pISSN 1976-913X
eISSN 2092-805X

J Inf Process Syst, Vol.8, No.4, December 2012 http://dx.doi.org/10.3745/JIPS.2012.8.4.555

An Adaptive Workflow Scheduling Scheme Based on an Estimated Data Processing Rate for Next Generation Sequencing in Cloud Computing

Byungsang Kim*, Chan-Hyun Youn**, Yong-Sung Park**, Yonggyu Lee** and Wan Choi***

Abstract—The cloud environment makes it possible to analyze large data sets in a scalable computing infrastructure. In the bioinformatics field, the applications are composed of the complex workflow tasks, which require huge data storage as well as a computing-intensive parallel workload. Many approaches have been introduced in distributed solutions. However, they focus on static resource provisioning with a batch-processing scheme in a local computing farm and data storage. In the case of a large-scale workflow system, it is inevitable and valuable to outsource the entire or a part of their tasks to public clouds for reducing resource costs. The problems, however, occurred at the transfer time for huge dataset as well as there being an unbalanced completion time of different problem sizes. In this paper, we propose an adaptive resource-provisioning scheme that includes run-time data distribution and collection services for hiding the data transfer time. The proposed adaptive resource-provisioning scheme optimizes the allocation ratio of computing elements to the different datasets in order to minimize the total makespan under resource constraints. We conducted the experiments with a well-known sequence alignment algorithm and the results showed that the proposed scheme is efficient for the cloud environment.

Keywords—Resource-Provisioning, Bio-Workflow Broker, Next-Generation Sequencing

1. INTRODUCTION

To date, workflow management systems (WMS) in the distributed environment are for scientific applications to solve sophisticated problems such as genomic analysis, drug discovery, disease identification, etc. As scientific applications become more complex, the management of resources that perform the workflow jobs has become one of the challenging issues [1, 2]. Recently many large research centers and universities have conducted the studies on the high per-

※ This research was equally supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF), which is funded by the Ministry of Education, Science and Technology (2012-0006425); and by the Next-Generation Bio-data Based Genome Research & Commercialization, which is funded by the Ministry of Knowledge Economy (10040231); and by the IT R\&D program of MKE/KEIT. (10038768, The Development of a Supercomputing System for Genome Analysis)
Manuscript received January 25, 2012; first revision June 5, 2012; accepted September 6, 2012.
Corresponding Author: Chan-Hyun Youn
* Dept. of Information and Communications Engineering, KAIST, Daejeon, Korea (bs.kim@kaist.ac.kr)
** Dept. of Electrical Engineering, KAIST, Daejeon, Korea ({chyoun, lyonggyu, miracle0318}@kaist.ac.kr)
*** Electronics and Telecommunications Research Institute, Daejeon, Korea (wchoi@etri.re.kr)

Copyright ⓒ 2012 KIPS

formance-computing platform to address the analysis of genomic data. The GNARE [3] project is one of the biggest genomic grid projects, which is conducted by the Argonne National Laboratory. It analyzes the genomic sequence and submits computing-intensive to the grid environment. Similarly the PUMA2 [4] system is an interactive, integrated bioinformatics environment for high-throughput genetic sequence analysis and metabolic reconstructions from sequence data. In addition, a new generation of non-Sanger-based sequencing technologies (next generation sequencing: NGS) has delivered on its promise of sequencing DNA at unprecedented speed, thereby enabling impressive scientific achievements and novel biological applications [5].

Recent advances in cloud computing have made it possible to analyze very large data sets in scalable and cost-effective ways. However, most of them focus on a batch-processing scheme of data stored in a local file system [6]. Even though many solutions give us various distributed methods for analyzing huge data, they focus on static resource provisioning with batch processing scheme in local computing farm and data storage. Since all datasets should be located in the local site before the processing starts, the transfer overhead of a huge amount of input datasets, as well as an unbalanced execution time of different problem size, increases the total completion time.

In this paper, we first describe a distributed bio-workflow broker (DBB) model, which cooperates with different workflow applications. On the DBB, we propose an adaptive resource-provisioning scheme, including run-time data distribution and collection service (DCS), which decouples the data transfer process and data processing processes. The DCS makes it possible to eliminate the transfer time for both the input and output dataset from local storage to cloud data centers by parallelizing the data placement and data processing module. On the DCS platform, the adaptive processing element provisioning problem (APEPP) is proposed to optimize the allocation ratio of computing elements to each dataset in order to minimize the total makespan under resource constraints. For evaluating our proposed scheme, we utilize a well-known algorithm for next generation sequencing tools and a pipelined application, which was used in a 2009 genome research article for the first Korean genome [9].

The remainder of this paper is organized as follows: in Section 2, we describe the workflow that has been integrated with the NGS model and the cloud based sequence alignment service. In Section 3, we propose an adaptive resource-provisioning scheme for the run-time data distribution and collection service. We evaluated our algorithm with six new different sequence alignment datasets and the effectiveness of our proposed model in Section 4. Finally, we make a conclusion in Section 5.

2. WORKFLOW INTEGRATED BIO-COMPUTING ENVIRONMENT

2.1 Integrated Genome Workflow Model

A new generation of non-Sanger-based sequencing technologies have delivered on its promise of sequencing DNA at an unprecedented speed, thereby enabling impressive scientific achievements and novel biological applications [5]. Metabolic syndrome is a cluster of symptoms such as diabetes, obesity, hyperlipidemia, and high blood pressure [1]. However, we are still not really sure whether this type of disease prediction system is trustworthy or not. For the case of complex diseases such as diabetes, we do not know for sure how many SNPs are related and it is absolutely certain that very complex gene-gene or SNP-SNP interactions would be made. Also,

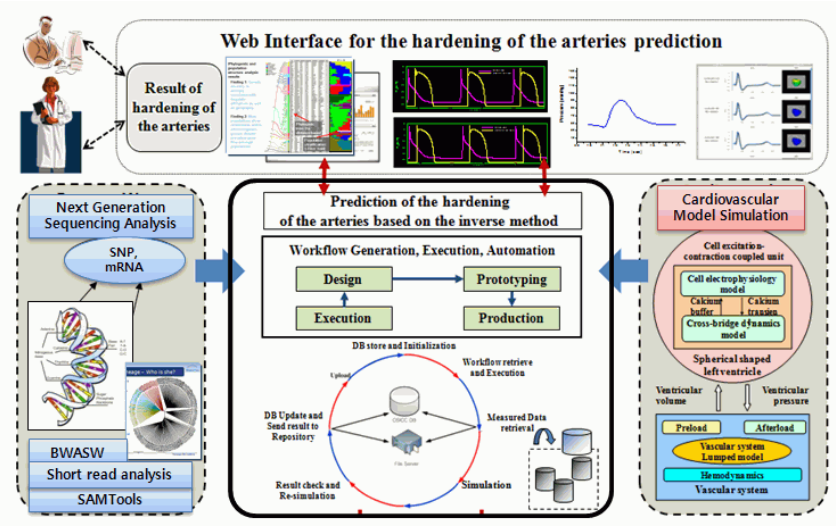


Fig. 1 System architecture for integrated genome analysis and cardiovascular simulation workfkow

there exists various kinds of sub-types of diabetes; each of which has different genetic and environmental factors. In the bioinformatics technology, the next generation sequencing method which provides the sequence matching between reference data to personal DNA samples, provides a faster mapping algorithm compared to conventional approaches [8]. The complexity of the applications to compose the individual tasks and the data processing capability requires large scale workflow management methodologies and the parallelism for reducing total execution time as shown in Fig.1.

2.2 A Distributed Bio-Workflow Broker on Clouds Computing Environment

To cooperate between workflows in real-time, we describes a conceptual idea of the integrated workflow system. The middle layer includes functions that manage the interaction among different bio-workflow services such as the pressure wave monitoring, the cardiovascular model simulation, and the genome sequencing alignment services that have been implemented in the workflow model. The system could automate the simulation data and the experimental workflow processes. This type of large-scale data analysis workflow model needs a huge size of computing and storage infrastructure for performing overall workflow tasks with an in-house method. However, it is expensive to prepare enough resources and the efficiency of the resources is relatively low since all tasks do not require the same computing capacity. For the hybrid cloud model, it is possible to outsource the entire or a part of the workflow tasks into the public cloud. In this case, on-demand resource provisioning is possible whenever it is needed. We suggest a hybrid cloud model for the workflow-computing model, which has distributed workflow services combined by cloud service model. As shown in Fig. 2, the distributed bio-workflow broker (DBB) system is located in the middle layer between the end user and the cloud service. The DBB functions as the bridge between the bio-services and the cloud data centers, as shown in Fig.2. The DBB stores the bio-service services, such as the SNP analysis for DNA or for the metabolic disease identification system, with both the genome database and the cell metabolism

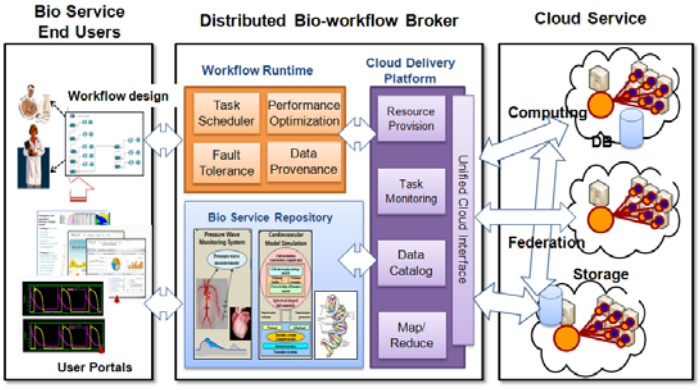


Fig. 2. Distributed bio-workflow model combined by cloud service

measurement with a unified interface.

To control the workflow execution flow, a task monitoring function is built to identify the task status including submission, execution, and publishing. However, the solutions focus on static resource provisioning with a batch-processing scheme in local computing farm and data storage. Since all datasets should be located in the local site before the processing starts, the transfer time of a huge dataset, as well as an unbalanced execution time of different problem sizes, increases the total completion time. In the following section, we propose the adaptive resource-provisioning scheme for both the data preparation process as well as for the data processing process.

3. ADAPTIVE RESOURCE PROVISIONING ON CLOUDS

3.1 A Continuous Data Distribution and Collection Scheme

We propose to use a data distribution and collection service (DCS) for handling a real-time data analysis platform that has been deployed on cloud computing environment in order to prevent data transfer latencies. Since the DCS stores a small part of the total dataset, it should continuously replenish the input data from the data source. The DCS manages such continuous data streams from the data source to the processing elements (PEs), which are located in remote data centers. Let $\{j|j=1,2,...,J\}$ be the kind of genome dataset for analyzing the sequence. As shown in Fig. 3 the DCS has finite buffer pairs - an application data buffer (ADB) and a receiver queue (RQ) for each dataset j. The ADB is a temporal storage for application input data and the RQ functions as the admission controller for demands from lower level nodes. The upstream queue and downstream queue means the communication channel. Assuming that a data object is a countable processing unit, the buffers in the DCS are modeled as queuing systems for which the arrival and departure processes are the data transfer time and data processing rate.

For each dataset, the stability condition is constrained by:

$$\forall j \in J,\ \mu_j(t) \geq \lambda_j(t) \tag{1}$$

where $\mu_j(t)$ and $\lambda_j(t)$ are the data transfer rate and data processing rate for each dataset j.

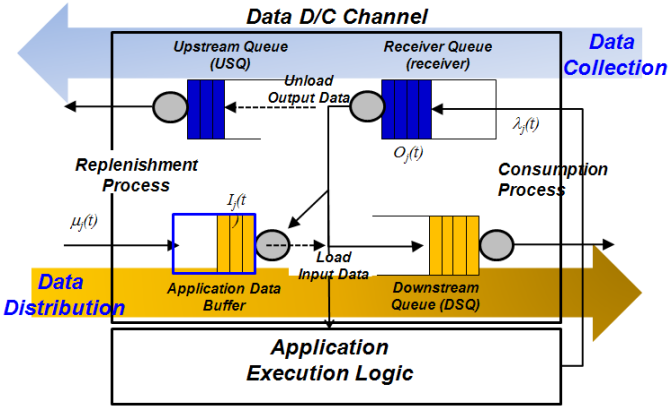


Fig. 3. The data distribution and collection service (DCS) deployed in a public cloud

3.2 The Run-time Processing Element Provisioning Algorithm

For all dataset J, we describe an adaptive process element provisioning problem (APEPP) under a given set of PEs with the following notations:

The goal of the APEPP is to find the optimal PE distribution ratio α_j on given datasets. This is achieved by equalizing the makespan of each stream j such as:

- Adaptive PE provisioning problem (APEPP):

$$\begin{aligned} &\text{minimize} && T[D] \\ &\text{subject to} && \mathrm{T[D]} = \max_{j \in J} \left[T[D_j(t)] \right] \\ & && \sum_{j \in J} \alpha_j = 1 \end{aligned}$$

In order to obtain the APEPP, we utilize the proportional allocation method. Let $\gamma_j(t)$ be the *unit processing rate* of stream j at time t as:

Table 1. Notations and Descriptions

Notations	Descriptions
K	Available number of PEs
$D_j(t)$	Remaining dataset of stream j on time t
$\lambda_j(t)$	Processing rate of stream j on time t
$T[D_j(t)]$	Estimated completion time of the stream j at time t
$T[D]$	Total completion time (makespan)
α_j	Allocation ratio of PEs for stream j,

$$\gamma_j(t) = \frac{\lambda_j(t)}{\alpha_j K} \tag{2}$$

where the α_j means the average allocation ratio of the PEs in stream *j*. On the time *t*, then, we can obtain the estimated completion time of each stream by:

$$T[D_j(t)] = \frac{D_j(t)}{\lambda_j(t)} \tag{3}$$

The estimated completion time is different to the checking point. Using Eq. (3), we can find the newly calculated allocation ratio as being:

$$\forall \alpha_j, \alpha_j = \frac{T[D_j(t)]}{\sum_{j \in J} T[D_j(t)]}. \tag{4}$$

Based on Eq. (4), The APEPP optimizer in the DCS balances the PE allocation ratio. Even though the APEPP algorithm assumes the fixed size of PEs, it is easily applicable to the elastic environment by increasing or decreasing the available PEs in the run-time. As a result, the proposed APEP is presented in Algorithm 1, in which the *K* and *J* are the total number of PEs and the total number of dataset respectively. In Step 1, it performs the initial allocation ratio (α_j) by dividing the total PEs into the total dataset such as $K_j = a_j K$. After finishing the initial allocation of the PEs, it determines the next checkpoint for performing the APEP in the run-time. On the other hand, the APEP algorithm is described in Step 2. When arriving at the check point, the APEP calculates the unit processing rate in Eq. (2), the estimated completion time in Eq. (3), and finally the allocation ratio in Eq.(3) sequentially until all of the data processing is finished.

Algorithm 1. APEP Algorithm

Step. 1 : Initial Scheduling Stage	
1	*Set **K** and **J***
2	*Allocation ratio for each dataset **j**, **a_j ← 1/J***
3	*The number of PE allocation for each dataset **j**, **$K_j = a_j * K$***
4	*Allocate **K_j** to each dataset **j***
5	*Set the next check point for APEP*
6	*Go to the Step 2*
Step. 2 : APEP Scheduling Stage	
1	**If** (arrives the check point && the data processing is not completed) **then**
2	*Calculate unit processing rate for each dataset based on (2)*
3	*Calculate estimated completion time based on (3)*
4	*Calculate allocation ratio based on (4)*
5	*Allocate **K_j** to each dataset **j***
6	*Set the next check point for APEP*
7	*Go to the Step 2*
8	**Else** *Exit*
	EndIf

4. EXPERIMENTAL RESULTS

4.1 A Sequencing Alignment Application for Experiment

Fig. 4 shows a pipeline application that was used in a 2009 genome research article. In the figure, the *wet-lab work process* retrieves the personal genome information and makes format to sequence and the *dry-lab work process* performs various pipelined jobs such as indexing, alignment, pairing, formating, and reformatting for extracting the abnormal state in the sequence. We utilize the Burrows-Wheeler Aligner (BWA) application for the workflow, which is an efficient program that aligns relatively short nucleotide sequences against a long reference sequence such as the human genome. It implements the two algorithms—the bwa-short and the bwa-sw [8]. On the workflow tasks, the *align* finds the suffix array (SA) coordinates of the input data with reference data. We prepared three different dataset (input data) with two kinds of reference data as shown in Table 2. The size of the input dataset refers to the problem size. We made six cases by combining the two reference data above and three input reads.

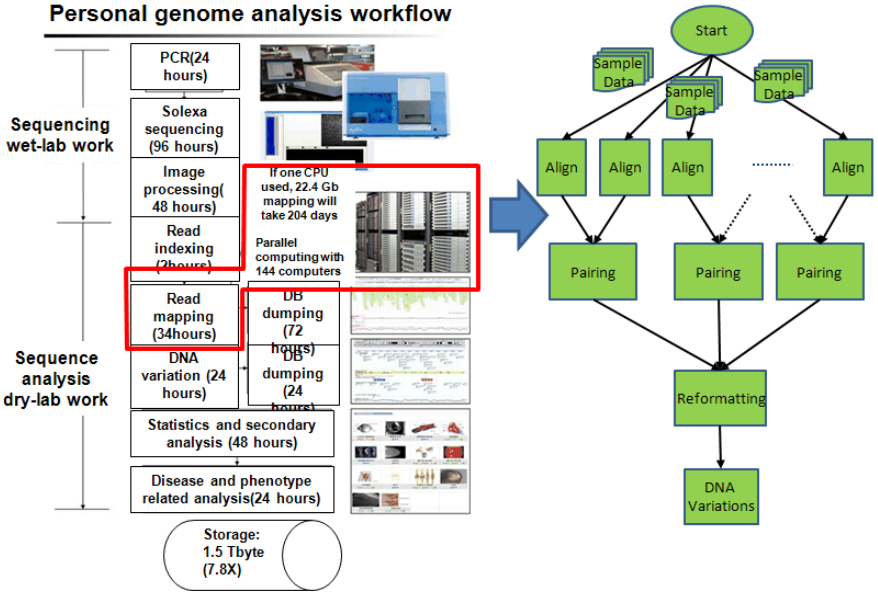


Fig. 4. A pipeline application used in a 2009 genome research article [9]

Table 2. Experimental Parameters

Index (dataset)	File Name	Size	Total Sequences
Reference	chr22.fa	0.8GB	-
Dataset1	R10_75X75_200_s_1_1_sequence.txt	2.01GB	40,658,816
Dataset2	R14_75X75_200_s_4_1_sequence.txt	2.11GB	42,423,298
Dataset3	R16_75X75_200_s_2_1_sequence.txt	2.63GB	52,789,906
Dataset4	R21_s_1_1_sequence.txt	3.48GB	77,500,842

4.2 Experimental Environment

As shown in Fig. 5, we utilized a total of 32 PEs using 8 machines with 4 cores (Intel(R) Xeon(R) CPU W3520 2.67GHz) and a single data distribution server. For the resource configuration, we conducted two different evaluations to validate the effect of the continuous data distribution scheme and the effect of the adaptive resource-provisioning scheme, respectively.

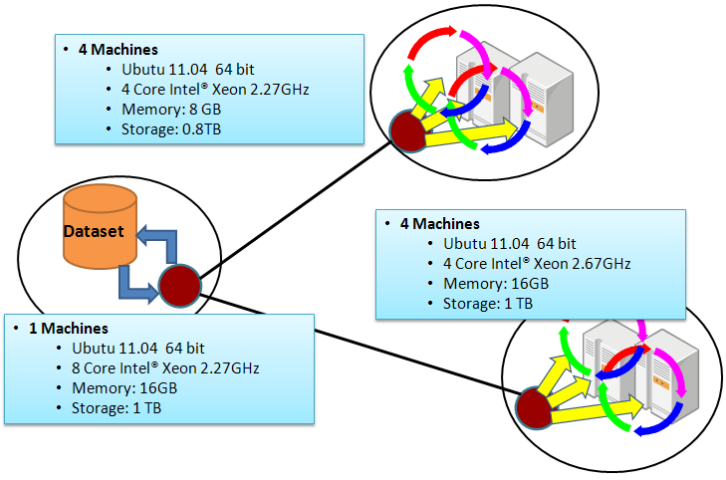
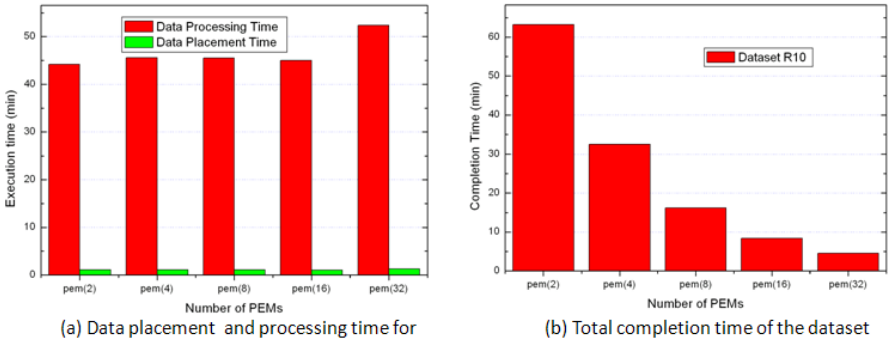


Fig. 5. An experimental environment and resource configurations

4.3 The Effect of the Continuous Data Distribution Scheme

In the first experiment, we conducted a single dataset (dataset1) and we divide the total sequences (40,658,816) to 262,144 for distributing the dataset. For evaluating the effect of the continuous data distribution scheme we increased the degree of the parallelism as (2,4,8,16,32). Fig. 6(a) shows the data placement time and the data processing time for each chunk. Even though the numbers of the processors (PEM) increased, the data placement time did not increase,



(a) Data placement and processing time for each chunk

(b) Total completion time of the dataset

Fig. 6. The Execution time of each chunk and the totoal completion time of the total dataset with different parallelism

as it is ignorable when compared to the data processing time. In addition, Fig. 6(b) shows the almost linear scalability in accordance with increase in the number of processors (PEMs)

4.4 The Effect of Adaptive Resource Provisioning

The second experiment that we conducted was with 4 different kinds of datasets and we applied three different scheduling policies. We compared the policies, which are listed below.

- Static/fixed provisioning- static allocation with a fixed number of processors.
- Static/proportional provisioning- proportional allocation with the dataset size.
- Adaptive provisioning - re-allocation based on the estimated completion time.

The static/fixed scheduling simply allocates the same number of processing elements. Second, the static/proportional one schedules the data in proportion to the dataset size. Finally, we applied our proposed adaptive scheme. As shown in Fig. 7, the proposed adaptive scheduling scheme shows a small variation in the completion time among all of the datasets, as compared to the other two scheduling policies. For each policy, the largest completion time among all datasets as presented as 41 minutes, 35 minutes, and 28 minutes, respectively. Since the total completion time was determined by the largest completion time. As a result, the proposed adaptive scheduling scheme minimized the total completion time, even though the completion time of R10 and R14 are larger than those of the static/fixed scheme.

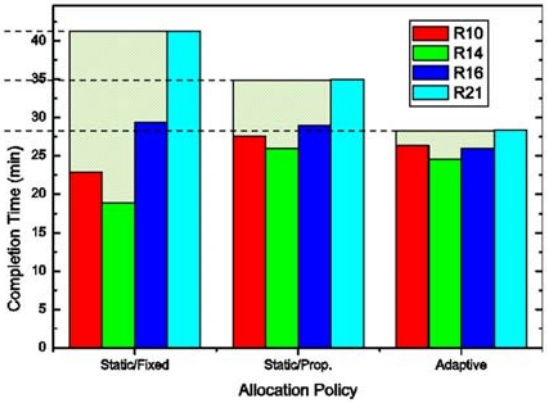


Fig. 7. Comparision of the total completion time for the three different resource-provisioning policies

5. CONCLUSION

In this paper, we proposed a data distribution and collection service (DCS), which decoupled data transfer and data processing procedures. The DCS is possible to eliminate the time to transfer the input and output dataset from the local storage to cloud farms. Particularly, the DCS autonomously regulates the data transfer process by synchronizing the data transfer process and data processing procedure, respectively. In the proposed DCS, we further exploited the adaptive resource-provisioning scheme for optimizing the allocation ratio of the computing elements to the dataset in order to minimize the total makespan under resource constraints. We examined the

experiments with a well-known sequence alignment algorithm and the results showed that the proposed scheme was efficient for the cloud environment.

REFERENCES

[1] C. H. Youn, E. B. Shim, et al, *A Cooperative Metabolic Syndrome Estimation with High Precision Sensing Unit*, IEEE Transaction on Biomedical Engineering, Vol. 58, No. 3, pp809-813, March 2011.

[2] G. O. Young, *Synthetic Structure of Industrial Plastics*, in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp.1564.

[3] D. Sulakhe, M. D'Souza, M. Syed, A. Rodriguez, Y. Zhang, E. Glass, M. Romine, and N. Maltsev, *GNARE - A Grid-based Server for the Analysis of User Submitted Genomes*, Nucleic Acids Res. vol. NAR-00335-Web-B-2007.Rl, 2007.

[4] Maltsev N, Glass E, Sulakhe D, Rodriguez A, Syed MH, Bompada T, Zhang Y, D'Souza M. *PUMA2 Grid-based High-throughput Analysis of Genomes and Metabolic Pathways*. Nucleic Acids Res. Vol.34, 2006.

[5] C. S. Schuster, *Next-generation Sequencing Transforms Today's Biology*, Nature Methods, Vol.5, 2008.

[6] Apache Hadoop. http://hadoop.apache.org/

[7] Li, H., Homer, N. *A Survey of Sequence Alignment Algorithms for Next-Generation Sequencing*, Briefings in Bioinformatics 11(5) September, 2010.

[8] Li H. and Durbin R. *Fast and Accurate Long-read Alignment with Buffows-Wheeler Transform*, Bioinformatics, Epub, 2010.

[9] Ahn. S. M, et.al, *The first Korean Genome Sequence and Analysis: Full Genome Sequencing for a Socio-ethnic Group*, Genome Research, 2009.

[10] Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. *Basic Local Alignment Search Tool*, Journal of Molecular Biology 215(3) October 1990.

[11] Kepler Project. https://kepler-project.org/

[12] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, *The Sequence Alignment/Map format and SAMtools*, 1000 Genome project data processing subjroup, 2009.

[13] DNA Sequencing, http://en.wikipedia.org/wiki/DNA_sequencing

[14] S. Deng, *A Study on Policy Adjuster integrated Grid Workflow Management System*, MS Thesis, Information and Communications University, Korea, 2008.

[15] C. H Han, C. H Youn, W. Jung, *Web-Based System for Advanced Heart Disease Identification Using Grid Computing Technology*, 21st IEEE International Symposium on Computer-Based Medical Systems, 2008.

[16] C. H Youn, B. Kim, and E. B Shim, *Resource Reconfiguration Scheme Based on Temporal Quorum Status Estimation for Grid Management*, IEICE Trans. Comm. E88 (11) (2005) 4378-4381.

Byungsang Kim

Byungsang Kim received an MS degree from KAIST in 2004 in information and communicatons engineering and a BS degree from Dongkuk University in 2002. He was a researcher in Korea e-Science research group in KISTI, Korea from 2006 to 2009. He is currently a Ph.D student in the Dept. of information and communications engineering in KAIST, Korea. His research is in the field of resource management with scalability in elastic computing environments such as grid or cloud computing and the large-scale data analysis platform and big data analytics.

Chan-Hyun Youn

Chan-Hyun Youn received the B.Sc and M.Sc degrees in Electronics Engineering from Kyungpook National University, Daegu, Korea, in 1981 and 1985, respectively. He also received a Ph.D. in Electrical and Communications Engineering from Tohoku University, Japan, in 1994. Since 2009, he has been a professor at Department of Electrical Engineering in KAIST, Daejeon, Korea. He also was a Dean of Office of Planning Affairs and a Director of Research and Industrial Cooperation Group at former Information and Communications University, in 2006 and 2007. He was a Visiting Professor at MIT in 2003 and has been engaged in the development of Physio-Grid system with Prof. R.G. Mark`s Group in LCP (Laboratory for Computational Physiology) of MIT since 2002. He also is a Director of Grid Middleware Research Center at KAIST. Where, he is developing core technologies that are in the areas of mobile cloud, mobile collaboration system, Internet computing workflow management, distributed network architecture, communication middleware, advanced e-Healthcare system, e-Health application services and others. Currently, he is serving the Editor-in-Chief of KIPS (Korea Information Processing Society), and an Editor of Journal of Healthcare Engineering (U.K.), and served head of Korea branch (computer section) of IEICE, Japan (2009, 2010). He is a member of IEEE, KICS and IEICE, respectively.

Yong-Sung Park

Yong-Sung Park is a researcher of the Grid Middleware Center, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. He received the B.S. degree in Information and Communication Engineering from Daejeon University, Daejeon, Korea in 2006. Currently, he is a M.S. candidate in Department of Information and Communication Engineering at Chungnam National University, Daejeon, Korea. Before joining the Grid Middleware Center of KAIST, he worked for KUBIX as a Assistant Manager of Network Division.

Yonggyu Lee

Yonggyu Lee received his BSEE degree from Chonbuk National University in 1996, and MSEE and Ph.D. degrees from Korea Advanced Institute of Science and Technology (KAIST), Korea in 2000 and 2008, respectively. From 1996 to 1997, he was at Multimedia Laboratory of Hyundai Electronics Industries, Korea, and was a main member of MPEG programming and circuit board designers for CD/DVD-players. From 1998 to 1999, he was at Tera Optical Team of Electronics and Telecommunications Research Institute, Korea as an Adjunct Researcher and was a main member of optical switch interfacing programmers. From Jan. 2001 to Feb. 2001, he worked for Nortel Networks, Ottawa, ON., Canada as a Visiting Researcher and was a member of network simulation programmers for future Peta networks. From March 2008 to Feb. 2009, he worked for Optical Internet Research Center (OIRC), Korea as a Post-Doc. Now he is working for KAIST as a research fellow at Grid Middleware Center (GMC). His interests include cloud computing, super computing, cloud resource broker and scheduler, next generation Internet, and QoS guaranteed packet network.

Wan Choi

Wan Choi is received his BS in electric engineering from Kyungpook National University in 1981. He also received his MS in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea in 1983. In 1985, he joined ETRI, Daejeon, Rep. of Korea, and works for the Cloud Computing Department as a director of the researching staff. He is currently the project leader of the development of supercomputing system for the genome analysis. His research current areas of research include supercomputing, cloud computing, and big data analysis platform.