A Web-based Toolkit for Scheduling Simulation Using GridSim

Oh-han Kang Sang-seong Kang
Dept. of Computer Education, Andong National University, South Korea
ohkang@andong.ac.kr edukang@andong.ac.kr

Abstract

We implemented a web-based grid scheduling platform, which can model a system and simulate a scheduling method in grid computing. The scheduling platform is able to perform resource modeling, task modeling, algorithm compiling, simulation, and performance evaluation rapidly in a web environment. The implemented scheduling platform can be applied as a foundation for grid research and can be used to analyze the efficiency of the scheduling algorithm.

1. Introduction

There are several tools that are support applications for a scheduling simulation in grid computing environments. To simulate the scheduling algorithm with the application of current grid scheduling toolkit, several procedures such as the establishment of an appropriate development environment, the analysis of the source code, the programming for resource modeling and operational modeling, and the practice of the scheduling algorithm are required. These procedures incite repetitive operations for researchers and diminish the efficiency of the scheduling algorithm. Also, Australia's GRIDS research center developed Visual Modeler(VM)[1], a GUI-based modeling tool, and also enabled simple drawing up of source codes in resource modeling and application modeling, which the research center developed for GridSim[2]. However, VM merely facilitated the modeling procedures, and showed no improvements in the functions such as simulation, analysis of capacity, the storage and management of resource and application data.

In this paper, we suggested and actualized Web-based Grid Scheduling Platform (WGridSP), which designed a system, and simulated a scheduling method in a grid computing environment. The actualization of WGridSP enables, with the help of a grid computing environment, the modeling of scheduling algorithm and simulation. WGridSP is composed of testbed manager, application manager, algorithm compiler, scheduling simulation, and efficiency-analyzing module. WGridSP can carry out resource modeling, task modeling, algorithm compiling, simulation, and rapid evaluation in a web environment. Therefore, WGridSP furnishes researchers with the environment where they can concentrate on the actualization of algorithm and the analysis of results. The comparison of GridSim, a typical Java-based scheduling toolkit, and WGridSP is illustrated in Table 1.

Table 1. Comparison of GridSim and WGridSP

		GridSim	WGridSP
Compile/Execution environment		Mandatory	Optional
Toolkit structure analysis		Mandatory	Mandatory
Resource/ Application modeling	Tool	Text editor	Web browser
	Style	Java source	Database
	Compile	At Shell	Optional
Implementation of scheduling algorithm	Tool	Text editor	Web browser
	Style	Java source	Core algorithm
	Compile	Shell	Server
Simulation	Exec	Shell	Web
	Result	Console/Log	Web page
Performance evaluation	Tool	Text editor	Web browser
	Exec	Shell	Web
	Param	Source code	Web input
	Result	Console/Log	Web page/Graph

2. The Structure of WGridSP

WGridSP has utilized GridSim, a Java-based grid scheduling toolkit as a simulation tool. GridSim, through its choice of various variables in global grid environment and the selection of SimJava[3] package as a foundation which is frequently preferred in the simulation of distributed environment, succeeds in becoming the grid scheduling toolkit with high practicality and stability. WGridSP not only functions as an interface between user and GridSim by the means of the web, but also makes recycling possible with the management of resource modeling, application modeling materials and the scheduling algorithm.

Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06)
0-7695-2694-2/06 $20.00 © 2006 **IEEE**

2.1 Testbed manager

Testbed manager allows users to conduct resource modeling easily and rapidly. Especially, resource information planned by users is stored in the database through the management of each testbed, and used for algorithm simulation and performance evaluation. Testbed is composed of one or more resources, which, in turn, is composed of machines. These machines are composed of more than one processor. Generally, resources composed of more than one machine are regarded as the "Cluster system". Properties assigned to WGridSP's resources are illustrated below.

◎ Strategy for assigning processors: It is the property determining how to assign processors when the number of processors occupied by the resource is less than the number of operations waiting to be processed. In WGridSP, one can choose between the time sharing method, which allocates and conducts more than two works to the specific processor concurrently and the space sharing method, which processes later work after earlier assigned work is finished.

◎ Transmission speed: It is a transmission speed of which a specific resource is connected to the network. The size of the input data necessary to process the operation influences the size of the outcome data after processing and the amount of time required for the process

◎ Cost: Defining the cost of a specific resource can be utilized in the scheduling algorithm which considers cost.

◎ Processor conducting ability: It indicates the ability to process an operation and is expressed in MIPS units. It influences the length of the operation and the amount of time required for processing.

◎ The number of processors: It indicates the number of processors possessed by the machine. More processors enable a greater number of operations that can be managed.

2.2 Application manager

The application manager supports modeling of application operations. The application information designed by a user is stored in a database and, thereafter, is used for simulation. Each application is composed of one or more tasks and these tasks retain the following properties.

◎ Length: The task's required amount of time of CPU. Bigger values indicate a longer amount of processing time

◎ Input size: It indicates the size of the data needed to process the task. Because the data need to be transmitted to an assigned resource, input size affects the amount of time needed to transmit the task to resources.

◎ Output size: It indicates the size of the data produced by resources after the processing task. Also, it influences the required time needed to transmit results from resources to scheduler.

2.3 Scheduling simulation

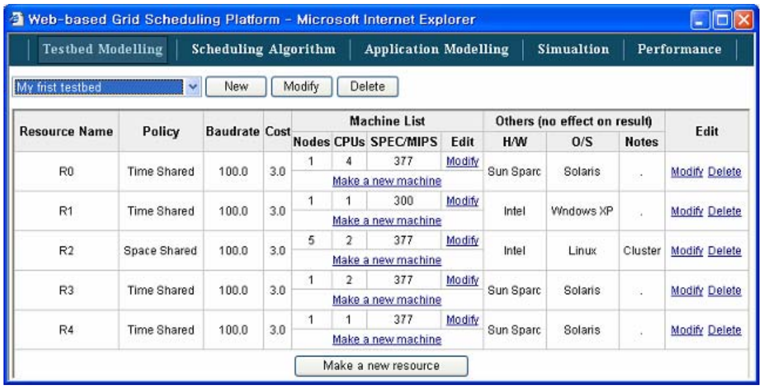
WGridSP demands the core part of scheduling algorithm. The source code which garners necessary grid information is produced automatically, immediately before compiling, and these compiling results are transmitted to users to fix the possible errors. The selection of the testbed, application, algorithm required for scheduling simulation can be confirmed through the web-browser. By validating which resources of Gridlet are assigned to which period of time, the analysis of a detailed operational situation is possible.

In the evaluation of the scheduling algorithm, the testbed for evaluation, the property limits of the task, and the choice over more than one scheduling algorithm can verify the change in the time needed to finish the work according to the amount of work. The performance time and the size of input and output data produce a random number in uniform distribution or normal distribution. As a result, to obtain the exact measured value, the average value related to the designated number of repetitions should be acquired, to eventually get the final performance time.

3. Experiments and Results

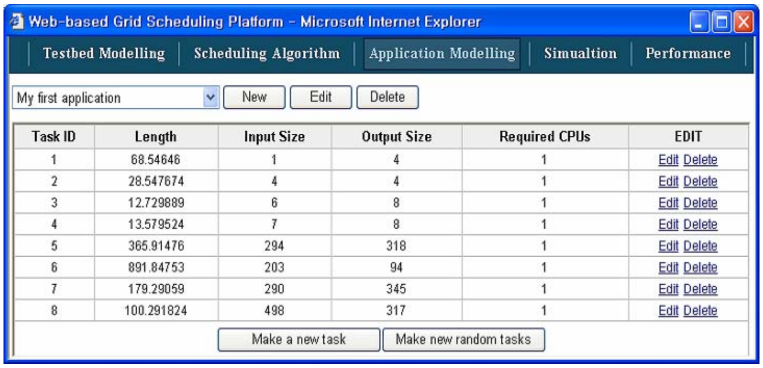
As illustrated in this paper, the usages of WGridSP to simulate grid scheduling are shown in the following examples. WGridSP is composed of testbed manager, application manager, algorithm compiler, scheduling simulation, and efficiency-analyzing module. Testbed manager produces a new testbed for simulation and composes resources included in the testbed. In this example, two resources are defined and details are demonstrated in Fig. 1.

Using the application manager, users can directly assign each variable of Gridlet, which is the list of processes needed to be managed. Moreover, users can assign the specific range, and use or produce random numbers. Fig. 2 shows how 10 variables of Gridlets are defined by using random numbers.



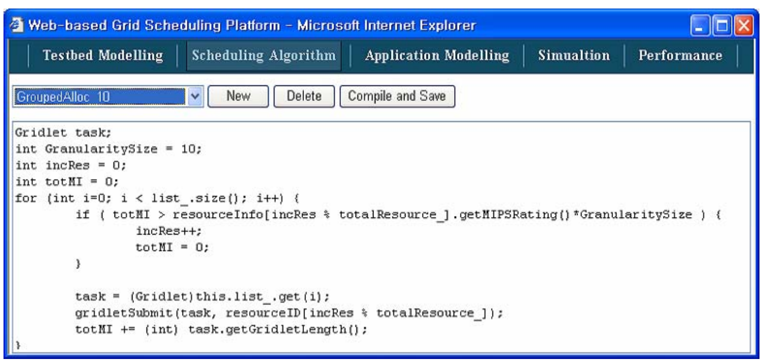
Resource Name	Policy	Baudrate	Cost	Machine List				Others (no effect on result)			Edit
				Nodes	CPUs	SPEC/MIPS	Edit	H/W	O/S	Notes	
R0	Time Shared	100.0	3.0	1	4	377	Modify Make a new machine	Sun Sparc	Solaris	.	Modify Delete
R1	Time Shared	100.0	3.0	1	1	300	Modify Make a new machine	Intel	Wndows XP	.	Modify Delete
R2	Space Shared	100.0	3.0	5	2	377	Modify Make a new machine	Intel	Linux	Cluster	Modify Delete
R3	Time Shared	100.0	3.0	1	2	377	Modify Make a new machine	Sun Sparc	Solaris	.	Modify Delete
R4	Time Shared	100.0	3.0	1	1	377	Modify Make a new machine	Sun Sparc	Solaris	.	Modify Delete

Fig. 1. The example of testbed modeling



Task ID	Length	Input Size	Output Size	Required CPUs	EDIT
1	68.54646	1	4	1	Edit Delete
2	28.547674	4	4	1	Edit Delete
3	12.729889	6	8	1	Edit Delete
4	13.579524	7	8	1	Edit Delete
5	365.91476	294	318	1	Edit Delete
6	891.84753	203	94	1	Edit Delete
7	179.29059	290	345	1	Edit Delete
8	100.291824	498	317	1	Edit Delete

Fig. 2. The example of application modeling




```
Gridlet task;
int GranularitySize = 10;
int incRes = 0;
int totMI = 0;
for (int i=0; i < list_.size(); i++) {
        if ( totMI > resourceInfo[incRes % totalResource_].getMIPSRating()*GranularitySize ) {
                incRes++;
                totMI = 0;
        }

        task = (Gridlet)this.list_.get(i);
        gridletSubmit(task, resourceID[incRes % totalResource_]);
        totMI += (int) task.getGridletLength();
}
```

Fig. 3. Compiling of scheduling algorithm

The scheduling algorithm in Fig. 3 is showing how Grouping-Based Scheduling[4] with 10 Granularity Size is applied to WGridSP environment.

The simulation of the selected testbed, application, algorithm that are defined above indicates assigned results and execution results as shown in Fig. 4. When [My first application] is applied to [My first testbed] as SequentialAlloc scheduling algorithm, the total amount of time needed to end all tasks is 22.52 as shown in Fig. 4. The process indicating how resource is transmitted to each task to perform and acquire simulation results is expressed in the chart, which is classified by a different period.

Fig. 5 is the result of evaluating three scheduling algorithms in [My first testbed]. In the graph, the X-axis indicates the number of tasks, while the Y-axis indicates the amount of time required to finish all work. The period of work, and the size of input and output are set up to comply with normal distribution, and the graph shows the average value of ten simulation results. The evaluation result demonstrates grouping-based scheduling[4] with 10 granularity size as the most competent.

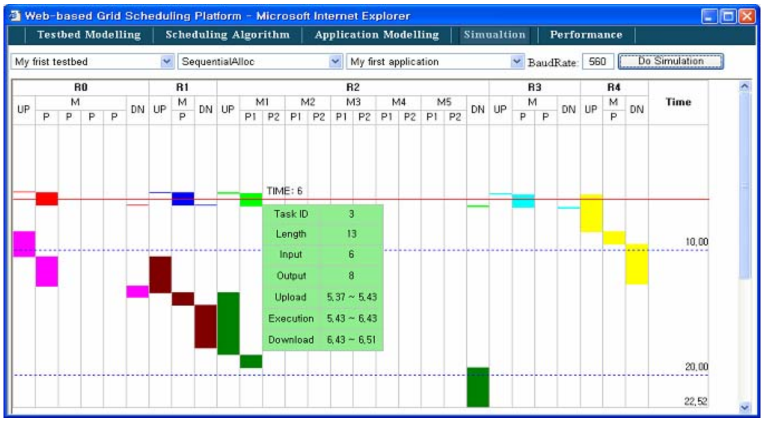


Fig. 4. Simulation results

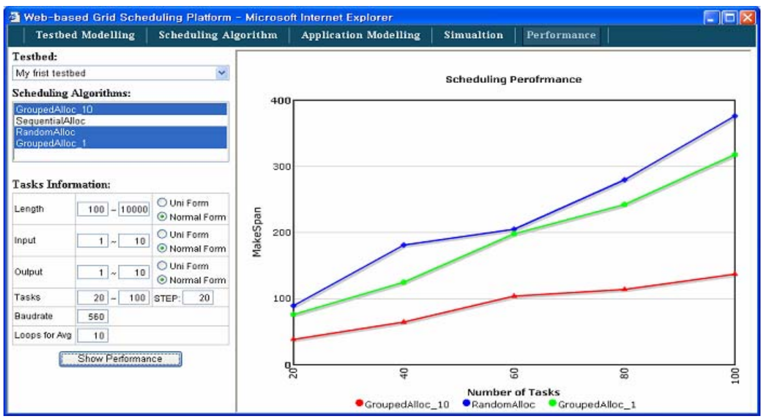


Fig. 5. Scheduling performance

4. Conclusion

In this paper, GridSim, a Java-based grid scheduling toolkit, is utilized to present web-based Grid Scheduling Platform(WGridSP) which can rapidly execute resource modeling, operational modeling, algorithm compiling, and simulation in a web environment. WGridSP can save a considerable amount of time and effort, due to its ability to simulate grid scheduling in a Web environment.

5. Acknowledgments

This work was supported by grant No. B1220-0601-0018 from the University Fundamental Research Program of the Ministry of Information & Communication in Republic of Korea.

6. References

[1] Anthony Sulistio, Chee Shin Yeo, and Rajkumar Buyya, "Visual Modeler for Grid Modeling and Simulation (GridSim) Toolkit", ICCS 2003, LNCS 2659, pp 1123-1132, 2003.

[2] R. Buyya, and M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing", The Journal of Concurrency and Computation, Vol. 14, pp. 1175-1220, 2002.

[3] F. Howell and R. McNab, "SimJava: A Discrete Event Simulation Package for Java with Applications in Computer Systems Modeling", Proc. of 1st Int. Conference on Web-based Modeling and Simulation, Society for Computer Simulation, 1998.

[4] N. Muthuvelu, J. Liu, Nay Lin Soe, S. Venugopal, A. Sulistio and R. Buyya, "A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids", AusGrid2005, Conferences and Practice in Information Technology, Vol. 44, 2005.