IP-Paging base Resource Management and Task Migration in Mobile Grid Environments

DaeWon Lee
Division of General Education, SeoKyeong University, Korea
daelee@skuniv.ac.kr

Abstract

Grid computing using wireless networks is receiving increasing attention and is expected to become a critical part of future grid computing. However, the inherent challenges of mobile environments such as mobility management, disconnected operation, device heterogeneity, service discovery, and resource sharing are significant issues in mobile grid computing. To achieve the best performance in a mobile grid computing environment, the mobile devices with the lowest probability of mobility should be selected for use first. We therefore focus on the idle state of mobile devices and use IP-paging scheme to identify the idle mobile devices. For this, we propose a user-defined checkpoint technique for task migration, which is based on the information of when a mobile device will leave the network or stop due to low battery. Our checkpoint technique performs checkpoints by two conditions: when a mobile device leaves its current cell and when it turns off due to low battery capacity. By performance evaluation, proposed task migration scheme decreases total execution time and total job completion time.

Keywords: *Mobile Grid, IP-Paging, Task Migration, Checkpoints.*

1. Introduction

Grid computing is distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and high-performance orientation [1]. In the first stages of grid computing, most research has focused on fixed networks [1,2,3,4,5]. Because of improved Internet techniques, grid computing using wireless networks is now the subject of growing attention. It is expected to become a critical part of future grid computing involving mobile hosts to facilitate user access to grid networks to offer extended computing resources. Previous methods of resource selection in grid computing were suitable for resources based on the user requirement. However, because of the mobility in wireless networks, this resource selection method will not result in the best performance for mobile grid computing. In a mobile grid computing environment, the mobile device that has the lowest probability of mobility should be selected as a resource first to maximize grid performance. We therefore focus on idle mobile devices. To achieve this purpose, this paper uses an IP-paging scheme [6,7,8] to find idle mobile devices and to gather information about them using an extended paging message.

The inherent challenges of mobile environments such as mobility management, disconnected operation, device heterogeneity, service discovery, and resource sharing are significant issues for grid computing. To overcome these challenges, we propose a task migration scheme using checkpoints for mobile grid computing [9,10,11,12]. This task migration scheme performs better than a re-execution scheme, although it does incur additional cost to perform a checkpoint procedure. For continuous grid services, it is useful to

predict when a device will leave the network or stop due to low battery. In this paper, we use a user-defined checkpoint technique for task migration. This technique applies in two situations: when a mobile device leaves its current cell (determined by an analysis of its signaling strength) and when a mobile device turns off due to lower battery capacity.

The rest of this paper is organized as follows. Section 2 presents related works on mobile grid computing and the IP-paging scheme. Section 3 describes the proposed system architecture. Section 4 gives details of resource management for mobile devices. In Section 5, the task migration using the user-defined checkpoints is presented. Section 6 presents the performance evaluation of the proposed paging mobile grid system. Finally, we conclude the paper with future works in Section 7.

2. Related Work

In this section, we present three kinds of related works that are fundamental of this paper. First of all, the proxy based mobile grid architecture is presented for our proposed architecture. Secondly, the IP paging scheme is presented for idle mobile resources. Finally, the check pointing technique is presented for task migration in mobile grid environment.

2.1. Mobile Grid Computing

Two architectures exist for mobile grid computing: proxy-based and agent-based one. Proxy-based mobile grid architectures, designed to use mobile devices as resources for computational grid, have been proposed in [9,10,11,12].

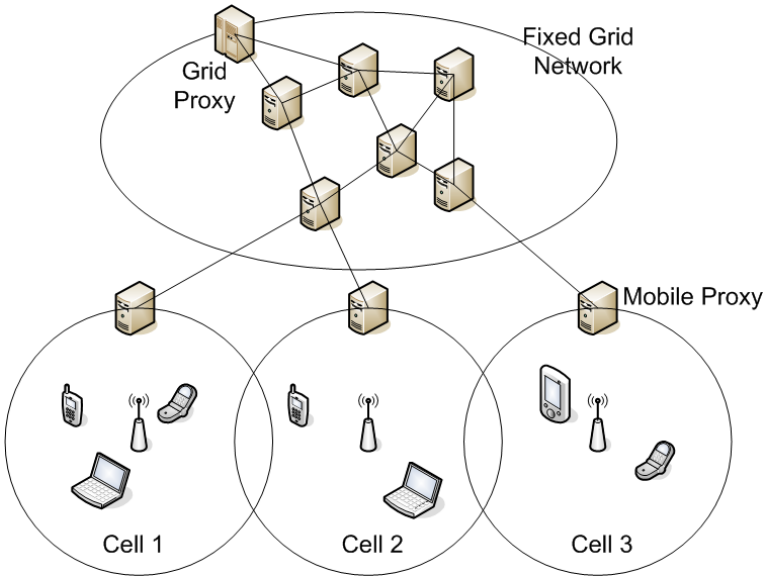

Figure. 1. Proxy-based mobile grid architecture

Figureure 1 shows a proxy-based mobile grid architecture. This architecture includes a cluster of mobile devices. The cluster is connected to a base station that acts as a router/node or a grid proxy server on the grid. The base station has two functions. It works as a wireless access point (AP) and as a proxy server that is responsible for data transfer, QoS, and resource access policy. In the proxy-based mobile grid architecture, mobile devices use a base

station to access the grid, and base stations as proxy-servers allocate the tasks to mobile devices [9,10,11,12,13,14]. On the contrary, the agent-based mobile grid architecture [11,12] uses mobile agents to provide, share, and access resources in grid networks.

Both architectures mainly focus on utilizing wireless networks for grid computing without considering how to manage mobile devices. A successful mobile grid architecture requires dealing with mobility management, disconnected operation, device heterogeneity, service discovery, and resource sharing.

2.2. IP-Paging

Registration and paging techniques are important functions in cellular networks to minimize the signaling overhead and enhance the efficiency of mobility management. However, early mobile IP supports only registration function. Thus, mobile IP users do not actively communicate most of the time; i.e., they are often in an idle state. Figureure 2 shows the state transition diagram of mobile devices.

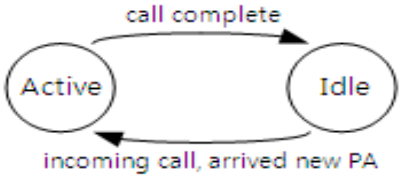

Figure. 2. State transition diagram of mobile device

IP-paging is one way to provide a more scalable and efficient location tracking scheme [6, 7]. It is a procedure that allows a wireless system to search an idle mobile device when a message is destined for it so that the mobile user does not need to register the precise location with the system whenever it moving. The IP-paging has two major benefits: reduction of signaling overhead and reduction of power consumption [6,7,8]. The reduction of power consumption is an important issue because grid tasks are executed on the mobile devices with limited battery capacity.

2.3. Checkpoint Technique in Grid Computing

Establishing an efficient task migration is very important in mobile grid environments. If one mobile device fails while executing a task, the task migration transfers the task to another mobile device and the task can be re-executed from the position of checkpoint. However, the task migration results in the increase of communication costs to save the state of the executing task on a stable storage. So, the task migration scheme does not guarantee better performance than the re-execution scheme [15,16,17]. The checkpoint technique in grid computing is divided into two types: system level and user-defined. The system level checkpoint is not suitable for heterogeneous resources. Meanwhile, two kinds of checkpoints for the user-defined checkpoint can be applied to the heterogeneous resources: coordinated checkpoint and independent checkpoint. However, coordinated checkpoint is not suitable for the mobile grid environment because the coordination process entails additional communication costs.

3. System Architecture

Mobile grid computing is based on a wireless network structure, in which each mobile device communicates with the destination through a base station. Wireless grid network is connected to a fixed grid network through a gateway router. A proxy server located on the gateway router manages send/receive tasks. In this paper, we propose a wireless grid computing architecture based on IP-paging to enable the use of idle mobile devices as grid resources and address the problem of limited battery capacity. Figureure 3 shows the wireless grid computing architecture based on IP-paging.

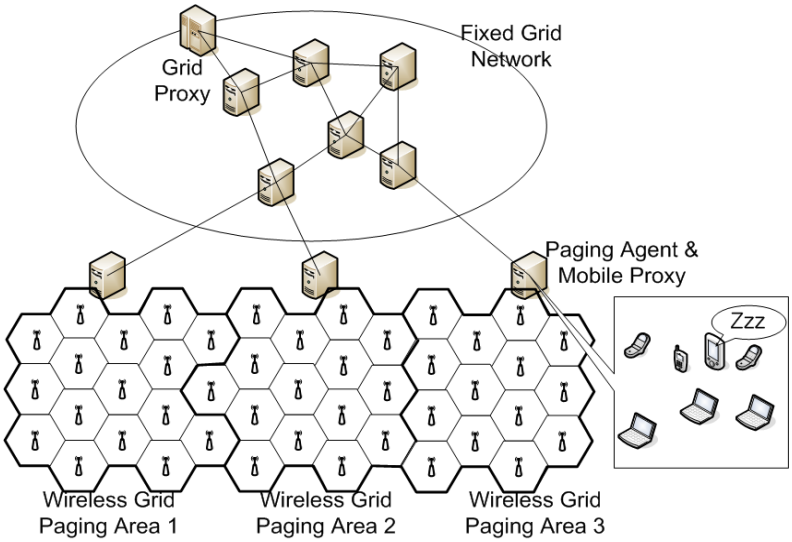

Figure. 3. Wireless grid computing architecture

The proposed wireless grid computing architecture consists of fixed grid networks and wireless grid networks. Grid users join a grid network through a grid proxy that operates as a pure meta-scheduler. It performs three operations: dividing a submitted job into small-size tasks, assigning the tasks to mobile proxies, and collecting finished tasks. A mobile proxy allocates tasks to mobile devices when they are assigned by the grid proxy. It has three important functions: a paging agent for resource management, a pure proxy for task distribution, and a stable storage for checkpoints.

4. Resource Management

In our architecture, resources are managed by the paging information of mobile devices. Each mobile device falls into one of two states: active and idle.

The mobile device registers with the paging agent when it joins the paging area of a new wireless grid or when it changes from the active to the idle state. The mobile device is removed from the paging agent when it leaves current paging area in a wireless grid or when it changes from the idle to the active state. In IP-paging scheme, a paging agent manages the address information of idle mobile devices. In this paper, we extend the IP-paging message format to manage resource status information as well as address information. Resource status information includes information on computing elements and storage elements. The information of computing elements includes class, CPU speed, number of CPUs, RAM size,

and power. The information of storage elements includes capacity and free space. Figureure 4 shows the grid paging registration message format. The resource information collected from mobile devices is stored in the paging cache.

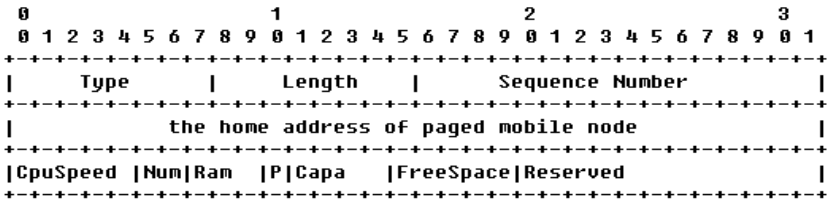

Figure. 4. Grid paging registration message format

Figureure 5 shows an example of the grid paging cache.

Num	Home address	Type	CpuSpeed	NumCpu	RamSize	Power	Capacity	FreeSpace	used
1	163.152.91.77	L	20	1	1024	C	10240	2048	A
2	163.152.91.78	L	27	1	2048	82	10240	2048	A
3	163.152.91.79	P	13	1	512	70	10240	2048	A
4	163.152.91.80	P	10	1	256	90	10240	2048	A
5	163.152.91.81	L	30	2	4096	C	10240	2048	U
-	----------	--	------	----	-----	---	------	------	--

Figure. 5. Example of grid paging cache

Network partitioning occurs in wireless environments because of the random movement of mobile devices. When tasks are assigned by a grid proxy, a mobile proxy creates a pre-candidate set of mobile devices based on the resource information in the paging cache. The mobile proxy sends a paging request message to the mobile devices in the pre-candidate set to confirm their state. The mobile proxy then receives paging reply messages from the mobile devices in pre-candidate set and decides upon a final resource set. The paging reply message contains additional information including the current address of the mobile device so that tasks can be sent to it. To prevent resource duplication, running mobile devices are so marked in the paging cache. Figureure 6 shows the grid paging reply message format.

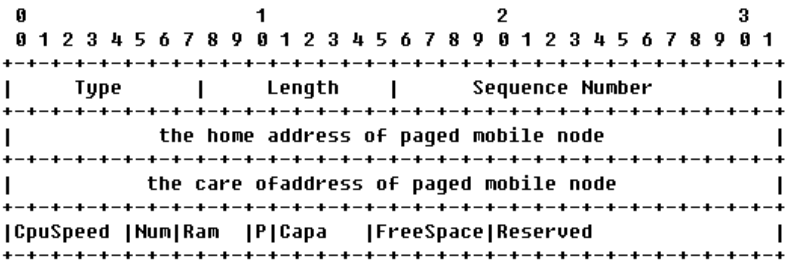

Figure. 6. Grid paging reply message format

Although we use IP-paging scheme in this paper, this does nothing to address the problems with network partitioning and battery discharge in wireless grid networks. Therefore, we propose a task migration scheme using user-defined checkpoints in the next section.

5. Task Migration Using User-defined Checkpoints

In this paper, we propose a user-defined checkpoint technique that performs checkpoints under two conditions: when a mobile device leaves its current cell (determined by analysis of the signaling strength) and when a mobile device turns off due to lower battery capacity.

5.1. Analysis of Wireless Signaling Strength

The wireless signaling strength is the strength of signal from the AP received at a mobile device. In wireless networks, handoff occurs when a mobile device moves to a new cell that is within signaling range of a new AP. When handoff occurs, mobile device's wireless connection with the current cell is terminated, and a new connection is established in the new cell. Depending on the condition of the wireless connection during handoff (e.g., when moving into a fringe area), the mobile device can disappear. In mobile grid computing, the disappearance of a mobile device during operation is an important issue because any task assigned to it must be started over again from the initial state. Therefore, checkpoints are necessary to maintain efficient operation. We suggest performing a checkpoint before handoff occurs. Figureure 7 shows the wireless signaling strength during handoff.

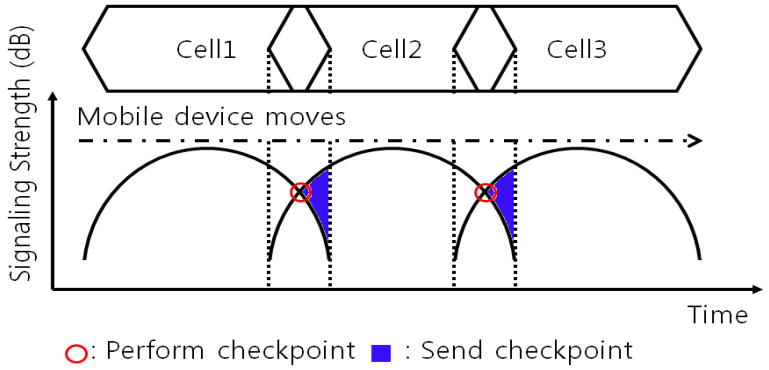

Figure. 7. Wireless signaling strength during handoff

Table 1 shows checkpoint operation according to signaling strength.

Table. 1. Checkpoint operation according to signaling strength

1: If current signaling strength is equal to new signaling strength
2: Perform checkpoint (current state)
3: Else if current signaling strength is less than new signaling strength
4: Send checkpoint to mobile proxy

5.2. Analysis of Battery Capacity

Mobile devices have limited battery capacity and can be classified as: charging and running. In a mobile grid environment, the charging state is the best case because a mobile device that is charging is not mobile, just like a wired device. It can be changed into the running state at any time. If a task is assigned to an idle mobile device in the running state, the battery consumption is nearly five times that of an idle mobile device without an assigned

task. In this paper, we perform a checkpoint when the battery capacity is within the range defined by the provider. Figureure 8 shows the battery capacity as a function of time.

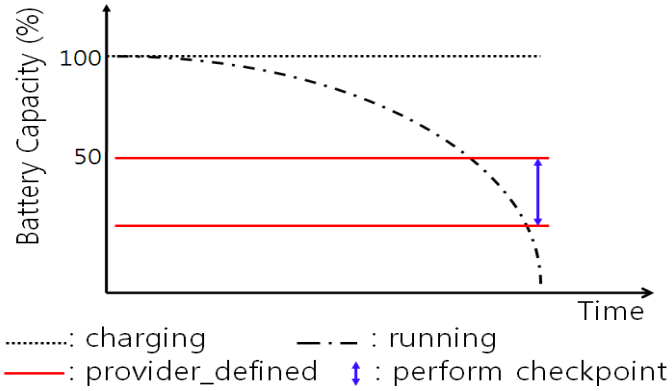

Figure. 8. Battery capacity as a function of time

Table 2 shows checkpoint operation according to battery capacity.

Table. 2. Checkpoint operation according to battery capacity

1: If current battery capacity is equal to battery capacity of provider defined
2: Perform checkpoint (current state)
3: Else if current battery capacity is less than battery capacity of provider defined
4: Send checkpoint to mobile proxy

5.3. Task Migration

Task migration occurs when a mobile device cannot execute the assigned task any longer. This can be due to network partition, discharged battery, or other reasons. The task migrates from one mobile device to another to minimize the completion time of the task. Table 3 describes broker service (task assignment), task migration, and re-execution phases along with an explanation of task completion time.

Table. 3. Broker service, task migration, re-execution phases, and task completion time

Broker service phase
1: Wait for tasks from grid proxy
2: Request grid paging cache
3: Decide pre-candidate set from paging cache
4: Send grid paging request current status to pre-candidate set
5: Wait for grid paging reply from pre-candidate set
6: Decide candidate set from pre-candidate set

7: Decide reserved mobile devices from candidate set

8: Send task to each mobile device

9: Request to change Used fields "U" and "N" in paging cache

10: If processing mobile device has disappeared, compare task completion times

Task migration phase

11: If completion time$_{checkpoint}$ is less than completion time$_{re\text{-}execution}$

12: Decide pre-candidate set from paging cache

13: Send grid paging request current status to candidate set

14: Wait for grid paging reply from candidate set

15: Decide reserved mobile device from candidate set

16: Migrate task to reserved mobile device

17: Request to change Used fields "U" and "N" in paging cache

Re-execution phase

18: If completion time$_{checkpoint}$ is greater than completion time$_{re\text{-}execution}$

19: Decide pre-candidate set from paging cache

20: Send grid paging request current status to candidate set

21: Wait for grid paging reply from candidate set

22: Decide reserved mobile device from candidate set

23: Send task to reserved mobile device

24: Request to change Used fields "U" and "N" in paging cache

Completion time

The completion time =

scheduling time + task assigned time + processing time

+ remaining time + collection time

The completion time$_{checkpoint}$ =

scheduling time + task assigned time + processing time

+ checkpoint time + migration time + remaining time + collection time

The completion time$_{re\text{-}execution}$ =

(scheduling time + task assigned time + processing time)

*** re-execution + collection time**

6. Performance Analysis

Figureure 9 shows the network model used for our simulations. The network model includes six routers and three gateway routers on an Ethernet-based fixed grid network. Each wireless grid paging area has 14-18 cells. Mobile devices have a wireless link (11Mbps, 802.11b) with an AP in each cell. The AP and gateway router in the fixed grid network are connected via an Ethernet-based Internet.

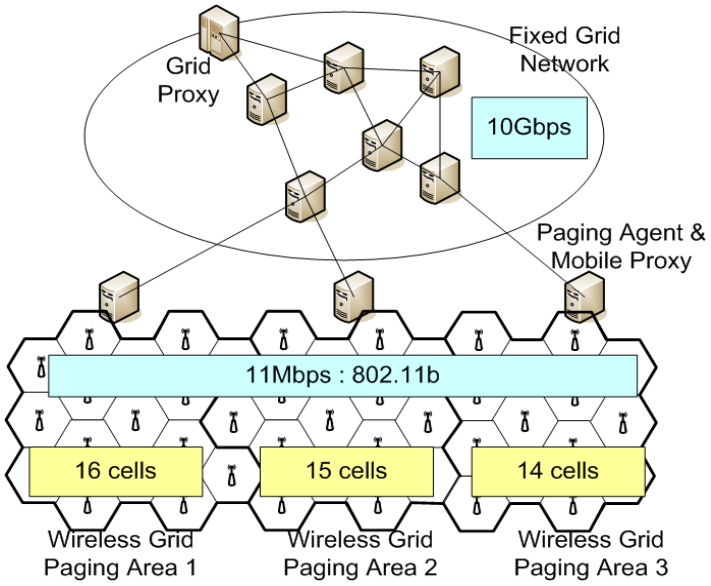

Figure. 9. Network model for performance analysis

The purpose of our simulations is to quantitatively evaluate the improvements in total task execution time and total job completion time in a system using the proposed enhancements, when compared with the architecture with no paging. Two parameters are studied: task execution time and job completion time. The task execution time is defined as the time that elapses between starting and finishing the task execution. We study the task execution time as a function of the number of available mobile devices and the probability of handoff. The job completion time is defined as time between scheduling a job and collecting the finished job. We study the job completion time as a function of the number of available mobile devices and the probability of handoff. For performance evaluation, a job that is divided into 20 tasks is executed on 20 mobile devices. We consider the following five cases:

- Case 1: 50 available mobile devices with 0% probability of handoff
- Case 2: 50 available mobile devices with 30% probability of handoff
- Case 3: 50 available mobile devices with 50% probability of handoff
- Case 4: 100 available mobile devices with 30% probability of handoff
- Case 5: 100 available mobile devices with 50% probability of handoff

Figureure 10 shows the total task execution time as a function of the number of available mobile devices and the probability of handoff. And, the ratio of task migration/re-execution is presented form handoff on both executing mobile devices and standing mobile devices.

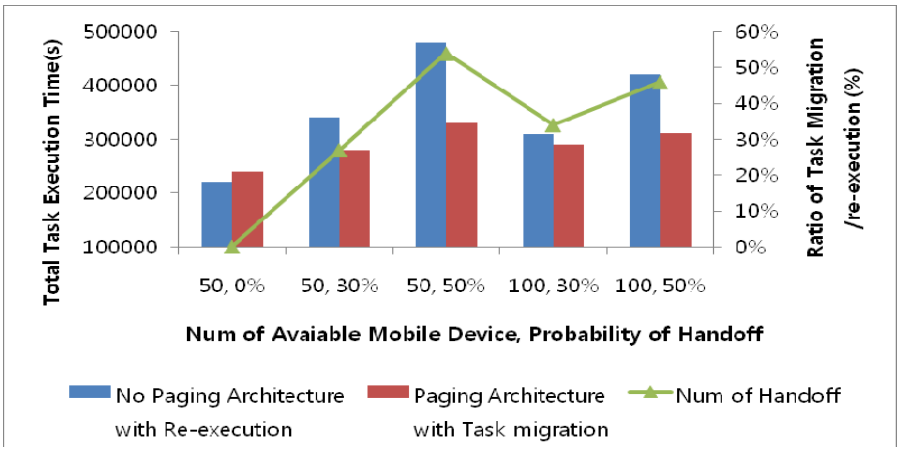

Figure. 10. Total task execution time as a function of the number of available mobile devices and the probability of handoff

The no-paging architecture was faster in Case 1 than the proposed paging architecture. This is because the proposed paging architecture spends additional time for paging-cache management, network delay due to paging request, and the paging reply message. And the ratio of task migration/re-execution is 0%, because there is no handoff in Case 1. As for Case 2, the proposed paging architecture was faster than the no-paging architecture. Because of the 30% probability of handoff, the tasks assigned to some mobile devices were re-executed. Additionally, our task migration scheme performed better than simple re-execution. The ratio of task migration/re-execution is 27%. The proposed paging architecture was faster in Case 3 than the no-paging architecture for the same reason as Case 2, and the difference in times was much greater than in Case 2. Because of the 50% probability of handoff, tasks assigned to many mobile devices were re-executed. And the ratio of task migration/re-execution is 54%. As we can see that the ratio of task migration/re-execution is rapidly increased by the increase of probability of handoff. The paging architecture was faster than the no-paging architecture in Case 4 as well, although the difference in times was much less than in Case 2. Due to the limited number of available mobile devices, neither paging architecture was able to find suitable new mobile devices easily. The proposed paging architecture had similar performance in Cases 2 and 4, due to the use of the pre-candidate set. The ratio of task migration/re-execution is 34%. The proposed paging architecture in Case 5 was faster than the no-paging architecture. Because of the 50% probability of handoff, tasks assigned to many mobile devices were re-executed. And the ratio of task migration/re-execution is 46%.

Figureure 11 shows the total job completion time as a function of the number of available mobile devices and the probability of handoff. The job scheduling time and network delay for sending the assign job and gathering the finished job were added to the total task execution time. Because we used a static value for the job scheduling time and the network delay for sending the assign job and gathering the finished job, the results of Figureure 11 have a similar tendency of these of Figureure 10. The no-paging architecture was faster in Case 1 than the proposed paging architecture. This is because the proposed paging architecture spends additional time for paging-cache management, network delay due to paging request, and the paging reply message. As for Case 2, the proposed paging architecture was faster than the no-paging architecture. Because of the 30% probability of handoff, the tasks assigned to some

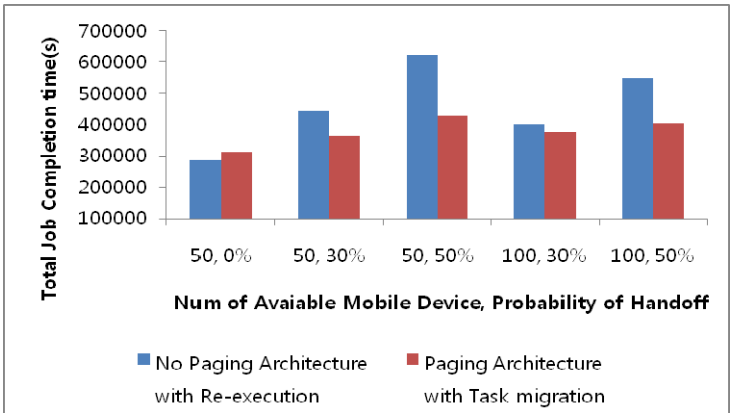

Figure. 11. Total job completion time as a function of the number of available mobile devices and the probability of handoff

mobile devices were re-executed. So, the total job completion time is increased. Additionally, our task migration scheme performed better than simple re-execution. The proposed paging architecture was faster in Case 3 than the no-paging architecture for the same reason as Case 2, and the difference in times was much greater than in Case 2. Because of the 50% probability of handoff, each mobile devices that assign task, performs task migration/re-execution. The paging architecture was faster than the no-paging architecture in Case 4 as well, although the difference in times was much less than in Case 2. Due to the limited number of available mobile devices, neither paging architecture was able to find suitable new mobile devices easily. The proposed paging architecture had similar performance in Cases 2 and 4, due to the use of the pre-candidate set. The proposed paging architecture in Case 5 was faster than the no-paging architecture. Because of the 50% probability of handoff, tasks assigned to many mobile devices were re-executed. Generally, re-execution technique has less additional operation than task migration technique that performs checkpoint, migration, and so on. However, re-execution technique is usually takes longer time than task migration technique, because re-execution technique always starts initial state of task.

7. Conclusion and Future Work

We have addressed the management of mobile devices in mobile grid environments by focusing on IP-paging, which is capable of managing idle mobile devices and grid resource status information. We proposed the IP-paging architecture for mobile grid environments to support mobile grid services. The proposed architecture can be used to deal with network partition problem, which is an important issue in creating a practical mobile grid environment with a task migration scheme using checkpoints. However, this imposed an additional cost due to task migration required to perform the checkpoint. In addition, being able to predict when a mobile device will leave the network or when its battery will fall below a certain level of charge is very useful in addressing these problems to provide continuous grid services. We focused on a user-defined checkpoint that is performed under two conditions: when a mobile device leaves its current cell and when a mobile device turns off due to lower battery capacity.

In this paper, we attempted to solve mobile resource management using IP-paging and task migration based on user-defined checkpoints. Many challenges still remain including

disconnected operation, job scheduling, device heterogeneity, and security. We will tackle these issues as future works. We will also extend our investigation into what wireless networks can contribute to grid computing in an effort to provide effective support to mobile grid computing.

References

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," Int. Journal of High Performance Computing Applications, vol. 15, no. 3, pp. 200-222, 2001.
[2] Ian Foster, and Carl Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 2004.
[3] I. Foster, A. Roy, and V. Sander, "A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation", 8th International Workshop on Quality of Service, pp. 181-188, 2000.
[4] I. Foster, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Global Grid Forum, 2002.
[5] I. Foster, "The Grid: A New Infrastructure of 21st Century Science," Physics Today, vol. 55, pp. 42-52, 2002.
[6] X. Zhang, J. G. Castellanos, and A. T. Campbell, "P-MIP: paging extensions for mobile IP," Mobile Networks and Applications, vol. 7, no. 2, pp. 127-141, April 2002.
[7] A. T. Campbell and J. Gomez, "IP Micro-Mobility Protocols," ACM SIGMOBILE Mobile Computer and Communication Review (MC2R), vol. 4, no. 4, pp. 45-54, Oct. 2001.
[8] C. Castelluccia, "Extending Mobile IP with adaptive individual paging: A performance analysis," INRIA, http://www.inrialpes.fr/planete/people/ccastel/, 1999.
[9] P. Ghosh, N. Roy, S. K. Das, and K. Basu, "A Game Theory based Pricing Strategy for Job Allocation in Mobile Grids," Proceedings of 18th International Parallel and Distributed Processing Symposium, pp.26-30, April 2004.
[10] V. Hingne, A. Joshi, T. Finin, H. Kargupta, and E. Houstis, "Towards a Pervasive Grid," Proceedings of 17th International Parallel and Distributed Processing Symposium, pp. 22-26, April 2003.
[11] M. Fukuda, Y. Tanaka, N. Suzuki, Lubomir F. Bic, and S. Kobayashi, "A mobile-agent-based PC grid," Proceedings of Autonomic Computing Workshop Fifth Annual International Workshop on Active Middleware Services (AMS'03), pp. 142, June 2003.
[12] R. M. Barbosa and A. Goldman, "MobiGrid Framework for Mobile Agents on Computer Grid Environments," MATA 2004, LNCS 3284, pp. 147-157, 2004.
[13] T. Phan, L. Huang, and C. Dulan, "Challenge: Integrating Mobile Wireless Devices into the Computational Grid," Proceedings of 8th International Conference on Mobile Computing and Networking, pp. 271-278, Sept. 2002.
[14] S. Wesner, J. M. Jähnert, M. Aránzazu, and T. Escudero, "Mobile Collaborative Business Grids: A short overview of the Akogrimo Project," http://www.akogrimo.org/download/ White_Papers_and_Publications/ Akogrimo_WhitePaper.
[15] G. Allen, D. Angulo, I. Foster, G. Lanfermann, C. Liu, T. Radke, and E. Seidel, "The Cactus Worm: Experiments with dynamic resource discovery and allocation in a grid environment," International Journal of High Performance Computing Applications, vol. 15, no. 4, pp. 345-358, 2001.
[16] P. Roe and C. Szyperski, "Transplanting in Gardens: Efficient Heterogeneous Task Migration for Fully Inverted Software Architectures," Proceedings of the Fourth Australasian Computer Architecture Conference, Jan. 1999.
[17] S. Krishnan and D. Gannon, "Checkpoint and Restart for Distributed Components in XCAT3," Proceedings of the fifth IEEE/ACM International Workshop on Grid Computing, pp. 281-288, Pittsburgh, Pennsylvania, Nov. 2004.

Authors

DaeWon Lee received the Ph.D. degree in Computer Science Education from Korea University, Korea, in 2009. He is currently a full time lecturer in the Division of General Education at SeoKyeong University in Korea. His research interests are in Grid computing, Mobile computing, Distributed computing and Fault-tolerant systems.