A Model of Problem Solving Environment for Integrated Bioinformatics Solution on Grid by Using Condor

Choong-Hyun Sun[1], Byoung-Jin Kim[1], Gwan-Su Yi[1], and Hyoungwoo Park[2]

[1] School of Engineering, Information and Communications University, 103-6
Munji-Dong, Yusung-Gu, Daejon 305-714, Korea
{chsun,bjkim,gsyi}@icu.ac.kr
http://rosetteer.icu.ac.kr

[2] Grid Technology Research Department, Korea Institute of Science and Technology
Information, Yusung, P.O. Box 122, Daejon 305-600, Korea
hwpark@kisti.re.kr

Abstract. To solve the real-world bioinformatics problems on grid, the integration of various analysis tools is necessary in addition to the implementation of basic tools. Workflow based problem solving environment on grid can be the efficient solution for this type of software development. Here we propose a model of simple problem solving environment that enables component based workflow design of integrated bioinformatics applications on Grid environment by using Condor functionalities.

1 Introduction

Bioinformatics field meets inevitable need of high-throughput computing resources as the size of biological data to be managed and analyzed is increasing with the progress of high-throughput biotechnology. Grid computation matches well with this need but it has been applied only to several compute-intensive bioinformatics tools. In addition, there are few examples of integrated bioinformatics solution using grid environment. Many biological analyses need the flexible and diverse integration of basic bioinformatics tools. One of the efficient solutions for this type of software development may be the workflow-based problem solving environment (PSE). There are still heavy overhead, however, to develop and implement this workflow-based model on current grid system or grid middleware. Concerning about this fact, the grid supporting technology in Condor [1] has lots of merit.

In this report, we especially take advantage of simple workflow design functionality of Condor by using a meta-scheduler, DAGMan [1] with many other Condor features for high-throughput computing application on grid. We realized two examples of integrated bioinformatics solutions about sequence searching and orthologous gene finding (OGF) based on our model of workflow based PSE on grid.

H. Jin, Y. Pan, N. Xiao, and J. Sun (Eds.): GCC 2004, LNCS 3251, pp. 935–938, 2004.

2 Model of Integrated Bioinformatics Solution on Grid

We propose a model of workflow-based bioinformatics PSE by using condor functionalities and show practical examples of integrated bioinformatics solutions. Fig. 1 (a) shows the structure of this model. It has basically two layers: components layer and integrated applications layer. The components layer consists of application component and interfacing component. Both of them are grid-enabled executable files. Application component is a typical bioinformatics tool which can perform the separable process of bioinformatics analysis. Interfacing components are all intervening programs necessary to combine application components and related data of the workflow for an integrated solution. The components can be reused or added for new integrated solution. Various workflows matching to specific solutions can be constructed by using simple Condor scripts with proper arrangement of the old and new components. In this way, the system can achieve the maximum flexibility and efficiency for the solution development corresponding to the diverse integrating requests of bioinformatics problems.

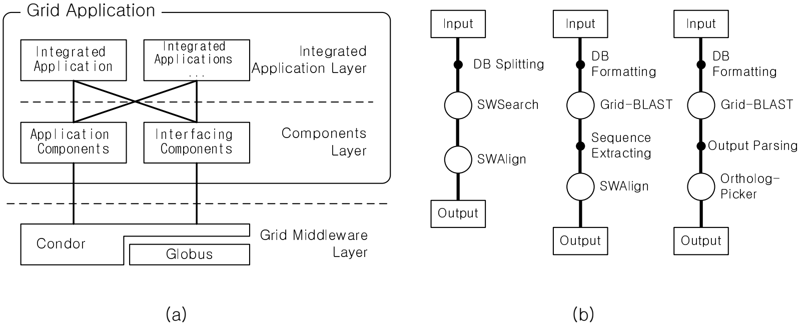

Fig. 1. (a)Structure of bioinformatics problem solving environment on Grid. (b)Examples of workflows for the integrated bioinformatics solutions. Application components and interfacing components are represented by white circles and black dots, respectively

2.1 Components Layer

We selected and implemented sequence search and alignment tools as application components because of their popularity in biological sequence analysis and the need for our integrated solutions. Among them, Grid-BLAST, Grid-FASTA, Grid-SWSearch, Grid-SWAlign, and Ortholog-Picker are briefly introduced here.

Grid-BLAST and Grid-FASTA grid-enabled version of BLAST [2] and FASTA [3]. These tools facilitate searching homologous sequences in multiple sequence databases for various numbers of querying sequences with grid resources. They generate a condor command script file to run the parallel job and submit it into condor pool. The scripts can have the options regarding proper usage of computational resources as well as the original options of BLAST and FASTA.

Grid-SWSearch is a homologous sequence searching program based on Smith-Waterman algorithm [4] that gives more precise results than BLAST or FASTA. In Grid-SWSearch, database and query sequences are divided and assigned to available computing nodes for the parallelization of computing jobs. The computing job is designed not to pass any massage to the other jobs to remove communication overhead and to run each job independently. After bundles of jobs are submitted to condor pool, each job calculates the similarity of all pair of sequences, performs statistical evaluation, and sends back the result. Grid-SWSearch does not show alignment to reduce execution time. We can execute Grid-SWAlign to see sequence alignment.

Grid-SWAlign is a grid enabled sequence alignment program based on Smith-Waterman algorithm. It receives multiple sequences to be aligned and makes a bundle of jobs on grid for all combinations of pair-wise alignment. We could improve the efficiency of both Smith-Waterman algorithm tools with the benefit of grid system.

Ortholog-Picker is an application to pick out orthologous genes from various genome sequence databases. Once Grid-BLAST searches all the gene sequences of query genome for the best hit of all the genes in the other genomes, the corresponding gene IDs are parsed from the BLAST output. In our definition, an ortholog is the group of three or more genes in which all genes are the best hit against the genomes of each other reciprocally.

Interfacing component is the program that process input or output data of application component to facilitate diverse combinations of application components that have their idiopathic input/output format. The examples of the functions of interfacing components are splitting, merging, converting, and formatting files , or extracting and rearranging the content of data.

2.2 Integrated Application Layer

Fig. 1 (b) shows three examples of the workflow for integrated bioinformatics solution. As shown in the left picture of Fig. 1 (b), the searching tool and alignment tool can be linearly arranged as a workflow described in a DAG script file. This work starts by splitting database into fragments. Multiple jobs of SWSearch are submitted and allocated at distributed computers by Condor. Output files are accumulated in the submission computer's directory. Finally, a bundle of jobs for SWAlign are submitted to show aligned sequences. One can easily change the components of this workflow by using DAG script file. For example, as shown in the middle picture of Fig. 1 (b), one may want to choose Grid-BLAST first for whole genome sequence comparison for quick search, then select the sequences of interest and run SWAlign to find more sensitive local alignments that could be missed by Grid-BLAST alone. The third workflow is the other type of example that integrate the sequence comparison tools for the Ortholog Gene Finding (OGF) in multiple genomes. Orthologs [5] are genes retaining the same function in different species that evolved from a common ancestral gene by speciation.

Identification of orthologs is critical for reliable prediction of gene functions in comparative genome analysis. OGF needs high-throughput computing due to the increasing number of sequenced genomes that are more than 60 for microbial genomes and 10 eukaryotic genomes, respectively. In this workflow, genome sequences are converted into query sequence files by FileMerger and converted again into database for BLAST search by BlastDBformatter. Grid-BLAST executes all to all BLAST search and the best hit and gene ID are parsed from BLAST Output files. Ortholog-Picker finds orthologous genes. These series of works are described in DAG script file and are controlled by DAGMan.

3 Conclusions

Most of real world bioinformatics analyses are dealing with heavy computational complexity and subject specific integrated problems. The problem solving environment with simple workflow on grid system can be very efficient model to resolve these problems.

Acknowledgements

This research was partially supported by University IT Research Center Project and by national grid project grant 2004-giban-04 through Korea Institute of Science and Technology Information

References

1. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: The condor experience. Concurrency and Computation: Practice and Experience (2004)
2. Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic local alignment search tool. J Mol Biol. **215** (1990) 403–410
3. Pearson, W., Lipman, D.: Improved tools for biological sequence comparison. Proceedings of the National Academy of Sciences of the United States of America **85** (1988) 2444–2448
4. Smith, T., Waterman, M.: Identification of common molecular subsequences. J Mol Biol. **147** (1981) 195–197
5. Tatusov, R.L., Koonin, E.V., Lipman, D.J.: A genomic perspective on protein families. SCIENCE **278** (1997) 631–637