Biologic Data Analysis Platform Based on the Cloud

Hoon Choi, Sang-Hwan Lee and Dong-In Park

Korea Institute of Science and Technology Information
{choid, sanglee, dipark}@kisti.re.kr

Abstract

To improve the research productivity in bioinformatics study by using effective means of large scale data analysis, there are many obstacles that need to be overcome They are standardization of data collection and analysis, management of computing and storage resources, easiness of parallel programming, and efficiency of data analysis job execution, to name a few. Among these, easiness of parallel programming is a crucial factor that contributes to usability and efficiency of large scale data analysis.

This paper describes a biologic data analysis platform based on cloud computing infrastructure. The platform provides an easy-to-use parallel data analysis environment, and ultimately enhances the productivity of bioinformatics research.

Keywords: *cloud computing, parallel programming, biologic data analysis, data management*

1. Introduction

With increasing amount of public biologic data and emerging high performance cloud computing, bioinformatics researchers are now capable of analyzing massive amount of data for a relatively cheap price [1]. Bioinformatics researchers prepare data sets by collecting and compiling from various disparate data sources, and compose analysis pipelines that consist of published or in-house programs and library functions. They read the prepared data sets and execute the pipelines with the computing and storage resources on their own and from public cloud providers.

In order to improve the research productivity by using effective analytical tools, there are so many obstacles that need to be overcome, such as time-consuming efforts of data collection and transformation, manual management of computing and storage resources, difficulty of parallel programming, and inefficiency of executing data analysis jobs. Among these, the difficulty of parallel programming is a crucial factor hampering a faster research.

Bioinformatics researchers tend to analyze data using programs that are already published. For this reason, they continue to spend numerous months on transformation of legacy sequential programs into scalable parallel programs suitable to do high performance computing. The improvement of scalability of sequential programs is achieved by fully utilizing the configuration of high performance computing systems.

For this, bioinformatics researchers study parallel programming techniques such as MPI [2], MapReduce [3], OpenMP [4] and Pthread [5] to develop and optimize the parallel scalable programs. At times, bioinformatics researchers hire professional developers to do it.

Nonetheless, newly developed legacy applications do not always meet the scalability. The speed of hardware development is faster than that of application program development. This kind of situation will always occur. There are two ways to solve this problem. If the scale of a problem to be solved is large, then optimized parallel programs need to be developed. In case of a relatively small problem, it is highly recommended that a data analysis platform based on virtual infrastructure is used.

This paper describes a biologic data analysis platform (BioDAP) based on virtual infrastructure, in order to enhance the productivity of bioinformatics research and the usability of biologic data analysis programs. BioDAP consists of a biologic data management service environment and a biologic data analysis programming environment. The biologic data management system manages data types of gene sequences and protein-protein interaction networks. The biologic data analysis programming environment deals with designing biologic data analysis pipelines, collecting and integrating heterogeneous data sources, and managing function library. Purpose of the biologic data analysis programming environment is to make the development of biologic data analysis programs more effective and usable, and to ultimately increase research productivity. BioDAP is based on virtual infrastructure management system [6] developed by the experience of KISTI (Korea Institute of Science and Technology Information) and KOBIC (Korean Bioinformation Center).

2. Requirements of BioDAP

Bioinformaticians compose an analysis pipeline to handle biologic data from multiple data sources. An exemplary user scenario is shown in Figure 1.

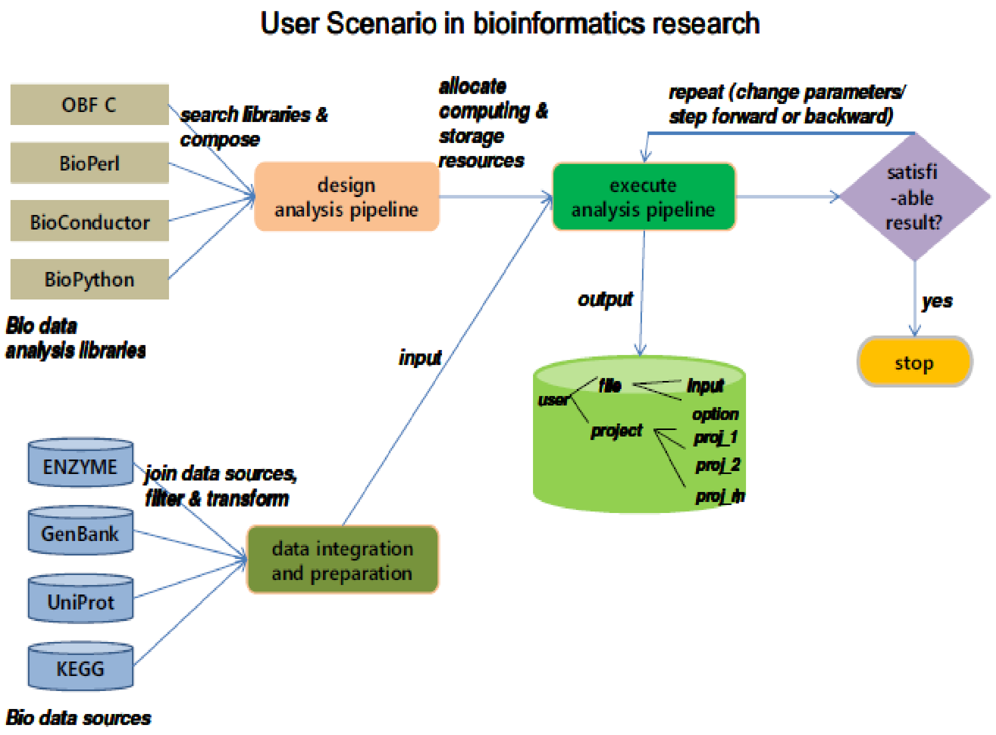

User Scenario in bioinformatics research
OBF C
BioPerl
BioConductor
BioPython
Bio data analysis libraries
search libraries & compose
design analysis pipeline
allocate computing & storage resources
execute analysis pipeline
repeat (change parameters/ step forward or backward)
satisfi -able result?
yes
stop
output
user
file
input
option
project
proj_1
proj_2
proj_in
input
ENZYME
GenBank
UniProt
KEGG
Bio data sources
join data sources, filter & transform
data integration and preparation

Figure 1. User Scenario in Bioinformatics Research

When analyzing a massive amount of data according to the user scenario, the barrier that bioinformaticians can encounter is the fact that it takes much time to obtain the result from an idea or a problem. Some of the causes in this difficulty are explained below:

- Bioinformaticians themselves have to put excessive amount of effort for collecting and integrating data from multiple data sources. In other words, tasks such as collecting data from various data sources, reformatting, joining, aggregation, and storing them are all done manually

- As bioinformaticians develop programs individually, or hire professionals to develop programs, the applicability of the programs is limited to only certain types of data. This heterogeneous nature between programs caused by what types of data they deal with is an obstacle for composing a pipeline that is needed for analysis of data. Even if there is a good program, the interface is not standardized; therefore it is not simple to integrate the program with the pipeline.

- After bioinformaticians compose an analysis pipeline, they analyze data according to the pipeline. They execute the analysis pipeline in a step by step process, rather than executing the whole pipeline all at once. In other words, researchers assess an output of a step, and then decide to go back to the previous step, stop, or move on to the next step. They continue to analyze data until they finds something that is scientifically meaningful. It is required that all the output values in the middle of the pipeline are managed, and the provenances which represent the relationships between input and output data sets are also managed.

- Bioinformaticians use multi-core computing. This involves parallel programming that is based on parallel programming models like MPI, OpenMP, MapReduce, and Pthread. Theses parallel programming models are not easily utilized by bioinformatics researchers. Thus, parallelization of sequential programs is time consuming.

These obstacles can be overcome through a data integration system, a pipeline composition tool based on workflow, an analysis pipeline execution engine, a data set and provenance management system, and a parallel programming environment that prioritizes the usability.

3. Design of BioDAP

In this section we describe the design of BioDAP to provide bioinformatics researchers with the usable and efficient analysis of biologic data from multiple data sources. To do this, BioDAP provides biologic data management service (BDBMS) environment and biologic data analysis programming (BDAP) environment on the top of parallel programming runtime system and virtual infrastructure management system. The architecture of BioDAP is shown in Figure 2. The BDBMS environment consists of data integration system, data set and provenance management system, and semantic mapping repository based on ontology. The BDAP consists of analysis pipeline composition tool, analysis function library, test data generator, debugger and analysis pipeline execution engine to enhance the programmability and traceability of parallel processing of biologic data. The parallel program runtime system, that supports MapReduce, OpenMP and MPI, hides the difficulty of creating and scheduling parallel tasks on virtual clusters. Virtual infrastructure management system creates and allocates an isolated cluster for each user’s job. Hence, BioDAP relieves bioinformaticians from the efforts of data preparation, parallel programming, task scheduling, and so on involved in parallel processing of data analysis.

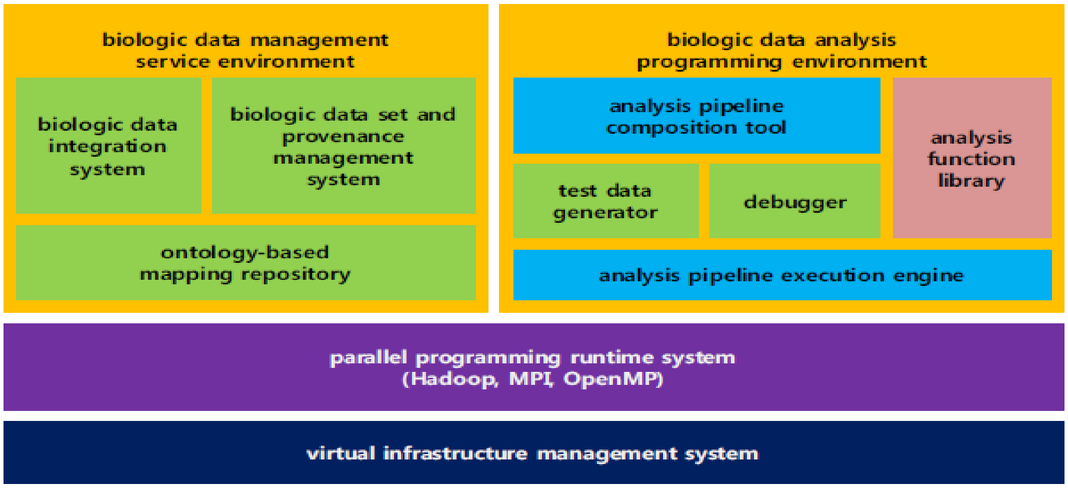

biologic data management service environment
biologic data integration system
biologic data set and provenance management system
ontology-based mapping repository
biologic data analysis programming environment
analysis pipeline composition tool
analysis function library
test data generator
debugger
analysis pipeline execution engine
parallel programming runtime system (Hadoop, MPI, OpenMP)
virtual infrastructure management system

Figure 2. Architecture of BioDAP

3.1. Virtual Infrastructure

There are two ways for resource scheduling to improve the physical resource utilization. One way is to provide a physical cluster that is required to execute an application using a fine grained scheduler like MESOS [7], without any virtualization. The other way is to make a virtual cluster that suits the application, using virtualization technology.

The first way is done through assigning a part of cores in a node to a legacy application as many as it needs to be executed, and then assigning the remaining cores in a node to other applications. The latter is easier to implement, and it allows resource isolation among virtual clusters. Since the virtual clusters do not interfere with each other, application can be executed at a higher performance and fault-tolerance even in the case of partial failure of each node. The only problem in virtual clusters is the fact that virtualization overhead can occur. On the other hand, fine grained scheduler causes no virtualization overhead. However, each node has to manage available number of cores in each node, and applications need to share resources at random. For this reason, development of fine grained scheduler is complicated and requires highly advanced system programming skills. Therefore, it is expensive and difficult to modify and customize the fine-grained scheduler. Hence, BioDAP is developed on the top of virtual infrastructure.

3.2. Biologic Data Integration System

Data integration system is able to collect specific data from various heterogeneous data sources. Data integration requires standardization of data format, data exchange protocol, vocabulary for controlling metadata, as pre-requisites. It also requires semantic mapping to integrate distributed data sources. For integration, construction of ontology-based mapping repository is essential.

3.3. Data Set and Provenance Management System

Data sets produced from each step of a pipeline should be managed with its provenance. The provenance includes information about input data set, analysis programs, and output data set and its version.

3.4. Data Analysis Programming Environment

Data analysis programming environment needs to ensure that bioinformatics researchers have usability and efficiency in parallel programming, and ultimately enhance research productivity. Hence, data analysis programming environment supports researchers with designing, implementing, and testing of parallel programs through biologic data integration system, analysis pipeline composition tool, analysis pipeline execution engine, and data set and provenance management system.

Conventionally, a data processing program is developed only after input data is refined and desired output is clearly defined. A developer starts to develop a program after he/she defines the data structure by browsing the data, and maps out required algorithm to get the desired result. However, when there is a situation where there is too much data, where data is not refined, or where it is impossible to pre-define or specify what valuable information is in the data, the way to developing programs is very far from the conventional method. A program evolves as data gets sampled, browsed, coded, and tested. Furthermore, in case of massive data analysis, it is difficult to define valuable information. Therefore, it is difficult to specify a processing logic to produce a desired output value. In this case, it is more appropriate to adopt an evolutionary method to develop analysis programs. Furthermore, the processing logic of a program would rather be expressed in a data flow style based on the flow of input and output data. Evolutionary development of data analysis programs needs to be supported with a data flow language based coding and test tools that include a data generator and a debugger.

Analysis pipeline composition tool provides researchers with ways to easily building pipelines for problem solving. A script language such as Python [8] is commonly used, but Parallel Python [9] and Scala [10] will be used for parallel processing. Data flow language can compose data analysis pipelines, and it can also handle parallel processing based on data partitioning. A commonly used data flow language for massive data analysis is Pig Latin [11]. Pig dataflow engine compiles a Pig Latin program to a sequence of MapReduce jobs. For this reason, bioinformatics researchers can use Pig in order to develop a parallel program efficiently and easily. The usability of the pipeline composition tool depends largely on the variety of library functions. Generally, statistical and mathematical models used commonly in bioinformatics research are reflected in R package [12].

3.5. Analysis Pipeline Execution Engine

Bioinformatics researchers store all the execution results from each step of the pipeline and present them. Users decide whether they will move forward, stop, or go back according to the result. In order to support the users in this process, a pipeline execution engine not only monitors the user’s pipeline execution, but also stores the execution results in different steps of the pipeline and sends them to the users. To serve this purpose, a pipeline execution engine interacts with data set and provenance management system and stores data, makes the data available for searching, controls access to provenance, and does version control.

4. Related Work

BioDAP is similar to Google App Engine [13] and VMWare CloudFoundry [14] in terms of providing users with application programming environment and data management facility. However, the analysis pipeline execution engine and the biologic data set and provenance management system are the components specific to BioDAP.

The pipeline execution engine has common features with scientific workflow [15, 16]. The difference between them is the fact that the pipeline execution engine traces repeated executions of a step and going a step forward or backward. To keep the results according to the execution flow of analysis pipeline, it is closely connected to the data set and provenance management system and stores the data sets with annotation and version control.

The data set and provenance management is currently under development in SciDB project [11]. SciDB is a multi-dimensional array data management system that supports scientific analysis applications such as R. As multi-genome analysis is becoming essential for personalized medicine in near future, SciDB is a good option for data management system in BioDAP.

5. Conclusions

We described BioDAP based on virtual infrastructure, in order to enhance the productivity of bioinformatics study and the usability of developing biologic data analysis programs. BioDAP provide bioinformatics researchers with a biologic data management service environment and a biologic data analysis programming environment. The biologic data management system manages data types of gene sequences and interaction networks. The biologic data analysis programming environment deals with designing pipelines for biologic data analysis, collection and integration of input data sets from heterogeneous disparate data sources, and management of functions library. BioDAP is based on virtual infrastructure management system developed by KISTI. BioDAP described in this paper contributes the programmability of pipelines for biologic data analysis and ultimately increase the productivity of bioinformatics research.

The data set and provenance management is currently under development in SciDB project [11]. SciDB is a multi-dimensional array data management system that supports scientific analysis applications such as R. As multi-genome analysis is becoming essential for personalized medicine in near future, SciDB is a good option for data management system in BioDAP.

Acknowledgements

This work was supported by the IT R&D program of MKE/KEIT [10038768, The Development of Supercomputing System for Genome Analysis].

References

[1] L. Stein, “The Case for Cloud Computing in Genome Informatics”, Genome Biology, vol. 5, no. 11, **(2010)**.

[2] MPI. http://www.mcs.anl.gov/research/projects/mpi/.

[3] J. Dan and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Proceedings of the 6th Symposium on Operatinr Systems Design and Implementation, San Francisco, USA, **(2004)** December 6-8.

[4] OpenMP. . http://openmp.org/wp/.

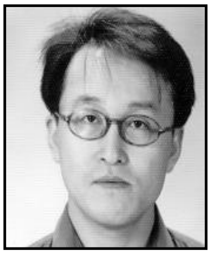
[5] Pthread. https://computing.llnl.gov/tutorials/pthreads/.
[6] J. Um, H. Choi, S. Song, S. Choi, H. Yoon, H. Jung and T. Kim, "Development of a virtualized supercomputing environment for genomic analysis", The Journal of Supercomputing, **(2012)**.
[7] B. Hindman, A. Konwinsky, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker and I. Stoica, "MESOS: A Platform for Fine-Grained Resource Sharing in Data Center", Proceedings of 8th Symposium on Networked Systems Design and Implementation, Boston, USA, **(2011)** March 30-Aril 1.
[8] Python. http://www.python.org/.
[9] Parallel Python. http://www.parallelpython.com/.
[10] Scala. http://www.scala-lang.org/.
[11] C. Olston, B. Reed, U. Srivastava, R. Kumar and A. Tomkins, "Pig Latin: A Not-So-Foreign Language for Data Processing", Proceedings of ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, **(2008)** June 9-12.
[12] R Package. http://www.r-project.org/.
[13] Google App Engine. https://developers.google.com/appengine/.
[14] Cloud Foundry. http://www.cloudfoundry.com/.
[15] Kepler. Scientific workflow. https://kepler-project.org/.
[16] Y. Han, "Bioworks: A Workflow System for Automation of Bioinformatics Analysis Processes", International Journal of Bio-Science and Bio-Technology, vol. 4, no. 3 **(2011)**.
[17] SciDB. http://www.scidb.org/

Authors

Hoon Choi is a principal researcher in the Department of Scientific Big Data Research in Korea Institute of Science and Technology Information (KISTI). He is working on the DB-centric computing for data intensive science

He received B.S. in Computer Science at Seoul National University in 1981, and received Ph.D. in Computer Science at Northwestern University in 1989.

Sang Hwan Lee, works as chief researcher and the head of the Department of Scientific Big Data Research at Korea Institute of Science and Technology Information (KISTI), Korea. He received his M.S. degree in Computer Science and Engineering from Korea University, Korea. His current research interests are Database, Text Mining, and Data-Intensive Science.



Dong-In Park is a principal researcher in the Department of Scientific Big Data Research at Korea Institute of Science and Technology Information (KISTI). He received his B.S. degree in Electrical Engineering from Sogang University at 1979. He has been working on natural language processing since 1979. He is a well-known scientist in the area of natural language processing, who achieved some striking performance in automation of Korean language. His current research interests are Big Data Analysis and Text Mining.