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

Byoung-Jin Kim, Chung-Hyun Sun, and Gwan-Su Yi*

School of Engineering, Information and Communications University, 103 -6 Munji-Dong, Yusung-Gu, Daejon 305-714, Korea BMDRC, Seoul, Korea

*To whom correspondence should be addressed. E -mail: gsyi@icu.ac.kr

Abstract

Grid system has the potential to resolve the current need of bioinformatics for super-computing environment inexpensively. There are already several Grid applications of bioinformatics tools. To solve the real-world bioinformatics problems, however, the various integration of each tool is necessary in addition to the implementation of more basic tools. Workflow based problem solving environment can be the efficient solution for this type of software development. There are still heavy overhead, however, to develop and implement workflow model on current Grid system. He re 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. We realized this model for practical bioinformatics solutions of a genome sequence analysis and a comparative genome analysis. We implemented necessary bioinformatics tools and interfacing tools as the components, and combine them in the workflow model of each solution by using the tools presented in Condor.

Introduction

The size of biological data to be managed and analyzed is increasing drastically with the progress of genomics researches and high-throughput biotechnology. As a result, this field meets inevitable need of high-throughput computing resources. Grid computation has been raised as a new solution for high-throughput and high-performance computing field, which matches with the compute-intensive bioinformatics application that includes the calculation of various NP complexity algorithms and the management of large data sets [1,2]. Recently, several groups have reported the examples of Grid application in bioinformatics field[3,4]. Most of the bioinformatics applications

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.

on Grid environment, however, are limited on the individual tools like sequence comparison methods which can be applied to solve the subset of biological data analysis[5]. Bioinformatics solution should be flexible and integrative to match the need of various biological problems. There are a few instances of integrated bioinformatics applications on Grid system and these are usually the results of very specific and complicated development[6].

It is noteworthy that some core analysis tools could be applied to many different bioinformatics subjects by adding some application specific tools. For example, the sequence comparison tools can be used in various genome sequence analyses with some different post-processing tools. One of the efficient solutions for this type of software development may be the workflow-based problem solving environment. There are still heavy overhead, however, to develop and implement this workflow model on current Grid system. It needs lots of effort to develop the high-level Grid utility for the specific applications. Although the Grid-computing environment has a significant potential to enhance the efficiency of bioinformatics research, those limitations huddle the propagation of Grid application in this field. To facilitate the development of Grid-application in bioinformatics, the overhead in interfacing the application and Grid-system software (Grid middleware) should be minimized. In this aspect, Condor[7] has lots of merit as a Grid supporting technology for writing Grid applications although two representative Grid middlewares are Globus[8] and Legion[9].

In this report, we propose a simple problem solving environment that allows easy implement of component based workflow of bioinformatics solution on Grid environment by using condor functionalities. We realized two examples of integrated bioinformatics solutions that are for genome sequence search with flexible selection of sequence comparison algorithms and for orthologous gene finding (OGF) among genomes. The implemented tools are Grid-BLAST, Grid-FASTA, Grid-Smith-Waterman algorithm as the common application components and the interfacing components specific for each of two integrated solutions.

Systems and Methods

Key Utilities of Condor for the Construction of Integrated Application on Grid

The goal of Condor is somewhat different from Globus or Legion that try to fulfill the role of general Grid-system software supporting complete Grid. Condor is rather specialized to support high-throughput computing applications on Grid system[10]. It has many resource management and job management functions supporting implementation and management of the computing application on Grid[11,12]. Here we introduce key features of Condor that we used for the implementation of our Grid application.

Workflow Design Utilities

DAGMan is a meta-scheduler that makes simple workflow design on gird possible. The DAG scripts can handle the dependency and the schedule of each job in DAG. DAGMan holds and submits jobs to the Condor queue at the appropriate times. We could simply make the workflow structure for our solution by define each executable component and their dependencies with DAG scripts as shown in later section of this paper.

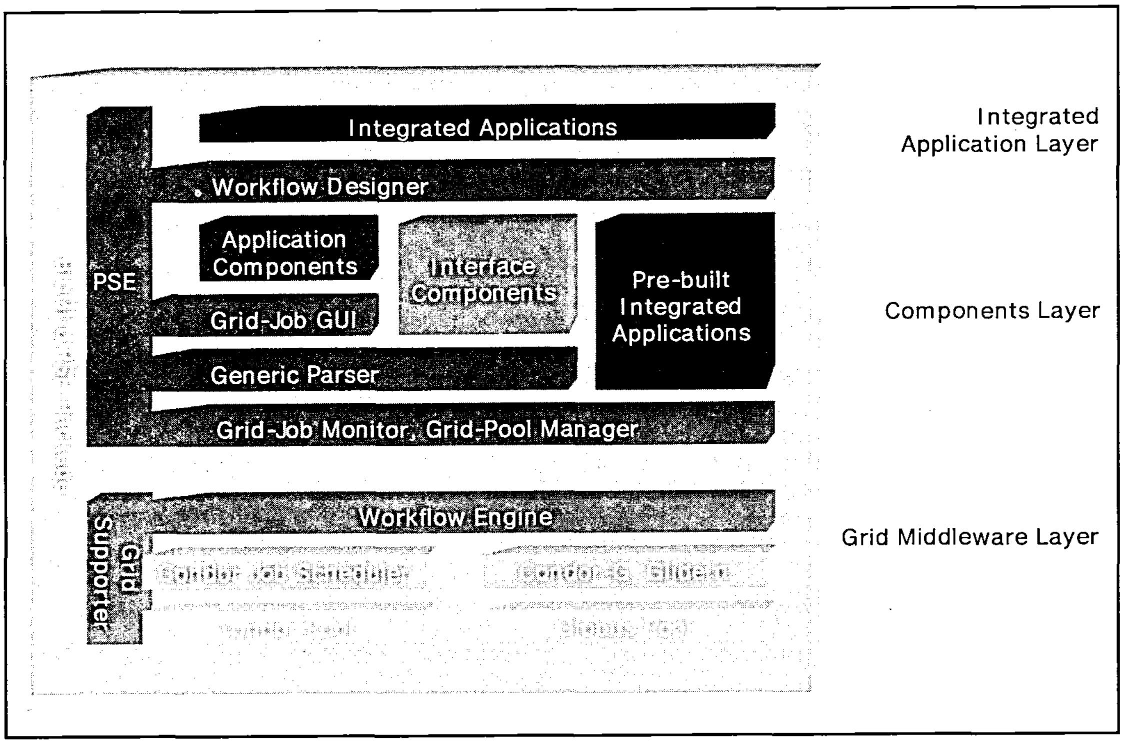

Fig. 1 Structure of bioinformatics problem solving environment on Grid.

Grid Computing Support Utilities

Basically Condor has a feature of flocking which allows the job submission among different Condor pools. Condor jobs can be submitted to the resources accessible through Globus interface by Condor-G In addition even if the architectures of middleware systems are different[13]. GlideIn enables the extension of Grid pool in reverse way. When the resources in other Grid middleware run GlideIn, they will temporarily join Condor pool and can share the resources. For example, Condor resources can be combined with Globus resources by using GlideIn on Globus. In these ways, the Grid applications on Condor system can be extended into the larger Grid.

Job Management and Resource Monitoring Utilities

Checkpoint[12] and migration utility help to secure and complete job. Many desired policies and requirement of job can be described by ClassAd mechanism[14] for resource status such as RAM memory, CPU type, CPU speed, Virtual memory size, physical location, and current load average.

Results and Discussion

Model of Integrated Bioinformatics Solution on Grid

We propose a model of the bioinformatics problem solving environment enabled by component based workflow design of integrated bioinformatics applications on Grid and show the practical example of implementation. Fig. 1 shows the structure of this model. It has basically two layers: Components Layer and Integrated Applications Layer. Components layer consists of application component and interfacing component. They are Grid enabled executable files or a Grid-enabled program inherently. Application component is an application which makes it possible to analyze bioinformatics data independently. Interfacing components include such functions like pre- and post-processing of application components, interfacing for input and output files and user interface. These components are reusable to build different integrated solution. Interfacing components take charge of important role to give the flexibility to the system corresponding to the diverse integrating request of bioinformatics problems. Integrated application can be a final solution or another application component. It usually takes advantage of a Condor command script file and a DAG script file which are executable by Condor. Various workflows corresponding to the specific solutions can be constructed by proper arrangement of components by using these Condor scripts and the system can achieve the flexibility corresponding to the diverse request of bioinformatics problems. The following subsections show implemented components and two example of integrated bioinformatics solution made in our model environment.

Application Components

One of primary bioinformatics applications is sequence alignment tool, which aligns a pair of DNA (or protein) sequences based on their similarity. Another fundamental tool in bioinformatics as a direct application of sequence alignment is the homologous sequence search tool. Diverse integrated bioinformatics solutions can be made by applying these tools. Because of the popularity of these tools and the need of our final integrated solutions, we implemented them as application components.

Grid-BLAST/Grid-FASTA

BLAST[5] and FASTA are popular applications that search sequence similarity and homology based on the local sequence alignment for query sequence against sequence database. We developed Grid-BLAST and Grid-FASTA to enable them to run in Grid environment. These tools facilitate searching homologous sequences in multiple sequence databases for various numbers of querying sequences by using 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. Condor negotiates proper computational resources for each job and allocates them. Each result files are saved in specific directory of the submission machine.

Grid-SWSearch

Grid-SWSearch is a homologous sequence searching program by pair-wise sequence comparison based on Smith-Waterman algorithm against sequence database on grid. As Smith-Waterman algorithm[15] is one of dynamic programming algorithm, it gives more precise results then output from FASTA (or BLAST) but requires longer execution time. We implemented improved Smith-Waterman algorithm which reduces (memory) space complexity by linear space algorithm[16] and computational complexity by the modified Gotoh's algorithm[17]. In Grid-SWSearch, database and query sequences are divided into pieces of bigger or same number than the number of nodes and assigned to each node to parallelize the work. Each job doesn't pass any massage to each other to remove communicational overhead and to run independently from each job. 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 (see below) to see sequence alignment if necessary.

Grid-SWAlign

Grid-SWAlign is a grid enabled sequence alignment program based on Smith-Waterman algorithm. This tool only has sequence alignment functionality and is useful when only one to many pair-wise sequence alignments for selected sequences are necessary. Grid-SWAlign receives multiple sequences to be aligned and makes a bundle of jobs on grid for all combinations of pair-wise alignment.

Ortholog-Picker

Ortholog-Picker is an application which generates orthologous genes from BLAST outputs. BLAST outputs are produced from BLAST search of every query sequences against genome sequence database files. Then the best hit and gene ID are parsed from each BLAST Output file. Ortholog-Picker selects orthologs that are, in our definition, the gene sequences of at least 2 reciprocal best hits among the sequences of compared genomes. Ortholog-Picker can be modified by different definition of ortholog constraining such as minimum number of reciprocal best hit and alignment score. Grid-FASTA output and Grid-SWSearch outputs can be also used as input instead of BLAST outputs.

Interfacing Components

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.

Integrated Applications

Integrated Sequence Comparison

There could be the need to use different type of sequence search and alignment methods or the combination of them depending on the purpose and the condition of sequence analysis. In our problem solving environment, we could make several integrated jobs of sequence comparison easily. First of all, we improve the efficiency of Smith-Waterman algorithm tools with the benefit of grid system. As shown in Fig. 2 (a), 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. They run and migrate if necessary.

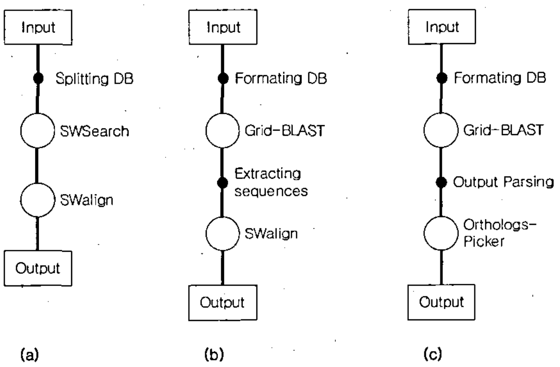

Fig. 2 Examples of workflows for the integrated bioinformatics solutions deployed on grid enabled problem solving environment. Various simple solutions for integrated sequence comparison can be designed as shown in (a) and (b). A unique solution like orthologous gene finding (OGF) can be made as shown in (c) with the addition of specific application component such as Ortholog-Picker. Application components and interafacing components are represented by white circles and black dots, respectively.

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, one may want run grid-SWAlign alone for just several sequences and inspect details of aligned part of sequences and maybe include one more component for parsing some feature in aligned part. Or, as shown in Fig. 2 (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.

Comparative Genome Analysis : Orthologous Gene Finding (OGF)

The second example solution that can be integrated by the sequence comparison tools is the ortholog finding in multiple genomes. Orthologs 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 and 10 for microbial genomes and eukaryotic genomes, respectively [18]. Fig. 2 (c) shows the workflow of OGF on grid. Genome sequences are converted into query sequence files and database for BLAST search by FileMerger and 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.

Conclusions

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

References

[1] National Center for Biotechnology Information. The genbank nucleotide sequence database. *http: // www. ncbi. nlm. nih. gov/ Genbank/genbankstats. html*.

[2] John Bent, Venkateshwaran Venkataramani, Nick LeRoy, Alain Roy, Joseph Stanley, Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, and Miron Livny. *Grid Resource Management*, chapter NeST - A Grid Enabled Storage Appliance. Kluwer Academic Publishers, 2003.

[3] A. Krishnan. Gridblast: High throughput blast on the grid. In *2nd International Conference on Natural Products*, Singapore, July 2002.

[4] Wei Shi and Wanlei Zhou. Large-scale biological sequence assembly and alignment by using computing grid. *GCC(Grid and Cooperative Computing)*, pages 26-33, 2003.

[5] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J Mol Biol.*, 215(3):403-410, October 5 1990.

[6] Junwei Cao, Jochen Fingberg, Guntram Berti, and Jens Georg Schmidt. Implementation of grid-enabled medical simulation applications using workfow techniques. *GCC(Grid and Cooperative Computing)*, pages 34-41, 2003.

[7] Douglas Thain, Todd Tannenbaum, and Miron Livny. Condor and the grid. In Fran Berman, Geoffrey Fox, and Tony Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons Inc., December 2002.

[8] I. Foster. A new infrastructure for 21st century science. *Physics Today*, 55(2):42-47, 2002.

[9] A. S. Grimshaw, M. A. Humphrey, and A. Natrajan. A philosophical and technical comparison of legion and globus. *IBM J. RES. & DEV*, 48(2):233-254, 2004.

[10] Douglas Thain, John Bent, Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, and Miron Livny. Pipeline and batch sharing in grid workloads. *in Proceedings of the Twelfth IEEE Symposium on High Performance Distributed Computing*, Seattle, WA, 2003.

[11] Todd Tannenbaum, Derek Wright, Karen Miller, and Miron Livny. Condor - a distributed job scheduler. In Thomas Sterling, editor, *Beowulf Cluster Computing with Linux*. MIT Press, October 2001.

[12] Michael Litzkow, Todd Tannenbaum, Jim Basney, and Miron Livny. Check-point and migration of UNIX processes in the Condor distributed processing system. Technical Report UW-CS-TR-1346, University of Wisconsin -Madison Computer Sciences Department, April 1997.

[13] J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke. Condor-g: A computation management agent for multi-institutional grids. *Cluster Computing*, pages 237-246, 2002.

[14] Rajesh Raman, Miron Livny, and Marvin Solomon. Resource management through multilateral matchmaking. In *Proceedings of the Ninth IEEE Symposium on High*

Performance Distributed Computing (HPDC9), pages 290-291, August 2000.

[15] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J Mol Biol.*, 147(1):195-197, March 25 1981.

[16] K.M. Chao, R.C. Hardison, and W. Miller. Recent developments in linear-space alignment methods: a survey. *J Comput Biol.*, 1(4):271-291, Winter 1994.

[17] O. Gotoh. An improved algorithm for matching biological sequences. *J Mol Biol.*, 162(3):705-707, December 15 1982.

[18] Roman L. Tatusov, Eugene V. Koonin, and David J. Lipman. A genomic perspective on protein families. *SCIENCE*, 278:631-637, 1997.