Proceedings of 2nd National Conference on Challenges & Opportunities in Information Technology (COIT-2008)
RIMT-IET, Mandi Gobindgarh. March 29, 2008

Integration of Mobile Computing with Grid Computing: A Middleware Architecture

Anil Kumar*, Salim Raza Qureshi **
Model Institute of Engg. & Techn, Kot Bhalwal, Jammu
*anil123455@rediffmail.com, **q_salimqureshi@rediffmail.com

Abstract— In this paper we present a proposal of a middleware to integrate mobile computing with grid computing. We describe the support to be provided by the combination of the two technologies, propose a middleware to support the development and management of mobile grids, and discuss how this scenario contributes to the development of a new age of mobile service applications.

I. INTRODUCTION

The development of mobile services involves two problem domains: (i) the problem of the computational infrastructure for the implementation of services, which regards the hardware and software structure that facilitates the connection of mobile devices (devices, data transport system, and others) and service base intrinsic to this scenario (location identification service, fault tolerance services, and others) [5] and; (ii) the problem of service implementation itself, in which the developer challenge is to integrate the available computational infrastructure and create a solution to the problem of the scenario in question. Furthermore, as described in [7], mobile computing imposes a degree of complexity inherent to the environment, such as dynamic environments, mobility, computational resource limitations, latency and instabilities in data transfer, energy supply limitations, and input/output interface limitations.

Several studies are being conducted and approaches are being proposed to circumvent these problems. Amongst these, we can cite advances in the areas of (a) data communication, such as faster and more reliable networks [8]; (b) distributed computing, through processing distribution among client and server [2]; and (c) software engineering, with the creation of applications that adapt their behavior according to the resource limitations [4]. However, to date the proposed methodologies don't offer a complete solution, and in this way they lead to ad hoc solutions, specific to the problem scenario being dealt. The lack of re-usability and homogenization causes incidental costs in the development of these applications.

We believe in the integration of grid computing and mobile computing technology through a middleware for the development of mobile service applications. This middleware integrates the tools of grid computing – e.g., data communication, directory and service distribution, security, resource discovery, and resource allocation – providing the developer a set of reusable and homogeneous resources.

This paper is organized as follows: Section II provides the background of the problem. Section III proposes system architecture and shows how it solves the problem. The paper concludes with section IV.

II. ANALYSIS

Based on the above example, on the characteristics of the necessary computational infrastructure, and on the implementation of the application for this environment, we present the requirements for this problem scenario:

(a) The data communication methods, necessary for data transmission from the sensors and integration of the actuators and personal assistants; (b) the security methods for data transmission and access control to service resources; (c) the method of resource registration that allows, for instance, the sensors or personal assistants entering and exiting control in the system; (d) the device management methods that allows to manage sensors, actuators, and personal assistants in a centralized way, despite the distribution, environment dynamism, and device heterogeneity, (e) the interface method of mobile devices that allows the communication with the users or the connected devices.; and (f) the methods necessary for work coordination between mobile users that allows collaboration between personal assistants.

The support for these requirements is beyond the scope of programming languages (e.g., Java 2 Micro Edition [1]) and requires a software solution that achieves device integration, development homogenization, software reuse, coordination, resource discovery, resolution of addressing problems, security, and other characteristics.

We believe there exists a relation between the solutions provided by Grid Computing [9] and the development of systems for the scenarios of mobile computing, as described before. For instance, mobile computing requires solutions for service discovery in a dynamic environment, establishment of data communication channels between devices and service providers, development tools that allow the integration of devices and distributed resources, and security, such as confidentiality of data and authentication. We suggest that grid computing technology provide the tools necessary for the implementation of these systems.

Initially, we will analyze grid computing software e discuss the solutions provide by them when applied in the development of mobile applications. This analysis will help in the identification of the shortcomings of the existent solutions and position our contribution.

III. RELATED WORK

Grid computing technology offers support for the implementation of mobile services for decision support systems. There are several software packages for the implementation of grid computing services. In this sub-section we will present the ones most known by the community and discuss their support for the creation of mobile services.

As described in [6], the development of decision support mobile services needs the support of: collaboration, user interface, context-awareness, and a inference process that allows the development of systems beyond the purely reactive. In an environment of limited resources, applications must be aware of the computational resources limitations, unreliable and intermittent communication networks, energy supply limitations, mobility, dynamic environments, and reduced user interfaces. These limitations are inherent to the environment and will be soothed, and not superseded, by future technological developments [7]. Based on the requirements described before, we evaluated the existent grid middleware in terms of collaboration support; context-awareness support; resource allocation support; dynamic environments support; and execution on mobile devices support.

Globus [3] is a open source software package developed by the Globus Alliance which offers a application development and grid computing systems toolkit. Its support for mobile service characteristics is found in collaboration through resource layer protocols which can obtain information and also control jobs, promoting collaboration and resource distribution. Also there is resource allocation provided through the resource manager (Globus Resource Allocation Manager – GRAM) which supplies a job sending and monitoring interface.

Gridbus [10], developed by the Grid Computing and Distributed Systems (GRIDS) Laboratory from University of Melbourne, is a open source grid software package for the development of grids for eScience and eBusiness. It uses several other middleware, such as Globus, Unicore [12], Alchemi [15], and others. Its support for mobile services characteristics is found in collaboration, resource allocation, and dynamic environments, provided by lower level middleware or core middleware, such as the ones described before.

Legion [11] is a middleware developed by a project at University of Virginia, and is defined as a meta-system based on objects (resources) with billions of hosts and trillions of objects linked together by high-speed networks, workstations, and supercomputers in a system that can aggregate different architectures, operating systems, and physical locations. The support provided by Legion for mobile service characteristics is seen in collaboration, in which the binding system makes possible the collaboration through tuples such as that supply object addressing (LOID) and management (LOA). There also exists resource allocation, provided by LOA (Legion Object Addresses) that incorporates a physical address as the IP and can distribute these resources through multicast.

UNICORE (UNIform Interface to Computer Resources) [12] is a middleware that integrates grid computing resources through a graphical interface developed in Java. Its support for mobile service characteristics can be found in collaboration provided through the UNICORE servers after the client/user authentication. The collaboration is achieved by the servers sending jobs to peer Unicore gateways, which execute the jobs and respond. The support of resource allocation is achieved by the Abstract Job Object (AJO), which is an object that sends and receives jobs

Table I summarizes the support provided by the middleware packages for the development of mobile computing services. From the related work analysis we conclude that the existent packages do not satisfy the requirements for the creation of mobile services, such as the support of collaboration, context awareness, resource allocation, dynamic environments, and execution on mobile devices.

Therefore, we identify the need for the creation of a Middleware for Mobile Devices that provides the necessary functionality to support the development of mobile service applications.

TABLE I. GRID MIDDLEWARE COMPARISON

	GLOBUS	*GRIDBUS*
Collaboration support	Yes	Yes
Context awareness support	No	No
Resource allocation support	Yes (GRAM)	Yes
Dynamic environment support	No	No
Mobile device execution support	No	Yes

LEGION	*UNICORE*
Yes	Yes
No	No
Yes	Yes
(LOA)	(AJO)
No	No
No	
	No
LEGION	
Yes	*UNICORE*
No	Yes
Yes	No
(LOA)	Yes
No	(AJO)
No	No
LEGION	No

IV. PROPOSAL

In the previous section, we presented our motivation, the analysis of the requirements needed for the implementation of applications that support mobile services, and the necessary characteristics of a mobile grid. Consequently, the question that now arises is:

A. System Architecture

In this sub-section we present the elements and interactions of our proposed **System Architecture** for the integration of mobile devices in computational grids.

GRID-MOBILE ARCHITECTURE

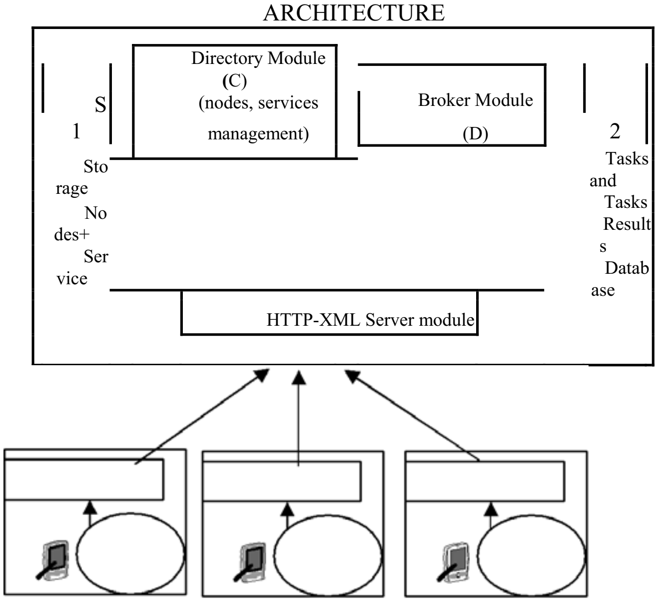

Fig. 1 depicts the architecture. The (A) middleware serves as a bridge that links (B) applications running on mobile devices and the (C, D) grid computing services running on central grid management offices. In these central offices, the (C) Directory Module stores service data in (S1) directory databases. The (D) Broker Module manages the resources, schedules jobs and monitors their execution, storing monitoring data and the job results in (S2) task databases.

To satisfy the requirements listed in the previous section, let's consider the architecture, it's components, and the following conclusions:

i) To satisfy the requirements of (a) data communication methods, the HTTP protocol is used, guaranteeing compatibility between devices.

ii) To satisfy the requirements of (b) security in data transmission and access control, authentication methods and cryptography are used in node registrations

iii) To satisfy (c) resource registration, the Directory Module uses the LDAP [14] protocol and maintains the resource registrations.

iv) To satisfy (d, f) device management and coordination, the Broker Module manages the devices using its management methods, which will be described in sub-section B.

v) To safisfy the (e) interface with the devices, XML technology [13] is used for data representation and visualization at the devices.

B. Application Case Study

In this sub-section we elaborate a case study of an application using the proposed architecture. Consider the following scenario:

A doctor consults a patient data, whom is being monitored by a sensor. The sensor periodically sends the patient's vital data for storage, which is made available to doctors.

For this to happen, the following steps must be attended:

i) Doctor's PDA registration on the grid;
ii) Previous sensor registration;
iii) Data request by the doctor;
iv) Patient search and consulting of the attached sensors
v) Data sent to doctor's PDA.

The following methods and architecture components are involved in the described scenario:

- In (i) and (ii) the method for authenticating the PDA on the grid is registerNode(PDA_type) and the Directory Module executes it.
- In (iii) and (iv) the method for data request is submitTask(searchPatient) which answers the

patientID with his characteristics and sensor capabilities.

- In (v) the method fetchTaskResults(taskid) sends the data and takes into consideration the limitations of the PDA which requested it.

In the described scenario, we verify that the resource homogenization is an advantage for the better management of the environment. In this context, being each resource a grid node, the middleware has through its methods the ability to manage information and physical resources.

The middleware is depicted in Fig. 2. Some of its API methods that enable its interaction with the environment are listed in the appendix.

CONCLUSION

This work proposes an architecture in which its main component, the middleware, adds value to the problem of mobility in grids and provides homogeneous grid resource management. As described in Section III, the proposal makes up to the requirements of mobility, management, data communication, security, and interface, which we believe are the basic requirements of any environment that provides mobility, context-awareness, and anytime/anywhere information. As future work, the middleware can be developed

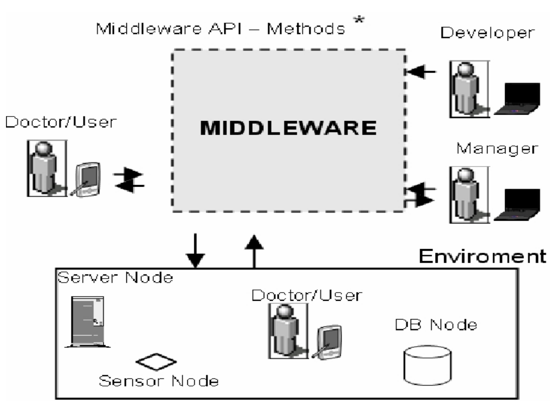

Figure 2. Middleware API and interactions

into a framework in which other modules such as p2p, security etc can be added.

APPENDIX

* Middleware API: Methods

Generals Methods
- registerNode(nodeName, nodeType);
- deleteNode(nodeName);

Submit and Receive Tasks Methods
- submitTask(nodeName, taskAction, taskData);
- checkTaskStatus (taskId);
- fetchTaskResults(taskid);

Management Methods
- getNodes();
- getNodeStatus(nodeName);
- setNode (Node, param);

REFERENCES

[1] Java 2 Micro Edition (J2ME) web-site, sun corporation, http://java.sun.com/j2me.

[2] S. Das and S. K. Das, editors. 5th International Workshop on Distributed Computing, volume 2918 of Lecture Notes in Computer Science, Kolkata, India, December 2003.

[3] I. Foster. Globus toolkit version 4: Software for service-oriented systems. IFIP In ternational Conference on Network and Parallel computing,

[4] E. Guigere. Java 2 Micro edition: The ultimate guide

[5] on programming handheld and embedded devices. John

[6] Springer-Verlag LNCS 3779:pp 2-13, 2005.

[7] E. Guigere. Java 2 Micro edition: The ultimate guide on programming handheld and embedded devices. John Wiley and Sons, Inc., USA, 2001.

[8] K. Henricksen, J. Indulska, and A. Rakotonirainy. Infrastructure for pervasive computing: Challenges. In K. Bauknecht, W. Brauer, and T. A. MÄuck, editors, Informatik 2001: Wirtschaft und Wissenschaft in der Network Economy Visionen und Wirklichkeit, volume 1 of Tagungsband der GI/OCG-Jahrestagung, pages 214-222, Universitat Wien, September 2001.