

A Resource Discovery Algorithm in Mobile Grid Computing based on IP-paging Scheme

Yue Zhang, Yunxia Pei

► **To cite this version:**

Yue Zhang, Yunxia Pei. A Resource Discovery Algorithm in Mobile Grid Computing based on IP-paging Scheme. IFIP International Conference on Network and Parallel Computing (NPC), Sep 2010, Zhengzhou, China. pp.402-411, 10.1007/978-3-642-15672-4_33 . hal-01054969

HAL Id: hal-01054969

<https://hal.inria.fr/hal-01054969>

Submitted on 11 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Resource Discovery Algorithm in Mobile Grid Computing based on IP-paging Scheme

Yue Zhang¹ and Yunxia Pei²

Center of Network, Henan Police College, Zhengzhou, China

Department of Math and Computer Science¹

Zhengzhou University of Light Industry, Zhengzhou China²

Abstract. The goal of this work is to enable mobile devices as a resource in the Grid network. We propose an IP-paging scheme based on the mobile grid computing framework to manage idle mobile devices. Within this framework, we discuss several research issues such as location tracking using IP-paging, limiting power using the Grid page cache, and service migration using reserved resources. We also describe a mobile proxy and a prototype for a paging agent.

Keywords: Mobile Grid Computing, IP-Paging, Resource Discovery Algorithm

1 Introduction

Grid computing differs from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and high-performance focus [1]. In Grid computing, the term resource management refers to the operations used to control how capabilities provided by Grid resources and services are made available to other entities such as users, applications, or services [2]. A resource manager is one of the most critical components of the grid middleware [4], since it is responsible for resource management that provides resource selection and job scheduling. Therefore, resource discovery, resource selection, and job scheduling have considerable influence on computing performance and are important issues in Grid computing.

In the early period of grid computing, most researches have focused on a fixed network [1, 2, 3, 4, 5]. With the improvements on Internet technology, the use of mobile devices is rapidly increasing. Researches in Grid computing [6] tried to combine Grid computing with mobile devices. However, according to properties of mobile environments, the management of mobile devices deserve many careful considerations, such as mobility management, disconnected operation, device heterogeneity, service discovery, resource sharing, and so on.

There are two approaches in mobile grid computing to address these problems: proxy-based mobile grid architecture [6] and agent-based mobile grid architecture. There are various solutions proposed in these two kinds of architectures. However, the existing solutions cannot solve the mobility management problems completely.

2 Related Works

This section briefly surveys the background on mobile grid computing and IP-paging, which is employed in our proposed architecture.

2.1 Mobile Grid Computing

There are two architectures for mobile grid computing. One is a proxy-based mobile grid architecture, and the other is an agent-based mobile grid architecture. Proxy-based mobile grid architectures to support lots of mobile devices in computational grid are proposed in [6]. Fig. 1. shows the proxy-based mobile grid architecture.

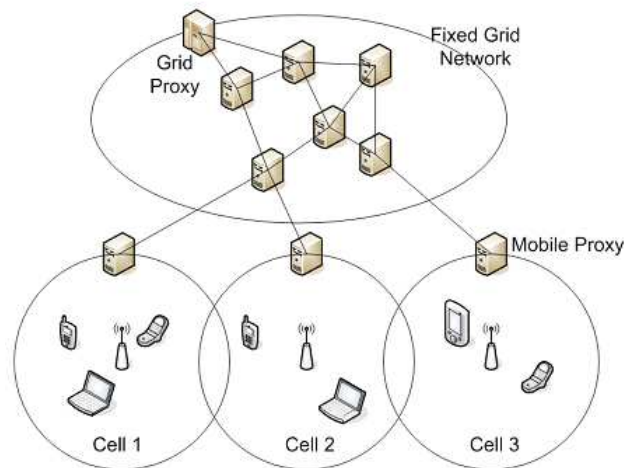


Fig. 1. The proxy-based mobile grid architecture

In this architecture, there is a cluster of mobile devices. The cluster is connected to a base station that is a router/node within grid or a grid proxy server. The base station has two functions. It works as a wireless access point and a proxy server that is responsible for data transfer, QoS, and resource access policy. The mobile devices use the base station to access grid, and the base station allocates the jobs requested from mobile devices to grid computing system. However, in the proxy-based architecture, mobile devices have no access to grid resources.

The agent-based grid architectures use mobile agents to provide, share and access distributed resources in grid network. Grid clients submit computing jobs to the computational grid, and the mobile agents carrying the user programs of the jobs travel among the remote hosts to search for available resources. By taking advantages of the mobile agent's mobility and autonomy features, a job can be executed at any suitable machines and may migrate to other nodes if necessary. The service tasks can migrate to different idle machines during their execution. In this way, load balancing and sharing of distributed computing resources in the computational grid can be achieved.

2.2 IP-paging

IP-paging is a procedure that allows a wireless system to search for idle mobile devices when there is a message destined for it, such that the mobile user does not need to register its precise location to the system whenever it moves. Paging has two major benefits:

- reducing signaling overhead
- reducing power consumption.

Fig. 2. shows basic paging reply message format in IP-paging.

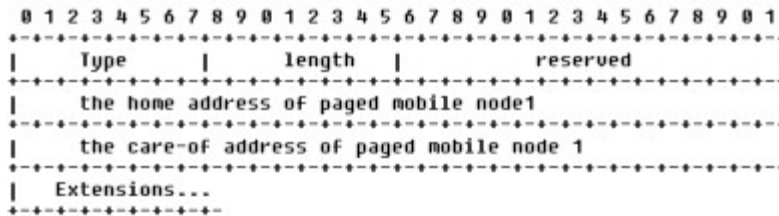


Fig. 2. IP-paging reply message format

In mobile grid computing environment, since grid tasks are executed at mobile devices, reducing power consumption is important issue to use a mobile device to perform such tasks.

3 Wireless Grid Paging Architecture

In this paper, we propose a wireless grid computing architecture based on IP-paging. We tried to enable dormant mobile devices as a grid resource and to solve limited battery problem. Fig. 3. shows the wireless grid paging architecture.

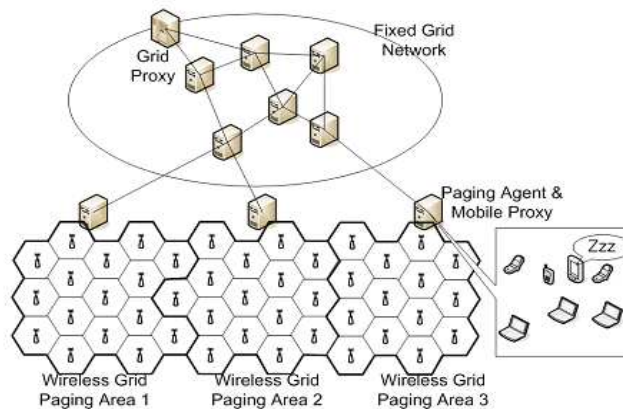


Fig. 3. The wireless grid paging architecture

In this wireless grid computing architecture, the movements of mobile device are

managed by a grid proxy or a grid controller. The grid proxy in our proposed architecture performs scheduling operation as a pure meta-scheduler. It performs following three operations: dividing a submitted job into tasks, assigning tasks to mobile proxy, and collecting the accomplished tasks.

3.1 the Paging Agent

In previous wireless grid computing architecture, the movement between cells is managed by a fixed grid network. However, our architecture doesn't consider the movement between cells of the mobile devices. It is because we use only idle mobile devices. Instead, we consider only the status of mobile devices and the movements between paging areas.

In IP-paging scheme, the status of a mobile device is classified into two categories: the active state and the idle state. The mobile device is registered to paging agent with following two conditions: joining paging area of new wireless grid and changing from the active state to the idle state.

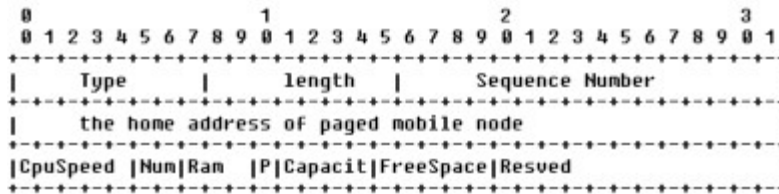


Fig. 4. The grid paging registration message format

Fig. 4. illustrates the grid paging registration message format. In IP-paging scheme, paging agent manages address information of an idle mobile device. In this paper we extend IP-paging message format to manage address information and resource status information.

3.2 Mobile Proxy

When tasks are assigned from a grid proxy, a mobile proxy composes a pre-candidate set by resource information on paging cache. And, to confirm status of resource, the mobile proxy sends paging request message to the mobile devices on the pre-candidate set. Then, the mobile proxy receives paging reply message from mobile devices on pre-candidate set, and decides final resource set. There is additional information in paging reply message that is the current address of mobile device that will issue the computing task. And, to prevent resource duplication, running mobile devices are marked on paging cache. Fig. 5. shows the grid paging reply message format.

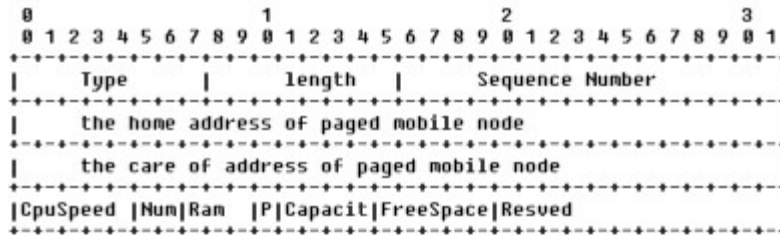


Fig. 5. The grid paging reply message format

4 Mobile Proxy and Paging Agent

Fig. 6. shows the components of a mobile proxy and the paging agent. The proposed mobile proxy and paging agent consist of grid middleware, ontology service, and UDDI registry.

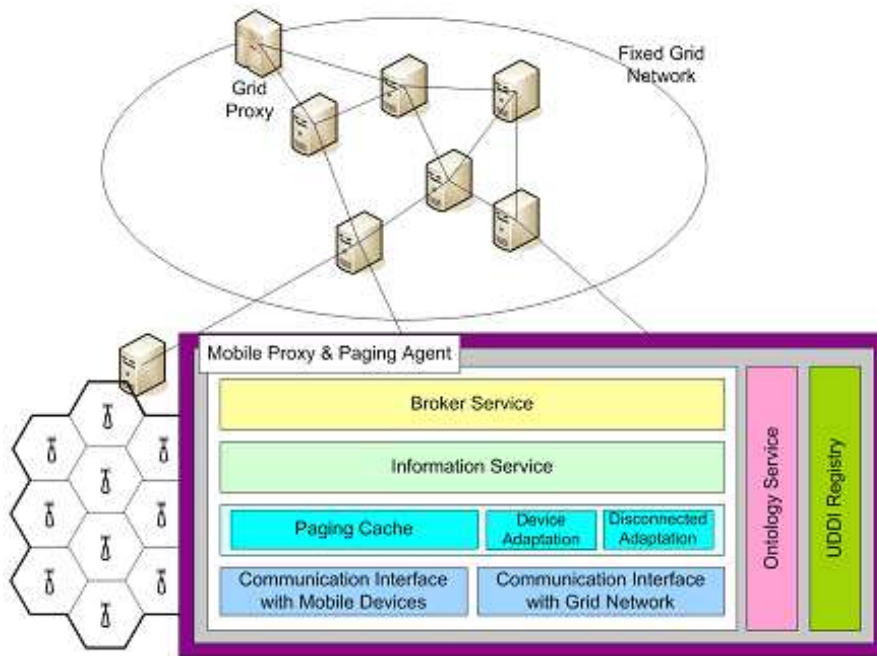


Fig. 6. The mobile proxy and paging agent

The grid middleware consists of the communication interfaces, a paging cache, an information service, and a broker service.

4.1 The Communication Interface

There are two communication interfaces: one is that between the mobile device and the mobile proxy, and the other is between the mobile proxy and the paging agent. The communication interface with mobile devices communicates with mobile devices by SOAP framework. The communication interface with grid network communicates with a fixed grid network and allows using the standard grid protocol (ex. GRAM, MDS, GIS).

4.2 Paging Cache

Mobile grid environment intends free communication between mobile devices and a fixed grid network. But it isn't possible, because there are different types of devices, interfaces, CPUs, capacity in the mobile environment. As shown in the section 3.2, the mobile proxy and paging agent manages mobile device status in its grid paging area. Fig. 7. shows the example of grid paging cache.

Num	Home Address	Type	CpuSpeed	NumCpus	RanSize	Power	Capacity	FreeSpace	Used
1	163.152.91.77	L	20	1	1024	R	10240	2048	N
2	163.152.91.77	L	16	1	512	40	10240	2048	N
3	163.152.91.77	P	13	1	256	70	10240	2048	N
4	163.152.91.77	L	30	2	2048	R	10240	2048	N
5	163.152.91.77	L	20	1	512	R	10240	2048	N
-	-	-	-	-	-	-	-	-	-

Fig. 7. The example of grid paging cache

The *Home Address* field denotes mobile device's home address. The *Type* field denotes type of a mobile device (i.e., L means laptop, P means PDA, C means cellular phone). The *CpuSpeed* field denotes CPU speed of a mobile device. The *NumCpus* field denotes the number of CPUs in a mobile device. The *RamSize* field denotes RAM size in a mobile device. The *Power* field denotes the remaining power of a mobile device (e.g., 40 means 40%, R means recharged). The *Capacity* field denotes all capacity of mobile devices. The *FreeSpace* field denotes usable capacity of mobile devices. And, the *Used* field denotes a mobile device is using or not in another grid operations (i.e., N means none, U means used).

4.3 Information Service

There are lots of grid services and grid resources in grid network. To provide grid services, service discovery and resource provision is needed. The globus toolkit is representative grid infrastructure to provide service discovery and resource provision. The principle of service discovery and the resource provision scheme of globus toolkit is that when a new job or task is assigned, it discovers and decides a suitable service or resource among all reserved services or resources. However, this scheme of globus is incongruent to mobile grid environment, due to the device mobility.

The proposed service discovery and resource provision scheme is based on paging

cache discussed in the section 4.2. When a new task is assigned, the paging agent composes a pre-candidate set by resource information on paging cache. And, to confirm the status of resource, the mobile proxy sends paging request message to the mobile devices on the pre-candidate set. The service discovery and the resource provision steps follow the grid paging registration/replay procedure discussed in the section 3.3. Table 1 shows the query processing.

Table 1. The Query Processing

ex) Q={R{R1 RN} Type = laptop and CpuSpeed 2.0 and RamSize 512 and Power = recharged and }
--

4.4 Broker Service

The broker service in mobile proxy and paging agent performs two operations: The first operation is delivery of an assigned task from mobile device to grid proxy. And, the second operation is to schedule an assigned task from grid proxy to mobile resources. Table2. shows the operation of broker service.

Table 2. The Operation Of Broker Service

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 to current status of pre-candidate set 5: Wait grid paging reply from pre-candidate set 6: Decide candidate set from pre-candidate set 7: Decide reserved mobile devices from pre-candidate set 8: Send task to each mobile devices of candidate set 9: Request to change used field U and N to paging cache 10: If processing mobile device changes its state to active or its battery status to discharging then 11: Receive processing data 12: Send processing data to reserved mobile device 13: Wait for tasks from each mobile devices

5 Performance Evaluations

5.1 Network Model

Fig. 8. shows the network model.

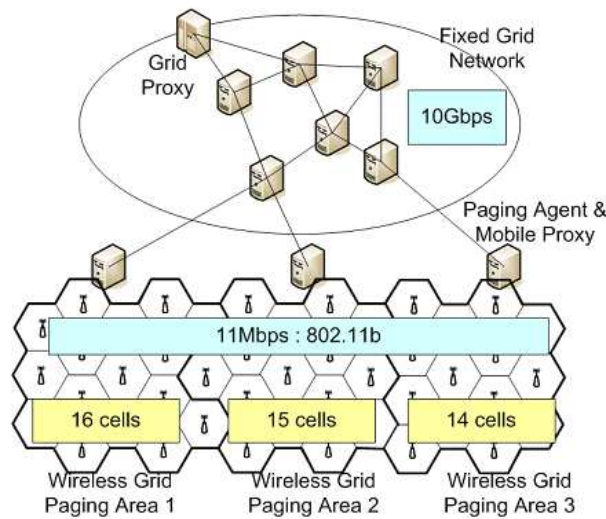


Fig. 8. The network model

The network model used for all the simulations includes six routers and three gateway routers on the Ethernet (10Gbps)-based fixed grid network. Each wireless grid paging area includes 14-18 cells. Mobile devices have a wireless link (11Mbps:802.11b) with the access point on each cell. Access point and gateway router of a fixed grid network are connected to each other through Ethernet-based internet.

5.2 Class Component

Fig. 9. shows the class of components.

The *Grid_Proxy* is located on a fixed grid network. And the *Mobile_Proxy* is located on a wireless grid network. The *Paging_Agent* is located with *Mobile_Proxy*. It makes paging cache by each mobile device's *MD_Status*. And the *Mobile_Device* is invoked by *MD_manager*.

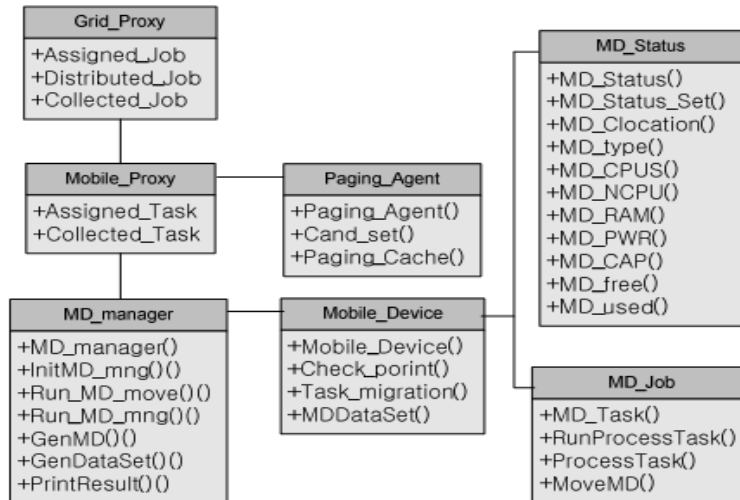


Fig. 9. The class of components

5.3 Performance Metrics

The purpose of the performance comparison is to quantitatively evaluate the improvements that total tasks would experience in a system using the proposed enhancements in comparison to the proxy-based scheme.

The parameters to be studied are as follows:

- Task execution time: Task execution time is defined for an executing task as the time that elapses between the starting task execution and the finishing task execution. We study task execution time for an increasing number of total tasks.
- Task loss: Task loss is defined for an executing task as the task loss by movement or power discharging. We study task loss for random movements and power discharging on execution time.

5.4 Evaluation and Discussion

We present the results of the impact of execution time in Fig. 10 with the following parameters: total tasks, starting task, movement, battery discharging, and task migration.

For all tasks, the proposed scheme shows a slight better performance than a proxy-based scheme. Because the proposed scheme selects idle mobile devices, selected mobile devices have low probability of movement or active state during execution. We can observe that the proposed scheme is slowly increased and the proxy-based scheme has rough variation, because mobile devices that the proxy based scheme selected, re-execute many times by movement and battery discharging.

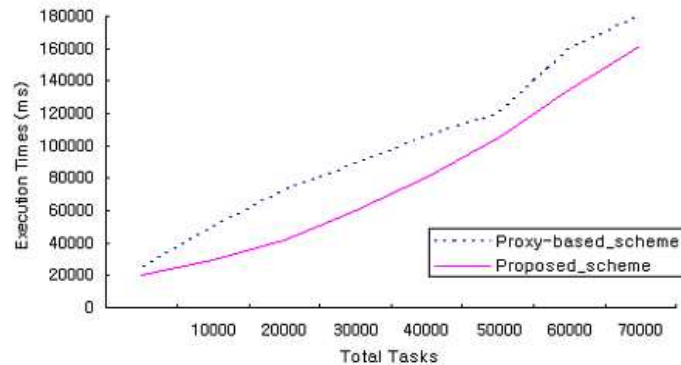


Fig. 10. The total execution time on total tasks

6 Conclusion and Future Works

In mobile grid computing, there are many problems to be solved, e.g., disconnected operation, location tracking, limited power, service migration, job scheduling, device heterogeneity, security, and so on. In this paper, we tried to solve location tracking using IP-paging, limited power using grid paging cache, and service migration using reserved resource. But there are still challenging problems such as disconnected operation, job scheduling, device heterogeneity, security and so on. We plan to tackle these issues in future works and develop mobile grid computing system based IP-paging. Moreover, we plan to extend our investigation into wireless network and implications in grid computing and try to provide effective mobility support computing.

References

1. S I. Foster, C. Kesselman, S. Tuecke, The Anatomy of the Grid :Enabling Scalable Virtual Organizations, *Int. Journal of Supercomputer Applications*, 3 (2001).
2. Ian Foster, Carl Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers (2008).
3. I. Foster, A. Roy, V. Sander, A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation, 8th International Workshop on Quality of Service (2000)
4. Foster I. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, *Global Grid Forum* (2009).
5. Foster, I. The Grid: A New Infrastructure of 21st Century Science, *Physics Today*, Vol.55, pp. 42-52 (2002).
6. P. Ghosh, N. Roy, S. K. Das ,K. Basu, A Game Theory based Pricing Strategy for Job Allocation in Mobile Grids, In *Proceedings of 18th International Parallel and Distributed Processing Symposium* (2004).