must be selected genome by genome.

Thus, researchers have greedy requirement that is brought to realization of a user-customizable and high performance analysis environment. Distributed computing is brought to realization of high performance environment for bioinformatics. However, it still has the issue of data placement. Our study is an attempt to construct a high-efficiency bioinformatics execution environment using a distributed processing for the above-described example.

2.2 Web Service

Bio-related database sites, such as NCBI, DDBJ, and KEGG, provide analysis services and databases via the Internet. These sites have developed analysis interfaces that can be accessed by remote computers via web services [16]. The notion of web services is a standardized way of integrating web-based applications using Web Service Description Language (WSDL). A user can access to a web service site using the protocol and parameters within WSDL. Bio-related database analysis via web

services contributes to bioscience by offering an analysis environment and the latest bio-related databases. However, the web services only handle a pre-defined set of the databases and cannot handle their subset demanded by individual researches. It means that web services cannot achieve the requirement by combined subset database by each of researchers. Therefore, researchers who want to perform customizable analyses must build an analysis environment without depending on the web services alone.

2.3 Mirroring

Mirroring is a technology that creates a backup file at a distributed location. Mirroring synchronizes master files and their backup files. One typical implementation is rsync [17], which provides a synchronized file and transfers only incremental update over the Internet. For example, the Bio-mirror project [18] aims to avoid excessive loads of file transfers from database sites. However this approach also has a serious problem of increasing growth of incremental update of the bio-related databases. Since the amount of bio-related databases has been increasing rapidly, mirrored hosts need to have very large storage comparable to the database sites.

2.4 Data-Staging

As described in the previous sections, data-intensive analysis for bioscience has been unachievable based on limitations related to the use of web service or mirroring. Web service or mirroring cannot provide update file management with user-customizable analysis environments. On the other hand, data-staging is a data-management technique in the field of distributed computing that enables both update file management and user-customizable analysis environments [19]. Data-staging usually transfers files related to the target job automatically before and after the job is executed. A transfer before a job is executed is called stage-in, and a transfer after a job is executed is called stage-out. Stage-in transfers necessary files from other nodes to a job submitted to a computing node. Stage-out transfers any files (i.e., result files) from a computing node to a user client or other nodes. Those files are then removed from the computing node when the job is finished. Thus, the data-staging technique enables the researcher to access the latest data files. Data-staging is most suitable for data-intensive analysis using a large number of bio-related databases. Therefore, we attempt to build a data-intensive analysis environment for bioinformatics using recent distributed computing technology.

3. Design and Implementation

We have developed a prototype system based on distributed computing with data-staging in a grid environment for data-intensive bioinformatics tools. Grid computing technology [20] is used for the management of computational and storage resources, which has become a standard architecture for distributed computing on wide-area networks. The Globus Toolkit [21] is becoming a de-facto standard implementation for grid computing. Various grid middleware and several grid components are available for grid solutions based on the Globus Toolkit that address the requirements of new grid projects [22]–[24]. The Globus Toolkit provides fundamental services and component libraries, such as a security infrastructure (GSI), a secure login shell (GSI–SSH), and a file transfer protocol (GridFTP) [25].

Figure 1 is an overview of the software configuration of the prototype system. We have installed Globus Toolkit 4.0.5 with GSI–SSH and GridFTP on all nodes of the prototype system. The prototype system has three important nodes: the scheduler node, computing nodes, and the database node. **Figure 2** shows how these work in the prototype system. In Fig. 2, the shaded regions indicate the components we have implemented. The term "job" indicates an executable command-line tool. The job scheduler chooses one computer to execute each requested job from the computing node.

The proposed system uses the request description file for describing job information. Our system regards each attribute in the request description file as parameters to control the behavior of one or more tools. When the user writes details of job set in this file and requests job set by submitting the file to the system, the job scheduler receives and parses the file. Then, target database files are transferred to the available computing node by stage-in. **Figure 3** shows an example of a request description file for executing a BLAST program against a bacterial genome sequence (*Helicobacter pylori* 26695).

The steps of the job execution are as follows (see Fig. 2): (1) A user requests a set of jobs (hereafter, job set) using a command-line tool with a request description file with the tar-

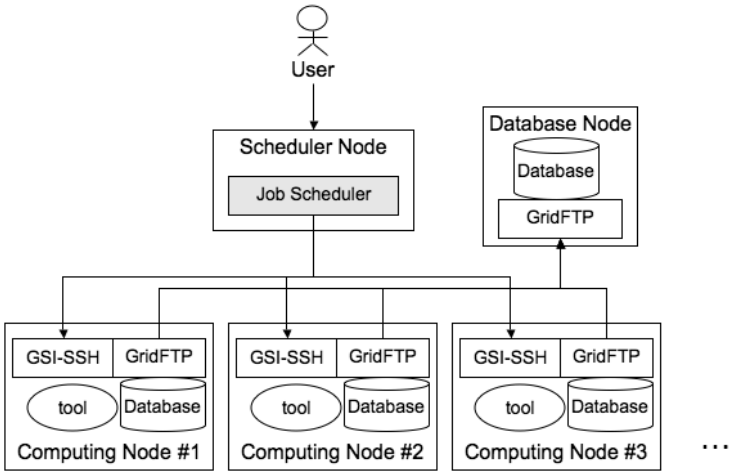

Fig. 1 Middleware composition.

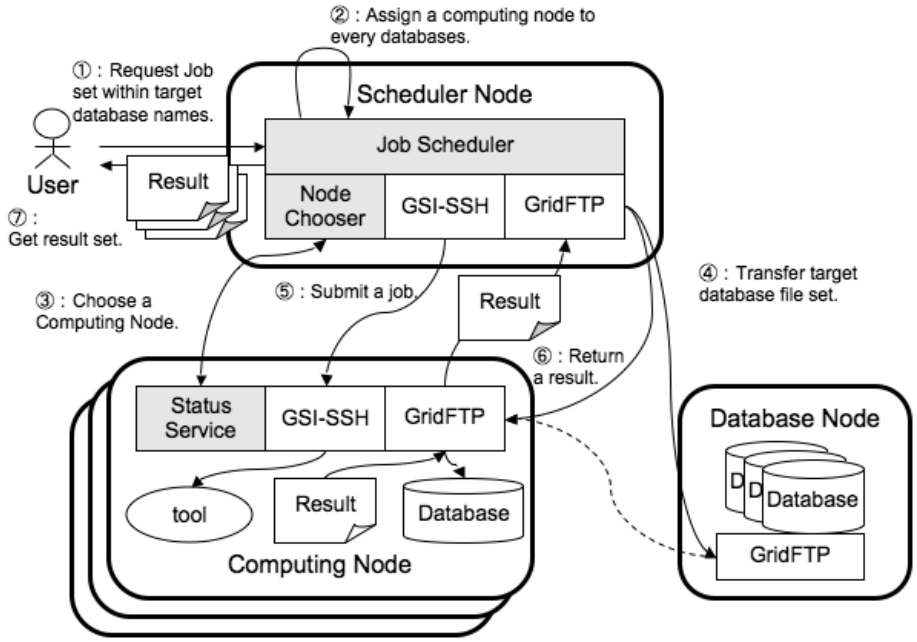

Fig. 2 Details of the system architecture.

get database names. A user can write the job set as the details of several jobs in the request description file. (2) The job scheduler assigns a computing node to each database by requesting the node chooser for choosing a computing node and an input data file for the requested job. (3) The node chooser checks which computer is available or not via the value of status service. It returns the current status of the computing node e.g., the load average and the usable storage volume. (4) If the computing node does not have database files, the job scheduler requests to stage-in the files via GridFTP. (5) The node executes the requested job. (6) The job scheduler requests to stage-out result files from the computing node. (7) Finally, when every job of the requested job set is finished, the job scheduler notifies the user of that the job set is finished.

4. Evaluation and Discussion

We developed a prototype system in the local area environment shown in **Fig. 4**. As an example of biological analysis, we performed the

```
<jobset>
<job>
<executable>/opt/blast-2.2.16/bin/blastall</executable>
 <directory>{$GLOBUS_USER_HOME}</directory>
 <database>Helicobacter_pylori_26695</database>
 <inputfile>/opt/blastdb/test1.aa.txt</inputfile>
 <argument>-p</argument>
 <argument>blastp</argument>
</executable>
</job>
<job> ··· </job>
···
</jobset>
```

Fig. 3 Request description file.

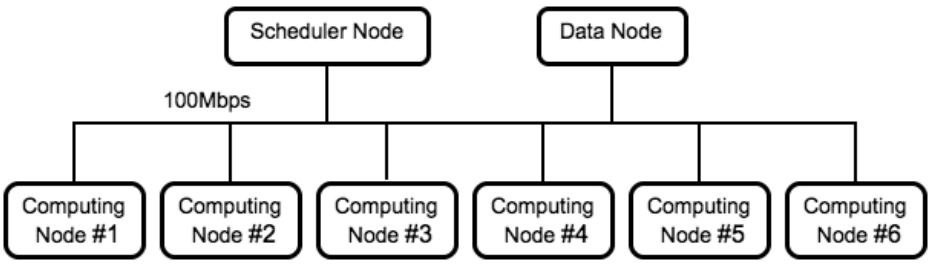

Fig. 4 Experimental environment.

Table 1 Execution time for all submitted jobs.

	Execution Time
Six nodes with the prototype system	50 m49.7 s
Single node without the prototype system	3 h25 m58.0 s

BLAST analysis described in Section 2.1 using a large amount of genome data (527 prokaryotic genomes) on six computing nodes. One node consists of a Pentium 4 (2.4 GHz) CPU, 4 GB of memory, and a 480–GB disk; and each of the other five nodes consists of a Xeon (2.0 GHz) CPU, 4 GB of memory, and a 1–TB disk. Red Hat Linux is installed on these six nodes. As the query of the search, we used a multi–FASTA format sequence data of 897 *E. coli* gene sequences, as described in Ref. 13). We also conducted the same searches on a single node without the proposed distributed-processing system and measured the resulting execution time.

Table 1 shows the results. The execution time using six nodes does not show a six-fold performance compared to the single node result. This is due to the overhead of submitting 527 jobs via GSI–SSH. Since GSI–SSH simply submits a job to a computing node, the overhead caused by distributed computing was within the allowable range. In addition, the prototype system is shown to enable researchers to submit several jobs to any computer node using data-staging on the grid environment.

However, there are two implementation issues to be addressed. First, the prototype system has a problem in indicating the data set demanded by individual users. It is difficult to identify such data set and to describe its location and file name in the request description file. We consider that a grid-wide name service is necessary for solving this problem. Secondly, for re-using files, it is necessary to check whether or not the files are updated at the database sites where the files are originally located. Since the number of files in bio-related databases is very large, it is not easy to identify which file is to be updated. Although every file in bio-related databases eventually needs to be checked for their updates, naive checks for the update of every file increase network traffic. We

consider that there is a trade-off between the size of the files to be transferred and the frequency to check their updates. This problem should be solved by a scheduler with consideration by calculating a total cost and an overhead of check of updates.

5. Conclusion

This paper describes a distributed-processing system for bioinformatics tools using data-staging and grid computing. The system is particularly useful since it enables researchers to provide customizable conditional parameters for the execution of the tools and to use specific data set they demanded.

Acknowledgments This work was supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) through the Science Grid NAREGI project, and by Grant-in-Aid for Scientific Research on Priority Areas "Information Explosion" from MEXT.

References

1) Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. and Wheeler, D.L.: GenBank, *Nucleic Acids Research*, Vol.35, Database Issue, pp.D21–D25, (2007).
2) Cochrane, G., Aldebert, P., Althorpe, N., Andersson, M., Baker, W., Baldwin, A., Bates, K., Bhattacharyya, S., Browne, P., Van den-Broek, A., et al.: EMBL Nucleotide Sequence Database: developments in 2005, *Nucleic Acids Research*, Vol.34, pp.D10–D15, (2006).
3) Okubo, K., Sugawara, H., Gojobori, T. and Tateno, Y.: DDBJ in preparation for overview of research activities behind data submissions, *Nucleic Acids Research*, Vol.34, pp.D6–D9, (2006).
4) Deshpande, N., Addess, K.J., Bluhm, W.F., Merino-Ott, J.C., Townsend-Merino, W., Zhang, Q., Knezevich, C., Xie, L., Chen, L., Feng, Z., et al.: The RCSB Protein Data Bank: A redesigned query system and relational database based on the mmCIF schema, *Nucleic Acids Research*, Vol.33, pp.D233–D237, (2005).
5) Brooksbank, C., Cameron, G. and Thornton, J.: The European Bioinformatics Institute's data resources: towards systems biology, *Nucleic Acids Research*, Vol.33, pp.D46–D53, (2005).
6) PubChem, http://pubchem.ncbi.nlm.nih.gov/
7) McKusick,V.A.: Mendelian Inheritance in Man, *Catalogs of Human Genes and Genetic Disorders, 12th edn. The Johns Hopkins University Press*, Baltimore, MD, (1998).
8) Edgar, R., Domrachev, M. and Lash, A.E.: Gene Expression Omnibus:NCBI gene expression and hybridization array data repository, *Nucleic Acids Research*, Vol.30, No.1, pp.207–210, (2002).
9) Kanehisa, M. and Goto, S.: KEGG:Kyoto Encyclopedia of Genes and Genomes, *Nucleic Acids Research*, Vol.28, No.1, pp.27–30, (2000).
10) PubMed Central http://www.pubmedcentral.gov/
11) MEDLINE http://www.nlm.nih.gov/pubs/factsheets/medline.html
12) Schuler, G.D., Epstein, J.A., Ohkawa, H. and Kans,J.A.: Entrez: molecular biology database and retrieval system, *Methods Enzymol.*, Vol.266, pp.141–162, (1996).
13) Butland, G., et al.: Interaction network containing conserved and essential protein complexes in Escherichia coli, *Nature*, Vol.433, pp.531–537, (2005).
14) Altschul, S., Gish, W., Miller, W., Myers, E. W., and Lipman D.: A basic local alignment search tool, *Molecular Biology*, Vol.215, pp.403–410, (1990).
15) Tatusov, R.L., Mushegian, A.R., Bork, P., Brown, N.P., Hayes, W.S., Borodovsky, M., Rudd, K.E. and Koonin, E.V.: Metabolism and Evolution of Haemophilus influenzae Deduced from a Whole-Genome Comparison with Escherichia coli, *Current Biology*, Vol.6, No.3, pp.279–291, (1996).
16) Papazoglou, M.P.: Service-oriented computing: concepts, characteristics and directions, *Web Information Systems Engineering 2003 Proceedings of the Fourth International Conference on*, pp.3–12, (2003).
17) Tridgell, A. and Macherras, P.: The rsync algorithm, *Technical Report TR-CS-96-05*, Canberra 0200 ACT, (1996). http://samba.anu.edu.au/rsync/
18) Gilbert, D., Ugawa, Y., Buchhorn, M., Wee, T.T., Mizushima, A., Kim, H., Chon, K., Weon, S., Ma, J., Ichiyanagi, Y., Liou, D.M., Keretho, S. and Napis, S.: Bio-mirror project for public bio-data distribution, *Bioinformatics*, Vol.20, pp.2338–2340, (2004).
19) Hatanaka, M., Nakano, Y., Iguchi, Y., Ohno, T., Saga, K., Akioka, S. and Matsuoka, S.: Design and Implementation of NAREGI Super-Scheduler based on OGSA Architecture, *IPSJ SIG Notes*, Vol.2005, No.57, pp.33–38, (2005).
20) Foster, I., Kesselman, C., and Tuecke, S.: The Anatomy of the Grid, *International Journal of Supercomputer Applications* (2001)
21) Foster, I. and Kesselman, C.: Globus: A

Metacomputing Infrastructure Toolkit, *International Jounal of Supercomputer Applications*, Vol.11, No.2, pp.115–128, (1997).
22) Wolski, R., Brevik, J., Obertelli, G., Spring, N. and Su, A.: Writing programs that run EveryWare on the Computational Grid, *Parallel and Distributed Systems, IEEE Transactions*, Vol.12, Issue. 10, pp.1066–1080, (2001).
23) Shimojo, S., Sekiguchi, S., Miura, K. and Matsuoka, S.: Current Status of Grid Computing Projects in Japan (Special Issue on Large-scale Computer Simulation), *Institute of System, Control and Information Engineers*, Vol.48, No.7, pp.244–249, (2004), http://www.naregi.org
24) The Enabling Grids for E-Science. http://www.eu-egee.org
25) Allcock, W.: GridFTP protocol specification, *Global Grid Forum 20, GridFTP Workshop Document* (2003).

(Received October 16, 2007)
(Accepted November 27, 2007)

(Communicated by *Tatsuya Akutsu*)

Yoshiyuki Kido received his B.E. degree from Osaka Sangyo University in 1999. He had engaged in the designing and development of enterprise mission-critical system at Liberty System, Ltd. from 1999 to 2002. After that, he joined Mitsui Knowledge Industry Ltd. in 2002 and has been working for the research and development for Grid systems since then. Also, he is a Ph.D. student at the Graduate School of Information Science and Technology, Osaka University from 2005. His concern of research is Grid portal and Data Grid middleware. He is a member of IEEE CS and IPSJ.

Shigeto Seno is an Assistant Professor of the Graduate School of Information Science and Technology, Osaka University. He received his B.E., M.E. and Ph.D. degrees from Osaka University in 2001, 2003 and 2006 respectively. He is a member of IEEE and IPSJ.

Susumu Date is an Associate Professor of the Graduate School of Information Science and Technology, Osaka University. He received his B.E., M.E., and Ph.D. from Osaka University in 1997, 2000, and 2002, respectively. He was Assistant Professor at the Graduate School of Information Science and Technology, Osaka University from 2002 to 2005. He was Assistant Professor at the Graduate School of Information Science and Technology, Osaka University from 2002 to 2005. He also had worked as a visiting scholar in University of California, San Diego in 2005. He is currenty working as a Specially-appointed Associate Professor for the internationalization of education in the Graduate School of Information Science and Technology, Osaka University through the MEXT-funded educational program. His research field is computer science and his current research interests include application of Grid computing and related information technologies. He is a member of IEEE CS and IPSJ.

Yoichi Takenaka received the M.E. and Ph.D. in 1997, and 2000 from Osaka University, respectively. He worked for Osaka Universitiy from 2000 to 2002 as assistant professor, and now he is associate professor at Graduate School of Information Science and Technology, Osaka University. His research intersts include Bioinformatics, DNA computing, and Neural Networks.

Hideo Matsuda is Professor of the Department of Bioinformatic Engineering, Graduate School of Information Science and Technology, Osaka University. He received his B.S., M.Eng., and Ph.D. degrees from Kobe University in 1982, 1984 and 1987, respectively. His research interests include computational analysis of genomic sequences, integrated biological databases, and data grid technology. He is member of JSBi, ISCB, IEEE CS and ACM.