Mobile Resource Reliability-based Job Scheduling for Mobile Grid

Sung Ho Jang[1] and Jong Sik Lee[2]
[1] Department of Information Engineering, Inha University
Incheon 402-751, South Korea
[e-mail: ho7809@hanmail.net]
[2] School of Computer Science and Information Engineering, Inha University
Incheon 402-751, South Korea
[e-mail: jslee@inha.ac.kr]
*Corresponding author: Sung Ho Jang

Received October 4, 2010; revised November 19, 2010; accepted December 11, 2010; published January 31, 2011

Abstract

Mobile grid is a combination of grid computing and mobile computing to build grid systems in a wireless mobile environment. The development of network technology is assisting in realizing mobile grid. Mobile grid based on established grid infrastructures needs effective resource management and reliable job scheduling because mobile grid utilizes not only static grid resources but also dynamic grid resources with mobility. However, mobile devices are considered as unavailable resources in traditional grids. Mobile resources should be integrated into existing grid sites. Therefore, this paper presents a mobile grid middleware interconnecting existing grid infrastructures with mobile resources and a mobile service agent installed on the mobile resources. This paper also proposes a mobile resource reliability-based job scheduling model in order to overcome the unreliability of wireless mobile devices and guarantee stable and reliable job processing. In the proposed job scheduling model, the mobile service agent calculates the mobile resource reliability of each resource by using diverse reliability metrics and predicts it. The mobile grid middleware allocated jobs to mobile resources by predicted mobile resource reliability. We implemented a simulation model that simplifies various functions of the proposed job scheduling model by using the DEVS (Discrete Event System Specification) which is the formalism for modeling and analyzing a general system. We also conducted diverse experiments for performance evaluation. Experimental results demonstrate that the proposed model can assist in improving the performance of mobile grid in comparison with existing job scheduling models.

Keywords: Mobile grid, job scheduling, agent system, reliability prediction

This work was supported by INHA University Research Grant.

DOI: 10.3837/tiis.2011.01.005

1. Introduction

Grid computing [1][2] has been noticed and developed by lots of researchers as the next generation computing platform during the last decade. Grid computing is a solution for large-scale computing problems which cannot be solved by existing supercomputers and is applied to diverse fields of biotechnology, aerospace engineering, e-business, and so on. Traditional grid computing systems [3] are able to provide huge computing capabilities to grid users as integrating static grid resources such as PCs and server systems through wired networks. However, the current grid environment is being changed from wired networks with static resources to wireless networks with dynamic resources by the development of communications equipment and network technology [4]. Mobile grid, a new grid computing platform, is being realized by the change in a grid environment.

Mobile grid [5] is a compound word of grid computing and mobile computing and inherits advantages and capabilities of traditional grids. The purpose of mobile grid is to improve the performance of grids by integrating mobile devices into a wire network-based grid environment and thus mobile grid contains both static grids and wireless mobile sites. In a mobile grid environment, dynamic mobile resources, such as smart phones and PDAs, are utilized to provide grid services, as well as conventional grid resources [6]. Unlike traditional grids, mobile grid can be applied to knowledge-intensive and adaptive application areas like seismological observation, meteorological observation, and e-Health by the advantages of mobile devices. As the capabilities of mobile devices have been substantially improved, a lot of grid communities have been interested in mobile grid. There is very much a work for mobile grid in progress by lots of researchers and organizations all over the world. Previous studies for mobile grid are biased towards using a mobile device as an interface. However, mobile devices will be utilized as grid resources as their performance is being improved and the number of mobile device users is really growing. Therefore, an effective job scheduling model is necessary and it is decisive of the whole grid performance. Existing grid infrastructures and scheduling models are unsuitable for a mobile grid environment since they do not consider mobile devices as valid and reliable grid resources. Grid middleware such as Globus [7] is too heavy and large to be installed on thin mobile devices, which do not have enough computing power, memory, and storage. Existing grid scheduling models are also inappropriate for a wireless network environment because they have no consideration for mobility and connectivity of mobile devices.

Therefore, this paper proposes the mobile resource reliability-based job scheduling model for mobile grid and presents the mobile grid architecture to integrate mobile resources into existing grid sites. In this paper, we lay emphasis on the factor that the connection state of mobile devices is changed as their owner moves. The proposed model measures diverse reliability metrics of each mobile resource, such as its connectivity and availability, and calculates its mobile resource reliability. The proposed model also predicts the mobile resource reliability for job allocation by a statistical prediction method described in Section 3.2. The predicted mobile resource reliability is used for job allocation. The proposed model can reduce job latency and job losses and improve utilization and throughput by allocating jobs to mobile resources with high mobile resource reliability.

This paper is organized as follows. Section 2 discusses grid scheduling and grid reliability as backgrounds of this paper. Section 3 proposes the mobile resource reliability-based job scheduling model for mobile grid. Section 4 demonstrates the effectiveness and efficiency of our prediction method and job scheduling model with experiment results. Finally, Section 5 concludes this paper.

2. Related Works

2.1 Grid Scheduling

Mobile grid needs a job scheduling policy to effectively allocate jobs to proper resources since an effective job scheduling policy can improve throughput and service time of mobile grid. So far, existing scheduling models [8][9] for traditional grids have been applied to mobile grid projects. These scheduling models are inadequate to be applied to mobile grid in spite of their simple structure and time complexity. Contrary to a traditional grid environment, a mobile grid environment is based on wireless networks and dynamic resources with mobility. It causes serious job losses when mobile resources are disconnected to WLANs or move to out of WLANs.

There have been several works related to job scheduling for mobile grid. The response time-based job scheduling model was proposed in [10]. In this model, a scheduler obtains the response time of all mobile resources and allocates a job to a mobile resource with the fastest response time. M. Ballette, A. Liotta, and S. M. Ramzy proposed the execution time prediction model for job scheduling in a mobile grid environment [11]. The execution time of a mobile resource is the sum of its processing time and data transmission time, which is effected by its predicted disconnectivity. Jobs are allocated to mobile resources in increasing order of execution time. These scheduling models provide relatively high throughput and response time, but concentrate jobs on mobile resources with high processing and data transfer speeds. P. Ghosh, N. Roy and S.K. Das proposed the cost effective job scheduling model based on a predetermined pricing strategy for mobile grid [12]. In this model, a scheduler decides the price per unit resource by the game theory and allocates jobs to mobile resources with cheaper cost. This model can minimize the total price, which a system pays to resources to complete jobs, but cannot assure the QoS of a grid computing service.

This paper proposes the mobile resource reliability-based job scheduling model to solve these problems. The proposed model overcomes the unreliability of mobile resources and prevents resources from job losses by selecting resources with high mobile resource reliability.

2.2 Grid Reliability

Mobile grid is fundamentally necessary to guarantee the reliability of mobile resources and the stability of networks for effective grid services because the reliability of mobile resources is closely related to their connectivity and mobility. For example, mobile resources, such as, Smart-phone and PDA, use 3G or Wi-Fi for networking. If a mobile resource is connected to 3G and its owner goes underground, the mobile resource is liable to disconnection. Also if a mobile resource is connected to Wi-Fi and its owner gets out of a Wi-Fi zone, the mobile resource cannot process a job normally.

Therefore, this paper focuses on grid reliability. The importance of grid reliability has been embossed by a lot of research for grid computing. A lot of studies on grid reliability has been progressed for network reliability [13][14] and resource reliability [15][16]. In some studies, grid reliability has been measured or analyzed by network reliability. In a grid environment, network reliability is defined as the probability for all of the grid computing programs to be executed successfully. The network connection can be failed due to excessive queuing delays or link failures. In this case, distributed programs cannot be normally executed at the time of disconnection and network reliability decreases. In [17], network reliability in grid computing is defined as grid program reliability, which is the probability of successful execution of the given grid application running on multiple grid resources in a grid computing system. For measuring network reliability, various metrics, such as congestion probability and link availability, are used in earlier studies. In other studies, resource reliability assures that grid resources can quickly process jobs for grid users. Namely, resource reliability represents how fast and stably grid resources can process jobs. If we guarantee resource reliability, we can improve the computing performance of the whole grid and increase the utilization of grid applications. In [18], resource reliability in grid computing is defined as grid service reliability, which is the probability that requested resources by grid applications are matched correctly and those resources execute the grid applications quickly.

Most of the researches on grid reliability only focus network reliability to execute jobs and receive their result without failure or resource reliability to process jobs more accurately and quickly. However, both of network reliability and resource reliability should be considered in a mobile grid environment to ensure success in processing jobs with a high level of QoS. Also, typical characteristics of a mobile grid environment are mobility, portability and wireless communication. These elements influence resource reliability and network reliability. Therefore, we proposed the mobile resource reliability that considers both network reliability based on connectivity of mobile resources and resource reliability based on their dynamic information, such as response rate, CPU availability, and so on.

3. Mobile Resource Reliability-based Job Scheduling Model

3.1 Design of Mobile Grid Architecture

The architecture of traditional grids is not suitable for a mobile grid environment because it is still too much heavy and large to be implemented on mobile devices (Laptops, PDAs, mobile phones, and etc.), of which performance and capacity are limited. Therefore, we propose the mobile grid architecture for effective mobile resource management in this Section.

The mobile grid architecture is composed as shown in **Fig. 1**. As mentioned earlier, mobile grid includes mobile devices into grid sites as grid resources and users. Mobile grid has to provide not only the interconnection among static resources but also the interconnection between static resources and mobile resources. Mobile grid middleware focuses on interactions with heterogeneous mobile devices and job allocation unlike traditional grid middleware, which provides a number of services such as authorization, security, resource discovery, and collaboration [6][19][20]. The context-aware middleware [21] and topology-aware middleware [22] have been proposed for grid computing, but they are unsuitable for a mobile grid environment.

Therefore, we design the mobile grid middleware that contains basic functions of existing grid middleware. Our mobile grid middleware provides functions on communication with existing grid middleware and resource management so as to include mobile devices in grid sites as available grid resources. Mobile resources are connected to wireless networks, such as wireless local area networks and cellular networks, and stored in a grid resource pool to construct virtual mobile grid sites. The grid resource pool is located in grid middleware and managed by the resource management system of existing grid middleware like Condor-G [23].

In order to interoperate mobile grid resources and static grid resources, our mobile grid middleware and mobile service agent are composed as shown in **Fig. 2**. The mobile grid middleware installed on a gate way consists of the mobile communication broker, the mobile cluster manager, the mobile job coordinator, the mobile job queue, and the mobile resource database.

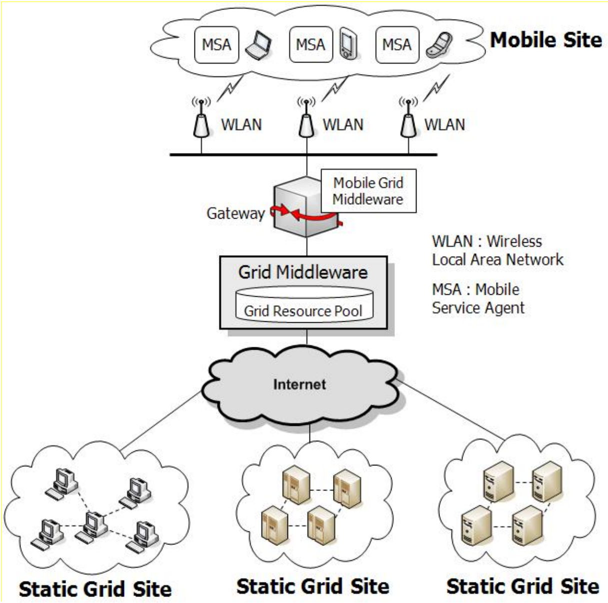

Fig. 1. Architecture of mobile grid.

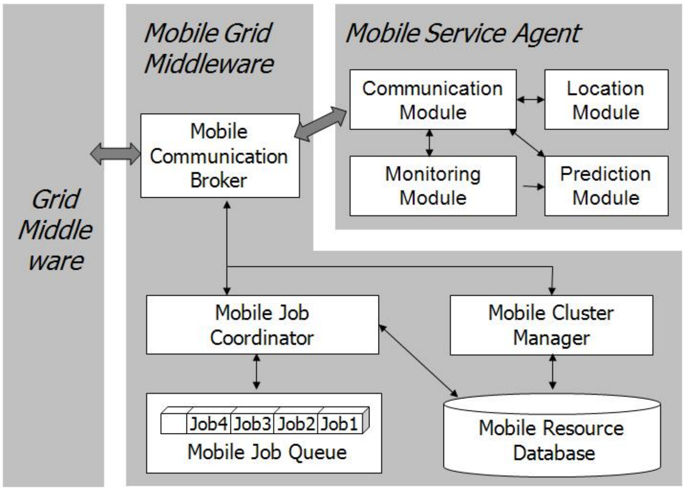

Fig. 2. Mobile grid middleware and mobile service agent.

- Mobile Communication Broker (MCB) – The mobile grid middleware provides basic functions for utilizing mobile devices as available grid resources. Therefore, the MCB supports communication between exiting grid and middleware mobile resources. The MCB uses SOAP messaging that is fundamentally provided in existing grid middleware as an interaction mechanism and connects mobile resources to established grid infrastructures through the Internet. The MCB also sends a SYN-ACK message to a mobile service agent.

- Mobile Cluster Manager (MCM) – The MCM offers cluster configuration data based on adjacency of mobile resources to reduce the complexity of resource selection and proffers advantages of the decentralized structure. Mobile resources in grid sites are grouped into several clusters.
- Mobile Job Coordinator (MJC) –The MJC takes charge of a job scheduler to distribute jobs. Contrary to existing job schedulers that allocate jobs based on computing capacities like CPU speed and node size, the MJC reads data from the mobile resource database and allocates jobs based on mobile resource reliability. If an error occurs in a mobile resource while a job is tried to be processed on the mobile resource, the MJC reallocates a failed job to an idle resource within the pertinent cluster. The job scheduling algorithm adopted in the MJC is presented in Section 3.2.
- Mobile Job Queue (MJQ) – The basic function of the MJQ is similar to that of existing job queuing systems such as PBS. When jobs are submitted from grid users, the MJC enqueues those to the MJQ sequentially. The MJC also dequeues a job from the MJQ and sends the job to a target resource at the time of job allocation.
- Mobile Resource Database (MRD) – The MRD is a database system that stores and manages resource data which includes static and dynamic information of mobile resources. The proposed job scheduling model uses dynamic information of mobile resources, such as resource availability and connectivity, as well as their static information. The dynamic information of mobile resources is updated and managed by the MRD. The MRD also performs queries requested from the MCM and/or the MJC and sends their results to the MCM and/or the MJC.

Mobile grid software installed on a mobile device is called ‘mobile service agent’. A mobile service agent is composed of four types of modules, which are the communication module, the location module, the monitoring module, and the prediction module.

- Communication Module (CM) – The CM takes charge of communication between the mobile grid middleware and a mobile service agent. This module is connected with the MCB of the mobile grid middleware through an asynchronous interface and sends an ACK message corresponding to a SYN-ACK message from the MCB. The CM also transfers location and condition data of a mobile resource to the MCB.
- Location Module (LM) – The LM measures the connectivity of a mobile resource by using its connection information. Assume that a mobile resource is connected to mobile grid through a WLAN. The location of the mobile resource is identified as a specific wireless access point (AP), which can be handed over to another AP according to its user movement. The AP is used for communication with the mobile grid middleware. The LM measures the AP and sends it to the CM. The LM also records the connection information of the mobile resource in a log file whenever the CM receives a SYN-ACK message from the MCB and decides the connectivity of the mobile resource based on its connection information.
- Monitoring Module (MM) – The MM monitors the current condition of a mobile resource and manages its condition data. Condition data is important to job allocation. Therefore, this module measures various parameters, such as response rate, recovery rate, CPU availability, and memory availability, and sends a condition message including these parameters to the MRD and the PM.
- Prediction Module (PM) – The PM provides an algorithm for predicting mobile resource reliability. PM calculates the current mobile resource reliability of a mobile resource by

using condition data received from the MM and predicts its expected mobile resource reliability after a job is allocated to the mobile resource at the scheduling cycle. This predicted mobile resource reliability is transmitted from the PM to the MRD for job allocation.

3.2 Mobile Resource Reliability-based Job Scheduling

This Section focuses on job scheduling among diverse issues of mobile grid mentioned in Section 3.1. Other issues related to communication service and location service will be covered in future work. In the existing grid environment, reliability is not an important reference item for job scheduling. However, when jobs are allocated to resources in a mobile grid environment, we should take account of the unreliability of mobile resources, which is increased by their mobility and connectivity. If a mobile resource is disconnected from wireless networks or lapses into overload, the mobile resource cannot process jobs normally. Therefore, we need to provide mobile resources with high reliability to grid users. In this Section, we describe the mobile resource reliability-based job scheduling method. **Table 1** shows variables used for the proposed job scheduling method in this section.

Table 1. Variables for mobile resource reliability-based job scheduling.

Variable	**Description**	**Variable**	**Description**
$P(c)$	The probability that a mobile resource is in “Connect”	R_{rec}	The recovery rate of a mobile resource
$\sum Tc$	The total time while a mobile resource stays in “Connect”	A_{cpu}	The CPU availability of a mobile resource
$\sum Td$	The total time while a mobile resource stays in “Disconnect”	U_{cpu}	The possessed CPU capacity of a mobile resource
$\sum Tr$	The total time while a mobile resource stays in “Reconnect”	T_{cpu}	The total CPU capacity of the mobile resource
R_{res}	The response rate of a mobile resource	A_{mem}	The memory availability of a mobile resource
n	The number of jobs allocated to a mobile resource	U_{mem}	The used memory capacity of the mobile resource
Tp	The processing time of a job allocated to a mobile resource	T_{mem}	The total CPU capacity of the mobile resource
Ti	The time required to transfer a job to a mobile resource	A_{batt}	The battery availability of a mobile resource
To	The time required to transfer the output of a job from a mobile resource	U_{batt}	The used battery capacity of the mobile resource
Ts	The time when the MJC sends a job to a mobile resource	T_{batt}	The total battery capacity of the mobile resource.
Tr	The time when the MJC receives the output from a mobile resource	$R(n)$	The network reliability of a mobile resource
ε, δ, κ, λ, γ	Optional weight values	$R(r)$	The resource reliability of a mobile resource
$R(mr)$	The mobile resource reliability of a mobile resource	ω	The smoothing constant to decide where to weight between present values and past values
α	The error set of predicted mobile resource reliability values	$fR(mr)$	The predicted mobile resource reliability value
β	The error set of measured mobile resource reliability values	$yR(mr)$	The measured mobile resource reliability value

Firstly, we measure the connectivity of each mobile resource for network reliability. The features of mobile grid, such as user mobility and asynchronous communication, cause the frequent disconnection and reconnection of mobile devices, therefore the connection state of a mobile resource can be classified into "Connect," "Disconnect," and "Reconnect" as shown in **Fig. 3**. "Connect" indicates when a mobile resource is normally connected with a wireless network. "Disconnect" indicates when the mobile resource is disconnected from the wireless network. "Reconnect" indicates when the mobile resource waits for its reconnection. Especially, "Reconnect" represents that a mobile resource tries to access to a wireless network and affects the recovery rate of the mobile resource.

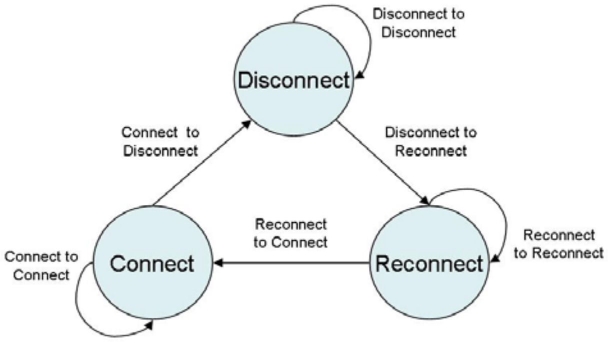

Fig. 3. Connection state transition diagram of a mobile resource.

In this paper, the connectivity of a mobile resource is the probability that the mobile resource is in "Connect" stably. We assume that a mobile resource in "Reconnect" is identical with that in "Disconnect" since both are impossible to normally process jobs. Therefore, $P(c)$ is:

$$P(c) = \sum Tc / (\sum Tc + \sum Td + \sum Tr) \tag{1}$$

Secondly, we measure the following five reliability metrics to calculate the mobile resource reliability of a mobile resource: response rate, recovery rate, CPU availability, memory availability, and battery availability.

Response rate represents how fast a mobile resource sends the result of job processing to the mobile job coordinator. The response rate is related to response time of mobile resources. The response time of a resource commonly means how long it takes to process a job by the resource. Therefore, we measure processing and transmission times of a job allocated to a resource and divide those times by the total time required to solve the job. We repeat the above steps for all jobs allocated to a resource and calculated the average value to measure the response rate of the resource as follows:

$$R_{res} = \frac{1}{n}\sum_{i=1}^{n}\frac{(Tp_i + Ti_i + To_i)}{(Tr_i - Ts_i)}, \qquad 0 \le R_{res} \le 1 \tag{2}$$

Recovery rate represents how long a mobile resource takes to recover from disconnection when it is disconnected and can be calculated as follows:

$$R_{rec} = \frac{\sum Tr}{\sum Td + \sum Tr}, \qquad 0 \le R_{rec} \le 1 \tag{3}$$

CPU availability represents how much the CPU capacity of a mobile resource is available and can be calculated as follows:

$$A_{cpu} = 1 - (U_{cpu} / T_{cpu}), \qquad 0 \le A_{cpu} \le 1 \tag{4}$$

Memory availability represents how much the memory capacity of a mobile resource remains and can be calculated as follows:

$$A_{mem} = 1 - (U_{mem} / T_{mem}), \qquad 0 \le A_{mem} \le 1 \tag{5}$$

Battery availability represents how much the battery capacity of a mobile resource is available and can be calculated as follows:

$$A_{batt} = 1 - (U_{batt} / T_{batt}), \qquad 0 \le A_{batt} \le 1 \tag{6}$$

We define that mobile resource reliability is a combination of network reliability and resource reliability because network reliability is related to fault tolerance. If a mobile resource is disconnected or downed, the down time may cause substantial job loss. Therefore, rapid recovery from failure and response to user request are needed [24]. Response rate and recovery rate also affect the connection state and communication time of a mobile resource. Therfore, the network reliability of a mobile resource is:

$$R(n) = \varepsilon \cdot R_{res} + \delta \cdot R_{rec} \qquad (\varepsilon + \delta = 1) \tag{7}$$

where ε and δ are decided by the current connection state of the mobile resource. In case of a mobile resource with “Connect”, ε is larger than δ since the mobile resource has to concentrate on quick response. Otherwise, ε is smaller than δ because a mobile resource with “Disconnect” has to concentrate on connection recovery. In this paper, we set ε and δ to 0.6 and 0.4 if a mobile resource is in “Connect”; otherwise, we set them to 0.4 and 0.6.

In a mobile grid environment, battery availability is an important measure which reflects the portability and mobility of a mobile resource. If a mobile resource is out of battery, the mobile resource becomes disconnected from wireless networks and impossible to run. In order to obtain resource reliability, we use CPU availability and memory availability that are traditional performance metrics as well as battery availability. Therefore, the resource reliability of a mobile resource is:

$$R(r) = \kappa \cdot A_{cpu} + \lambda \cdot A_{mem} + \gamma \cdot A_{batt} \qquad (\kappa + \lambda + \gamma = 1) \tag{8}$$

In this formula, we adjust κ, λ, and γ according to the type of jobs. For example, CPU availability has a high weight for solving high performance computing problems and memory availability has a high weight for solving large-scale data problems.

If a mobile resource is repeatedly connected or disconnected with a wireless network by its mobility, the mobile resource reliability of the mobile resource can be probably expressed as follows:

$$R(mr) = \{P(c) \cdot R(n) \cdot R(r)\} + \{(1 - P(c)) \cdot R(n) \cdot R(r)\} \qquad (0 \le R(mr) \le 1) \tag{9}$$

We can calculate the mobile resource reliability of each mobile resource when reliability metrics are changed. However, it is undesirable to recalculate mobile resource reliability using the above formula, whenever any reliability metric is changed. The reason is that it will cost too much and generate bottleneck by a huge increase in communication messages between mobile resources and the mobile job coordinator. Also, mobile resource reliability is time series data dynamically changed by variations in the amount of allocated jobs as time passes. Therefore, at a preset interval called the scheduling cycle, we predict mobile resource availability using the following formula, which is based on the exponential smoothing method [25] used in predicting various time series data of grid computing:

$$fR(mr)_n = \left[\prod_{i=1}^{n-1} (1 - \omega) \cdot \alpha_i + \omega \cdot \beta_i \right] \times fR(mr)_{n-1} \tag{10}$$

where ω can be decided from 0 to 1 and we obtained the optimal ω 0.9 by an experiment. We provide historical error sets (α and β) for long-term prediction and diminish system loads caused by repeated computation. Those error sets enable us to predict mobile resource reliability more precisely. α and β are calculated as follows:

$$\alpha_i = fR(mr)_i \,/\, fR(mr)_{i-1} \tag{11}$$

$$\beta_i = yR(mr)_i \,/\, yR(mr)_{i-1} \tag{12}$$

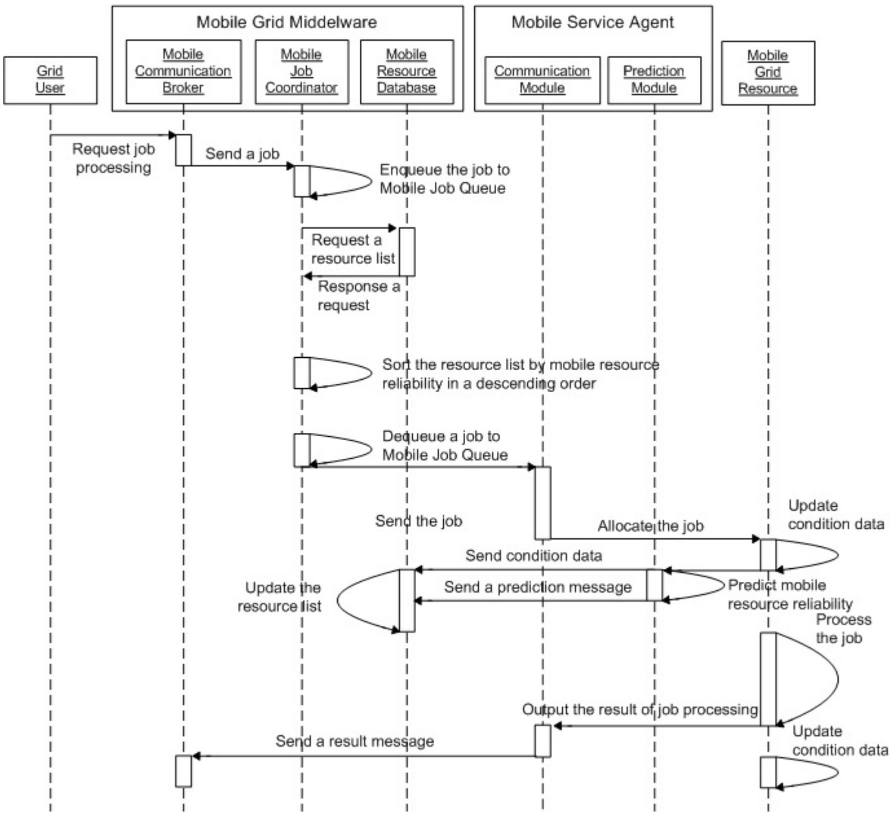

Fig. 4. Sequence diagram of the mobile resource reliability-based job scheduling.

The mobile job coordinator allocates jobs to proper mobile resources based on mobile resource reliability. As **Fig. 4** illustrates, the mobile resource reliability-based job scheduling procedures proceed as follows:

1. When a grid user requests job processing, the mobile communication broker sends a job from the grid user to the mobile job coordinator.
2. The mobile job coordinator enqueues the job to the mobile job queue.
3. The mobile job coordinator requests a resource list, which includes information about available resources, to the mobile resource database. The resource list involves the mobile resource reliability and other parameters of each mobile resource.
4. For job allocation, the mobile job coordinator sorts the resource list by mobile resource reliability in a descending order and dequeues a job from the mobile job queue.
5. The mobile job coordinator sends the job to the communication module of the mobile agent dispatched to a mobile resource, which is located in the top of the resource list.
6. The mobile resource, on which the job is processed, updates its own condition data and sends a condition message to the mobile resource database and the prediction module.
7. The prediction module calculates and predicts the mobile resource reliability of the mobile resource. The predicted mobile resource reliability is sent from the prediction module to the mobile resource database, and then the mobile resource database update

the resource list.

8. When the job is completed to process on the mobile resource, the result of the job is transmitted from the mobile resource to the mobile job coordinator through the communication module. At this time, the mobile resource updates its own condition data once again.

4. Performance Evaluation & Discussion

In this Section, we simulated the mobile resource reliability-based job scheduling model on the DEVSJAVA, which is a modeling and simulation platform for analyzing discrete event systems [26] and presented simulation results of prediction and job scheduling algorithms described in Section 3.2. For simplifying the complexity of simulations, we assume that three different types of resources are located within a grid network. Parameters of each resource type are presented in **Table 2**. Parameters in **Table 2** are used to represent the heterogeneity of mobile grid resources. As shown in **Table 2**, we set up different parameter values by the resource type because the heterogeneity of mobile grid resources hasn’t been formally defined yet.

Table 2. Parameters & values of resource types.

Type	*pT* (sec.)	*dR* (%)	*rR* (%)	*rT* (sec.)	*dT* (sec.)	*qS* (n)	*Mem* (MB)	*CPU* (Ghz)
A	16~20	3.3	80	20	2~5	10	125	1
B	14~18	2.5	90	25	3~6	15	256	1.5
C	10~14	2.0	100	30	4~8	20	512	2

In **Table 2**, processing time (*pT*) is the time required to process a job on a mobile resource. Disconnection rate (*dR*) indicates how many times a mobile resource is in “Disconnect.” In brief, the type A of a mobile resource is disconnected whenever the mobile resource receives 30 jobs. Recovery rate (*rR*) is the probability that a disconnected mobile resource succeeds in reconnection. If a mobile resource is failed in recovery, all jobs within the mobile resource are abandoned. Reconnection time (*rT*) is the waiting time required to convert a state of “Disconnect” into a state of “Connect.” Delay time (*dT*) indicates the sum of sending and receiving delay times. Queue size (*qS*) is the total queue length to store jobs and substitutes for storage capacity. We assume that all mobile resources are always on charge in experiments. CPU and memory represent the processing power of a mobile resource. If a number of jobs are inputted to a mobile resource and processed on the mobile resource, its processor utilization increases and its processing power is degraded. In order to express this performance degradation of a mobile resource, the used capacities of CPU and memory increase in proportion to the number of accumulated jobs in the queue of the mobile resource. If a mobile resource is the type A, its CPU and memory capacities are decreased to 80% when two jobs are accumulated in its queue. If the CPU and memory capacities of a mobile resource are less than 50%, we assume that the processing time of the mobile resource is increased by performance degradation as follows:

$$pT_{new} = pT * (1+Uc / Tc) \tag{13}$$

where pT_{new} is the changed processing time of the mobile resource, *Uc* is the used capacity of CPU or memory, and *Tc* is the total capacity of CPU or memory.

In our experiments, the job model is a single job, which is not a bag of tasks. A job is independently processed on a mobile resource, therefore it is non-DAG. If a job is divided into several tasks, the tasks should be processed on multi processors of a mobile resource by a parallel program. This parallel processing lies beyond the scope of our research. We generated and processed a total of 1500 jobs during simulation. Tasks are all the same size and the required time to process each job is decided by the processing time of a mobile resource.

Fig. 5 shows the component composition of this model. The simulation model consists of four types of components; the generator, the coordinator, the analyzer, and the grid site. The generator is in charge of a grid user and sends a job to the coordinator at two second intervals. The coordinator, a scheduler with several scheduling policies depending on simulated job scheduling models, contains a queue and a resource list. The coordinator also allocates the job to proper resources according to each job scheduling model. The coordinator also stores prediction messages from agents to the resource list. The grid site is a resource group, which includes *n* resources. The resource is composed of an agent and a processor. The processor processes a job received from the coordinator for its processing time and transmits its result to the analyzer. At the same time, the processor sends an updated condition data to the agent. The agent measures response rate, recovery rate, CPU availability, memory availability, and battery availability of the resource. Subsequently, the agent calculates and predicts mobile resource reliability at a preset interval. This predicted mobile resource reliability is stored in a prediction message, which is sent to the coordinator for job scheduling. The analyzer evaluates the whole performance of each scheduling model by job loss, utilization, throughput, and etc.

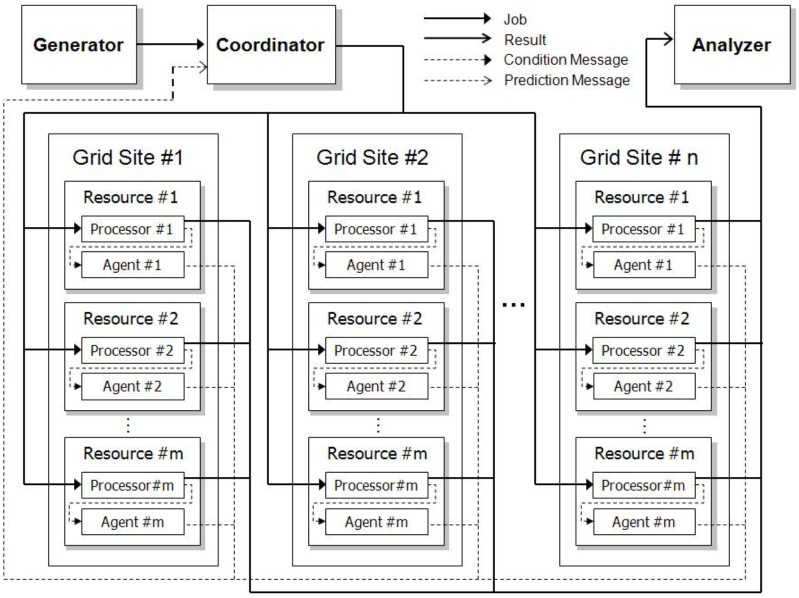

Fig. 5. Component composition of our simulation model.

4.1 Prediction Accuracy

As mentioned in Section 3.2, the smoothing constant ω of the mobile resource reliability prediction method is an important parameter to affect prediction accuracy. The range of mobile resource reliability is affected by changes of ω. We therefore conducted an experiment to find an optimal smoothing constant. In this experiment, the mean absolute percentage error (MAPE) [25] is used as a measure of prediction accuracy. The MAPE is calculated as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{yR_i - fR_i}{yR_i}\right| \times 100(\%) \tag{14}$$

where *yR* denotes a measured mobile resource reliability value and *fR* indicates a predicted mobile resource reliability value and *n* is the number of sample values. The more close to zero the MAPE is, the higher prediction accuracy becomes.

The value of ω is between 0 and 1. Thus, we compared the MAPE as changing ω from 0.1 to 0.9 by 0.1. **Table 3** represents the average MAPE by variations of smoothing constant ω. The least MAPE is obtained when ω is equal to 0.9. The more close to 1 ω is, the more precise we get a prediction value reflecting prediction error. In contrast, the more close to 0 ω is, the more similar to the prior prediction value, in which the range of fluctuation is too wide. By the result of **Table 3**, we are aware that an optimal value of ω for long-term prediction is 0.9.

Table 3. Average of the MAPE by changes in a smoothing constant.

Smoothing constant ω		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
MAPE (%)	Time = 500 sec.	9.28	6.86	5.92	5.28	4.66	4.31	3.99	3.58	3.15
	Time = 1500 sec.	14.13	12.82	11.6	11.28	9.79	9.33	8.4	6.49	5.12

However, it is uncertain that the mobile resource reliability prediction method is ideal to predict mobile resource reliability. We therefore demonstrate the suitability of our prediction method in comparison with existing prediction methods that are the moving average method [25], the Holt's exponential smoothing method [27], and the Brown's exponential smoothing method [28]. **Table 4** shows the MAPE of each method by the simulation time. In **Table 4**, the average MAPE of the proposed method is about 4.25 %. This value is 52% less than the MAPE of other methods. This result represents that the proposed method does not assure us of higher prediction accuracy than other methods in either case, but is better than other methods in case of predicting mobile resource reliability.

Table 4. Comparison of the MAPE (the mobile resource reliability prediction method (MRRPM) vs. the moving average method (MAM) vs. the Holt's exponential smoothing method (HESM) vs. the Brown's exponential smoothing method (BESM)).

Simulation time	MRRPM	MAM	HESM	BESM
200	1.71	0.19	5.18	3.78
400	3.98	8.30	7.17	7.26
600	4.76	13.65	7.93	8.33
800	5.23	16.79	8.35	8.99
1000	5.59	18.73	8.58	9.41

4.2 Prediction vs. Non-Prediction

For simulation, we set ω as 0.9, which was obtained in previous section. We also set κ, λ, and γ as 0.5, 0.25, and 0.25 because we assume that jobs generated during simulations are computing jobs. In order to verify the effectiveness of job scheduling with prediction, we compared the throughput and communication traffic of the mobile resource reliability based-job scheduling model (MRRJSM) with prediction to those of the MRRJSM without prediction. Throughput is to measure how quickly to process jobs and communication traffic is to measure how stably to process jobs without network congestion. Throughput can be calculated as follows:

$$Throughput = \frac{1}{t}\sum_{i=1}^{Nt} Np_i \tag{15}$$

where Np_i is the number of jobs processed by a mobile resource, Nt is the total number of mobile resources, and t is the simulation time.

The simulation result of the previous section shows there is little difference between the predicted mobile resource reliability and the measured mobile resource reliability. Therefore, we can guess that the throughput of the MRRJSM based on predicted values is almost identical to that of the MRRJSM based on measured values. As **Fig. 6** illustrates, the throughput of the MRRJSM with prediction is 0.94 jobs similar to that of the MRRJSM without prediction. That is to say that applying the proposed prediction method to the MRRJSM does not inflict a loss on throughput.

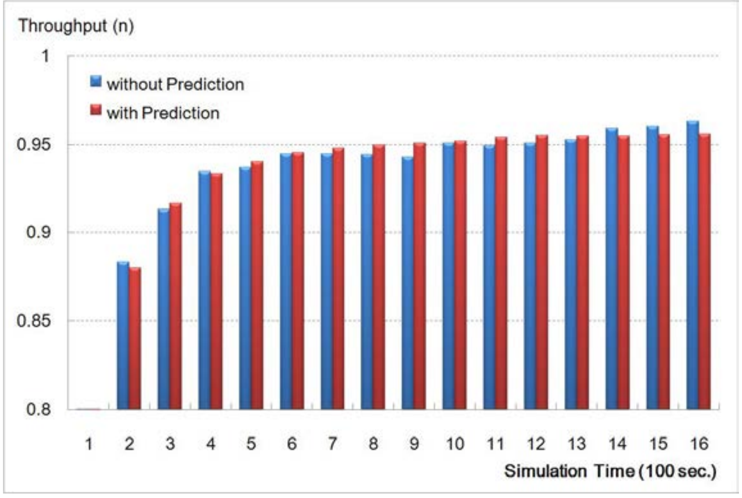

Fig. 6. Comparison of throughput (The MRRJSM with prediction vs. the MRRJSM without prediction).

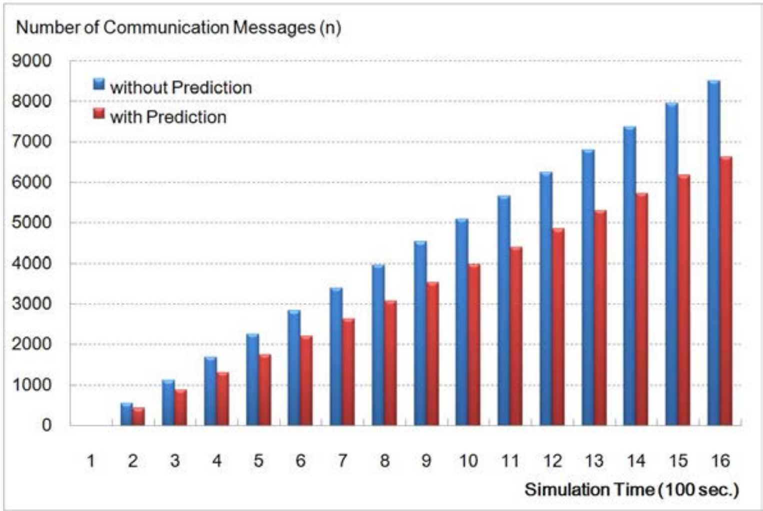

Fig. 7. Comparison of communication message (The MRRJSM with prediction vs. the MRRJSM without prediction).

We also compared the communication traffic of the MRRJSM with prediction to that of the

MRRJSM without prediction. In order to represent changes in communication traffic, the total number of communication messages is measured by the simulation time regardless of the size of messages. The MRRJSM with prediction generated messages about mobile resource reliability every scheduling cycle and the MRRJSM without prediction generated messages whenever the status of each mobile resource is changed. As **Fig. 7** illustrates, the MRRJSM with prediction generated 6,612 communication messages while the MRRJSM without prediction generated 8,502 communication messages during 1600s. On the average, the MRRJSM with prediction reduces 23% more communication messages than the MRRJSM without prediction. These results prove that the mobile resource reliability prediction method can help to solve concentration of workloads and curtail communication costs because it reduces the number of communication messages without a decrease in throughput.

4.3 Proposed Scheduling Model vs. Existing Mobile Grid Scheduling Models

In this Section, we demonstrate the efficiency and effectiveness of the MRRJSM in comparison with the response time-based job scheduling model (RTJSM) [10] and the execution time prediction-based job scheduling model (ETPJSM) [11].

The purpose of our paper is to propose a job scheduling model, which can assist quick and stable job processing in a mobile grid environment. In order to prove the contribution of our paper, we measured throughput, job loss rate, job latency, utilization, and communication traffic of each model. Throughput, job loss rate, and job latency are general performance metrics for evaluating scheduling strategies in a Grid-based resource model [29]. Utilization is a measure to verify whether jobs are evenly distributed by each scheduling model or not. Communication traffic is to measure how much each scheduling model can reduce communication cost and overhead. All experiments are repeated under the same conditions five times and their results are recorded in a file.

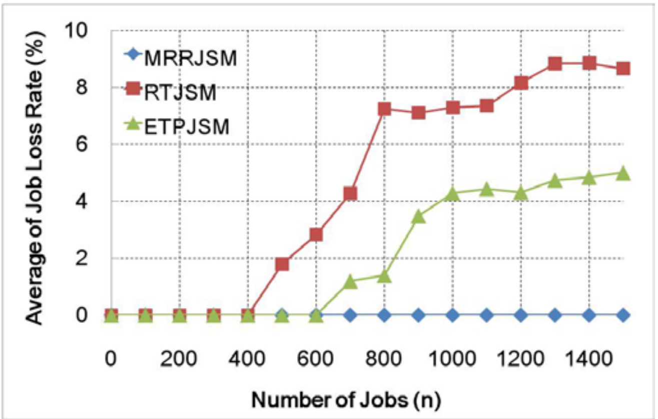

Fig. 8. Comparison of job loss rate (the mobile resource reliability-based job scheduling model (MRRJSM) vs. the response time-based job scheduling model (RTJSM) vs. the execution time prediction-based job scheduling model (ETPJSM)).

Fig. 8 graphs the comparison of job loss rate by the number of jobs. Job loss rate is the average rate of job losses, denoted by R_{pl}:

$$R_{pl} = [1-((n_{sp} - n_{lp})/n)] \times 100(\%) \quad (16)$$

where n_{sp} is the number of solved jobs, n_{lp} is the number of lost jobs, and n is the total number of generated jobs.

As **Fig. 8** illustrates, the MRRJSM did not generate any job loss, while the RTJSM and the

ETPJSM lost 8.66 % and 5.01% of total generated jobs during the simulation time. Although there are ample mobile resources to process all jobs, the RTJSM and the ETPJSM generate job losses because they occasionally allocated jobs to disconnected resources.

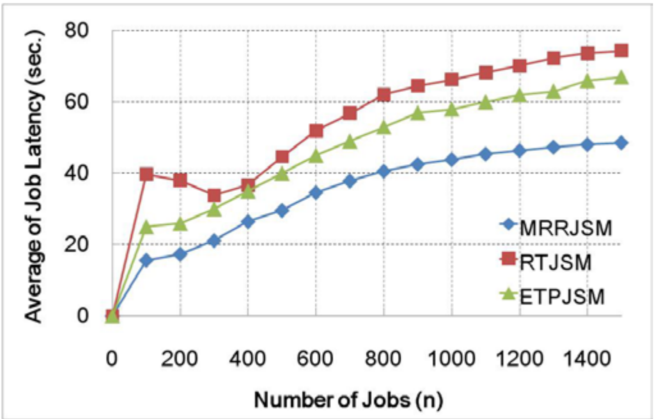

Fig. 9. Comparison of job latency (the mobile resource reliability-based job scheduling model (MRRJSM) vs. the response time-based job scheduling model (RTJSM) vs. the execution time prediction-based job scheduling model (ETPJSM)).

Fig. 9 graphs the comparison of job latency by the number of jobs. Job latency is the average necessary time to process a job and transfer its result, denoted by P_l:

$$P_l = \frac{1}{n}\sum_{i=1}^{n}(Ta_i - Ts_i) \tag{17}$$

where Ta_i is the allocation time of a job, Ts_i is the solved time of the job, and n is the total number of jobs.

As shown in **Fig. 9**, the average job latency of the MRRJSM is 34.1s while that of the RTJSM is 53.3s and that of the ETPJSM is 46.2s. This result shows that the MRRJSM can averagely reduce the time required to process a job by about 31% in comparison with the RTJSM and the ETPJSM. The RTJSM and the ETPJSM job allocation is biased to resources with short response or execution time, while the MRRJSM allocated jobs to resources moderately.

Fig. 10 graphs the comparison of utilization by the simulation time. Utilization is the average rate of resource utilization and can be calculated as follows:

$$R_{util} = \frac{1}{n}[(\sum_{i=1}^{n}(\frac{pt}{pt\ +\ wt})\times 100(\%))] \tag{18}$$

where pt is the total processing time of a mobile resource and wt is the total waiting time of the mobile resource, and n is the number of mobile resources.

On the average, the MRRJSM recoded 54% utilization, which is 29% higher than the average utilization of the RTJSM and the ETPJSM. In the MRRJSM, jobs were evenly distributed to all mobile resources. On the other hand, there was the unequal distribution of jobs in the RTJSM and the ETPJSM because they preferred specific resources with high speed for job allocation.

Fig. 11 graphs the comparison of throughput by the simulation time. The average throughput of the MRRJSM is 0.47 jobs that does not differ much from that of the ETPJSM, but is 6% higher than that of the RTJSM.

Fig. 12 graphs the comparison of communication traffic by the simulation time. As explained in the previous section, communication traffic is the number of communication messages. During simulation, the MRRJSM generated 2751 messages that are about the same as communication messages generated by the ETPJSM because both models provide prediction method and generate messages at a regular interval called scheduling cycle. On the other hand, the RTJSM recorded a total of 3672 messages since the RTJSM generated messages every time information on each mobile resource is updated.

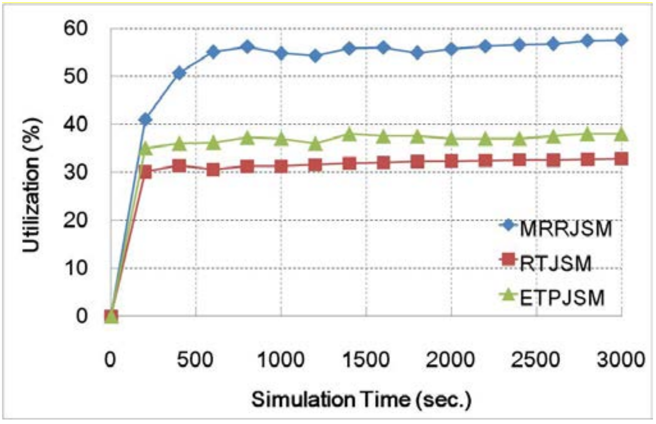

Fig. 10. Comparison of utilization (the mobile resource reliability-based job scheduling model (MRRJSM) vs. the response time-based job scheduling model (RTJSM) vs. the execution time prediction-based job scheduling model (ETPJSM)).

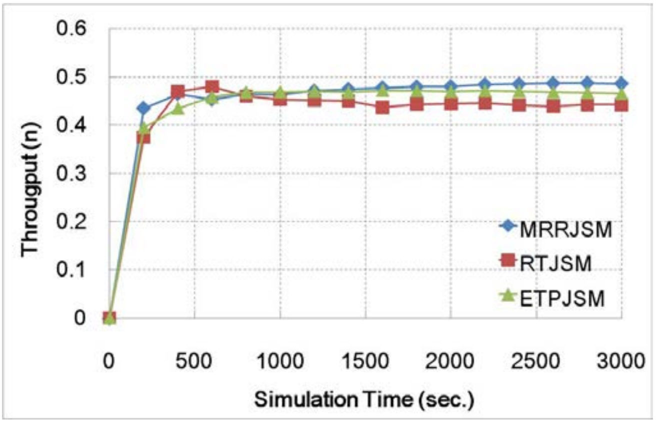

Fig. 11. Comparison of throughput (the mobile resource reliability-based job scheduling model (MRRJSM) vs. the response time-based job scheduling model (RTJSM) vs. the execution time prediction-based job scheduling model (ETPJSM)).

These results prove that the proposed model can provide the improved QoS and computing performance to grid users with stable job processing service and overcome the unreliability of a mobile grid environment in comparison with existing scheduling models.

As mentioned previously, we simulated each scheduling model five times and calculated the mean value and standard deviation of performance metrics by each simulated job scheduling model so as to insure the validity of simulation results. We also constructed confidence intervals for the differences between performance metrics of the proposed job scheduling model and those of the RTJSM and the ETPJSM by using the Welch confidence interval approach [30]. **Table 5** shows the means and standard deviations of performance metrics, as well as 97.5% confidence intervals the differences between performance measures of the MRRJSM and those of the RTJSM and the ETPJSM. By the result of **Table 5**, with approximately 95% confidence, we can say that the MRRJSM can process jobs more stably and quickly than the RTJSM and the ETPJSM because the MRRJSM reduced job loss rate, job latency, and communication traffic and improved utilization and throughput in comparison with other simulated models.

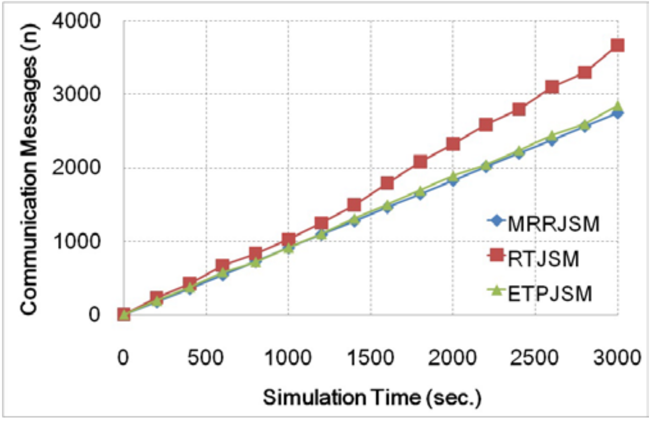

Fig. 12. Comparison of the number of communication messages (the mobile resource reliability-based job scheduling model (MRRJSM) vs. the response time-based job scheduling model (RTJSM) vs. the execution time prediction-based job scheduling model (ETPJSM)).

Table 5. Individual 97.5 percent confidence intervals for all comparisons of performance measures (the mobile resource reliability-based job scheduling model (MRRJSM) vs. the response time-based job scheduling model (RTJSM) vs. the execution time prediction-based job scheduling model (ETPJSM)).

Performance measure		MRRJSM (X_1)	RTJSM (X_2)	ETPJSM (X_3)
Job loss rate (%)	Mean ($\bar{X}_i$)	0.09	7.93	4.79
	$\bar{X}_1 - \bar{X}_i$		7.84	4.70
	Interval		(7.19, 8.47)	(4.36, 5.03)
Job latency (sec.)	Mean ($\bar{X}_i$)	47.70	75.47	67.19
	$\bar{X}_1 - \bar{X}_i$		27.77	19.49
	Interval		(26.41, 29.12)	(18.30, 20.68)
Utilization (%)	Mean ($\bar{X}_i$)	57.80	34.97	38.74
	$\bar{X}_1 - \bar{X}_i$		-22.83	-19.06
	Interval		(-24.39, -21.27)	(-20.53, -17.60)
Throughput (n)	Mean ($\bar{X}_i$)	0.49	0.44	0.46
	$\bar{X}_1 - \bar{X}_i$		-0.05	-0.03
	Interval		(-0.07, -0.04)	(-0.05, -0.02)
Communication Traffic (n)	Mean ($\bar{X}_i$)	2700.80	3668.40	2796.20
	$\bar{X}_1 - \bar{X}_i$		867.60	95.40
	Interval		(706.66, 1028.54)	(-18.77 , 209.58)

4.4 Discussion

Diverse job scheduling models for grid computing have been presented so far. Most of these job scheduling models have been developed for quick job processing. The mobile resource reliability-based job scheduling model (MRRJSM) proposed by this paper focuses on not only quick job processing but also stable job processing. The proposed model can realize job scheduling adaptable for dynamic characteristics of mobile grid by measuring mobile resource reliability and applying the statistical prediction method. We conducted experiments to evaluate the effectiveness of the proposed model.

Firstly, we demonstrated how much accurate the proposed prediction method is. Our prediction method performed 63% better than the moving average method [25], the Holt's exponential smoothing method [27] and the Brown's exponential smoothing method [28] in prediction accuracy in Section 4. 1. However, we were curious how the prediction method has an effect on the proposed job scheduling model.

Secondly, we compared the MRRJSM with prediction to the MRRJSM without prediction in Section 4.2. In case of throughput, there was little difference between the two. However, in case of communication traffic, the MRRJSM with prediction generated 6,612 communication messages while the MRRJSM without prediction generated 8,502 communication messages - about 23 percent difference. By these results, we were aware that our prediction method does not improve computing performance but can reduce communication load and cost.

Finally, we compared the MRRJSM with the RTJSM [10] and the ETPJSM [11], which are job scheduling models for mobile grid, in Section 4.3. In result, we confirmed that the MRRJSM provides more improved computing performance than the RTJSM and the ETPJSM. Contrary to stable and uniform job allocation of the MRRJSM, the RTJSM and the ETPJSM have a problem that job losses occur by job allocation biased toward mobile resources with short response or execution time. The results of utilization and job loss rate prove this fact.

Of course, the MRRJSM has some disadvantages whereas the MRRJSM provides many benefits. In case of communication traffic, the MRRJSM is likely to generate more communication messages than other scheduling models because of communication messages to measure and predict mobile resource reliability. However, we believe that messages for measuring and predicting mobile resource reliability do not have great effect on real communication traffic and cost since those messages are transferred between internal components. In addition, if the quick job processing takes precedence over stable job processing, existing scheduling models will probably be better than the MRRJSM. To solve this problem, we plan to develop a trade-off based job scheduling model which can provide various scheduling algorithms adaptable to requirements of grid users.

5. Conclusions

This paper deals with resource management and job scheduling of mobile grid. Serious job loss and latency can be generated in a mobile grid environment when a mobile resource is disconnected with wireless networks or the amount of allocated jobs exceeds the maximum throughput. The major purpose of this paper is to solve job scheduling problems in a mobile grid environment by taking the connectivity and availability of mobile resources into account. Therefore, we proposed the mobile resource reliability-based job scheduling model. This model calculates the current mobile resource reliability of each mobile resource on the basis of diverse reliability metrics and predicts its mobile resource reliability for job allocation by using the statistical prediction method. Jobs requested from grid users are allocated to resources with high mobile resource reliability.

We also designed the simulation model which simplifies various functions of the mobile resource reliability-based job scheduling model by using the DEVS modeling and simulation formalism [26] and conducted some experiments for performance evaluation. Empirical data shows that the proposed method predicts mobile resource reliability 63% more precisely than other prediction methods. Also, about 23% of communication messages were reduced without performance degradation if the proposed prediction method is applied to the mobile resource reliability-based job scheduling model. Furthermore, the proposed scheduling model provided higher utilization and throughput and lower job loss rate and job latency than those of existing job scheduling models.

Experiment results demonstrate that the proposed scheduling model can assist in improving the QoS and computing performance of mobile grid and overcoming the unreliability of mobile resources. The proposed job scheduling algorithm can also be extended to general grid systems because connectivity and availability are universally applicable to static resources as well as mobile resources. Nevertheless, there are still challenges such as communication service, location service, and security to realize ideal mobile grid. We will deal with those issues in future works and develop the prototype of a practical mobile grid scheduling model.

References

[1] F. Berman, G. Fox and T. Hey, “Grid Computing: Making the Global Infrastructure a Reality,” *John Wiley & Sons, Ltd*, 2003. Article (CrossRef Link).

[2] I. Foster and C. Kesselman, “The Grid: Blueprint for a New Computing Infrastructure,” Morgan Kaufmann Publishers, 1999. Article (CrossRef Link).

[3] I. Foster, C. Kesselman and S. Tuecke, “The anatomy of the grid: enabling scalable virtual organizations,” *International Journal of High Performance Computing Application*, vo.15, no.3, pp.200-222, 2001. Article (CrossRef Link).

[4] T. Phan, L. Huang and C. Dulan, “Challenge: integrating mobile wireless devices into the computational grid,” in *Proc. of 8th ACM Int. Conf. on Mobile Computing and Networking*, pp.271-278, 2002. Article (CrossRef Link).

[5] J. H. Oh, S. H. Lee and E. S. Lee, “An adaptive mobile system using mobile grid computing in wireless network,” in *Proc. of Int. Conf. on Computational Science and its Applications 2006*, pp.49-57, 2006. Article (CrossRef Link).

[6] W. Y. Zeng, Y. L. Zhao, J. W. Zeng and W. Song, “Mobile grid architecture design and application,” in *Proc. of 4th Int. Conf. on Wireless Communications, Networking and Mobile Computing 2008*, pp.1-4, 2008. Article (CrossRef Link).

[7] I. Foster and C. Kesselman, “The Globus project: a status report,” in *Proc. of 7th Heterogeneous Computing Workshop*, pp.4-18, 1998. Article (CrossRef Link).

[8] D. Kondo, H. Casanova, E. Wing and F. Berman, “Models and scheduling mechanisms for global computing applications,” in *Proc. of 16th Int. Parallel and Distributed Processing Symposium*, pp.79-86, 2002. Article (CrossRef Link).

[9] K. Li, “Experimental performance evaluation of job scheduling and processor allocation algorithms for grid computing on metacomputers,” in *Proc. of 18th Int. Parallel and Distributed Processing Symposium*, pp.170-177, 2004. Article (CrossRef Link).

[10] S. M. Park, Y. B. Ko and J. H. Kim, “Disconnected operation service in mobile grid computing,” *Lecture Notes in Computer Science*, vol.2910, pp.499-513, 2003. Article (CrossRef Link).

[11] M. Ballette, A. Liotta and S. M. Ramzy, “Execution time prediction in DSM-based mobile grids,” in *Proc. of 5th IEEE Int. Symp. on Cluster Computing and the Grid*, pp.881-888, 2005. Article (CrossRef Link).

[12] P. Ghosh, N. Roy and S.K. Das, “Mobility-aware efficient job scheduling in mobile grids,” in *Proc. of 7th IEEE Int. Symp. on Cluster Computing and the Grid*, pp.701-706, 2007. Article (CrossRef Link).

[13] X. Shi, H. Jin, W. Qiang and D. Zou, “Reliability analysis for grid computing,” *Lecture Notes in Computer Science*, vol.3251, pp.787-790, 2004. Article (CrossRef Link).

[14] H. L. Liu and M. L. Shooman, “Simulation of computer network reliability with congestion,” in *Proc. of the IEEE Annual Reliability and Maintainability Symposium*, pp.208-213, 1999. Article (CrossRef Link).

[15] C. Li, N. Xiao and X. Yang, “Application availability measurement in computational grid,” *Lecture Notes in Computer Science*, vol.3032, pp.151-154, 2003. Article (CrossRef Link).

[16] C. Li, N. Xiao and X. Yang, “Predicting the reliability of resources in computational grid,” *Lecture Notes in Computer Science*, vol.3251, pp.233-240, 2004. Article (CrossRef Link).

[17] Y. S. Dai, M. Xie and K. L. Poh, “Reliability analysis of grid computing systems,” in *Proc. of 9th Pacific Rim International Symposium on Dependable Computing*, pp.97-103, 2002. Article (CrossRef Link).

[18] Y. S. Dai, Y. Pan and X. Zou, “A hierarchical modeling and analysis for grid service reliability,” *IEEE Transactions on Computers*, vol.56. no.5, pp.681-691, 2007. Article (CrossRef Link).

[19] H. Jameel, U. Kalim, A. Sajjad, S. Lee and T. Jeon, “Mobile-to-grid middleware: bridging the gap between mobile and grid environments,” in *Proc. of European Grid Conference EGC 2005*, pp. 932-941, 2005. Article (CrossRef Link).

[20] F. Navarro, A. Schulter, F. Koch, M. Assunção and C. B. Westphall, “Towards a middleware for mobile grids,” in *Proc. of 10th IEEE/IFIP Network Operations and Management Symposium 2006*, pp.1-4, 2006. Article (CrossRef Link).

[21] L. Zhou, N. Xiong, L. Shu, A. Vasilakos and S. S. Yeo, “Context-Aware Multimedia Service in Heterogeneous Networks,” *IEEE Intelligent Systems*, vol.25. no.2, pp.40-47, 2010. Article (CrossRef Link).

[22] P. Bar, C. Coti, D. Groen, T. Herault, V. Kravtsov, A. Schuster and M. Swain, “Running Parallel Applications with Topology-Aware Grid Middleware,” in *Proc. 5th IEEE International Conference on e-Science*, pp.292-299, 2009. Article (CrossRef Link).

[23] K. Krauter, R. Buyya and M. Maheswaran, “A taxonomy and survey of grid resource management systems,” *Software Practice and Experience*, vol.32. no.2, pp.135-164, 2002. Article (CrossRef Link).

[24] D. Medhi, “Network Reliability and Fault Tolerance,” *Wiley Encyclopedia of Electrical and Electronics Engineering*, John Wiley, 1999. Article (CrossRef Link).

[25] B. Bowerman, R. O'Connell and A. Koehler, “Forecasting, Time Series and Regression,” 4th Edition, Thomson Books/Cole, 2004.

[26] B. P. Zeigler, H. S. Sarjoughian, S.W. Park, J. S. Lee, Y. K. Cho and J. J. Nutaro, “DEVS modeling and simulation: a new layer of middleware,” in *Proc. of 3rd Annual Int. Workshop on Active Middleware Services*, pp.22-31, 2001. Article (CrossRef Link).

[27] C. Chatfield and M. Yar, “Holt-winters forecasting: Some practical issues,” *Statistician*, vol.37. no.2, pp.129-140, 1998. Article (CrossRef Link).

[28] R.G. Brown, “Smoothing, Forecasting and Prediction of Discrete Time Series,” Englewood Cliffs, 1963.

[29] N. Thomas, J.T. Bradley and W.J. Knottenbelt, “Stochastic analysis of scheduling strategies in a Grid-based resource model,” *IEE Proceedings Software*, vol.151. no.5, pp.232-239, 2004. Article (CrossRef Link).

[30] A. M. Law, “Simulation Modeling and Analysis,” 4th Edition, McGraw-Hill, 2007.

Sung Ho Jang is received a B.S. degree in Computer Science and Information from Yongin University, Republic of Korea, in 2004. And, he received M.S. and Ph.D. degrees in Computer Science and Information Engineering from Inha University, Republic of Korea, in 2006 and 2011. His research interests include mobile computing; cloud computing, resource management, job scheduling, and artificial intelligence.

Jong Sik Lee is an associate professor in the School of Computer Science and Information Engineering, Inha University, Republic of Korea. He received B.S. and M.S. degrees in Electronics Engineering from Inha University, Republic of Korea, in 1993 and 1995. And, he received a Ph.D. degree in Electrical and Computer Engineering from University of Arizona, USA, in 2001. Dr. Lee's research interests include grid computing, mobile computing, resource management, software modeling and simulation.