

A Lightweight Platform for Integration of Mobile Devices into Pervasive Grids

Stavros Isaiadis and Vladimir Getov

Harrow School of Computer Science
University of Westminster, London, U.K.
{S.Isaiadis, V.S.Getov}@westminster.ac.uk

Abstract

For future generation Grids to be truly pervasive we need to allow for the integration of mobile devices, in order to leverage available resources and broaden the range of supplied services. For this integration to be realized, we must reconsider the design aspects of Grid systems that currently assume a relatively stable and resourceful environment. We propose the use of a lightweight Grid platform suitable for resource limited devices, coupled with a proxy-based architecture to allow the utilization of various mobile devices in the form of a single virtual “cluster”. This virtualization will hide the heterogeneity and dynamicity, mask the failures and quietly recover from them, provide centralized management and monitoring and allow for the federation of similar services or resources towards advanced functionality, quality of service, and enhanced performance. In this paper we are presenting the major functional components of the platform.

1. Introduction

Consumer mobile electronic devices such as personal digital assistants (PDA), mobile phones and digital cameras currently hold a very big share of the computer market pie. The trends are very likely to increase in the future, resulting in a very big mobile community. Such devices increasingly provide support for integrated multimedia equipment, intelligent positioning systems, and a diverse range of sensors.

The emergence of the Grid [17, 18] as the new distributed computing infrastructure has accelerated many changes in this field. Distributed systems can now cross several different organizational boundaries and different administrative and security domains, creating what has been termed “a virtual organization”. However, most of these systems do not take into consideration mobile and/or resource limited devices and their integration into the Grid as resource providers and not just consumers is very difficult.

This research work is carried out partly under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

For the Grid community such an integration is an opportunity to utilize available resources in the mobile community, increase its performance and capacity and broaden the range of available services. Current Grid system designers consider this unnecessary and bring in the argument that a couple of extra servers will provide much better performance. But for future generation Grids to be truly ubiquitous we must have the option of integrating mobile devices into Grid systems. In addition, currently, research and industry are focusing on mobile computing and hence mobile devices are destined to become more and more powerful in the near future.

There are a number of current efforts focusing on the integration of mobile devices into the Grid but they do not provide implementation considerations [2, 8], they deal only with infrastructure-less ad hoc networks [1] or they impose restrictions to the programming model or underlying platform [9] that can be used. The potential benefit and challenges of this integration, has been the theme in many other papers and research projects lately [3, 4, 5], but none of these provide any implementation methodology or propose an architecture to support this integration.

The aim of this paper is to present the status of our work in progress on a lightweight Grid platform, suitable for resource limited devices, coupled with a proxy-based architecture and a set of middleware components that will provide the foundations for enhanced functionality, increased reliability (even in unreliable wireless infrastructures) and higher service availability. The lightweight components based Grid platform has been introduced in [10] and is only briefly described here.

The rest of this paper is organized as follows: section 2 presents the lightweight Grid platform and the accompanying architecture. Section 3 briefly describes the most important functional components. Finally, section 4 concludes the paper and lists our future plans.

2. Lightweight Platform

Description

Contemporary Grid implementations/platforms have a very rich set of features – they were designed with built-in exhaustive set of functions. Current standards, software and toolkits for implementing and deploying Grids [11, 12] are also motivated to provide a generic computational Grid with all possible features built-in. The Open Grid Services Architecture [13], on which most of the current Grid platforms are based, is built as a feature rich platform. This approach ensures that service requests from applications are included in a complete set of features offered by the platform.

However, complexity (in terms of interactions, manageability and maintainability) of the implementation of any Grid platform based on this philosophy will be very significant. Additionally, deployment of these Grid systems demand considerable computing resources.

In our approach, instead of building the underlying platform with an exhaustive rich set of features, we create a lightweight core platform, built only with the minimal essential features. The authors of this paper have tried to identify this set of core components that are absolutely necessary for a functional lightweight platform in [12]

were more details of the platform can also be found. The resultant platform is generic and will be used as the foundation in the architectural context presented below, to provide an efficient and highly available yet lightweight infrastructure in unreliable and resource-limited environments.

Figure 1 illustrates the interaction of some of the major components of the platform. More details about some of these components are provided in section 3.

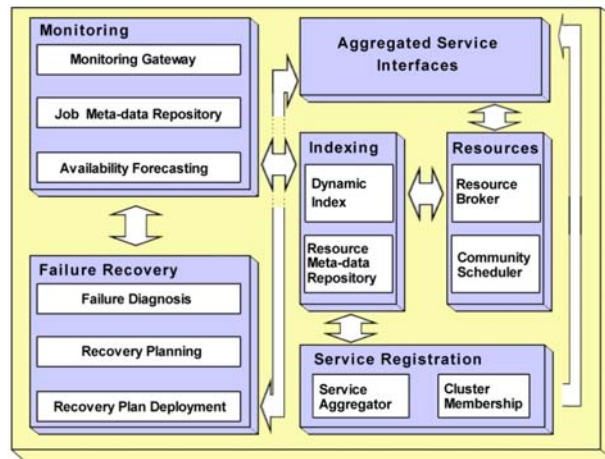


Fig. 1: Lightweight platform components

Architecture

A lightweight Grid platform is the necessary first step that will allow resource limited devices to contribute services and/or resources to the Grid. But when trying to merge an unreliable and very dynamic environment –as in mobile devices making use of wireless communications links, with a relatively reliable and static one –the Grid in its more traditional form, inevitably this will affect the overall reliability, and therefore performance, towards lower levels.

The fragile nature of such devices –due to inherent limitations and unreliable communication links, increases the failure rate. Whenever a failure occurred, the Grid components would have to reschedule and reallocate resources for the active application, possibly migrating data around the Grid thus reducing response and performance times. Considering that in the mobile end of the Grid, the failure rate is increased, this is not something we would like in busy, heavily loaded and complex Grid environments.

To overcome these problems, we are following a different approach: instead of directly presenting each mobile device to the Grid, we are grouping all mobile devices that fall into the same subnet (whether this is defined by physical or logical boundaries) and creating a virtual “cluster” that will be presented to the Grid. This cluster-based design requires a number of proxies that provide the interface point between the Grid and the lightweight cluster of devices. More details about this architectural design can be found on [19].

This architecture will provide a virtualization/abstraction layer in order to present a single interface to all similar resources in the cluster. For example, all storage resources will be presented to the Grid as one big storage pool available through a single interface at the proxy. This “aggregator” service as we call it, will provide access to the aggregated resources in the cluster in a uniform, location and device independent way. Underlying dynamicity and heterogeneity are hidden from the Grid.

This design has a number of significant advantages when built on top of the lightweight Grid platform: provides the foundations for failure recovery and high service availability, hides device and location details from the Grid, masks the dynamicity and unreliability of the environment, provides a uniform interface to the lightweight cluster making job submission and also programming easier, and can provide enhanced functionality and advanced services transparently to the Grid clients.

3. Platform Characteristics

Service Availability and Failure Recovery

For the purpose of failure detection and failure recovery, we are suggesting a monitoring framework based on tested schemes like heartbeats adapted for our centralized cluster based environment, and a well defined set of intelligent agents (or intelligents as mentioned in [14]) installed in the mobile devices that will gather necessary information. This information may include dynamic resource state information like usage statistics, failure ratios, downtime/uptime ratio or static device meta-data like hardware and software information, bandwidth capacity and more. Failure recovery based on the approach of intelligents, makes use of predefined scenarios and trigger lists. If an agent predicts a possible failure (like e.g. severe signal degradation that might lead to an offline status, or low battery levels) data and/or task migration may be required to pre-emptively deal with the failure and save as much computation effort as possible. Human administrator intervention is kept to a minimum.

Collected information can be fed to an analyzing/forecasting component that in cooperation with the cluster’s indexing components will provide recommendations as to what is the best possible resource usage plan. A dynamic index classifies available resources/services according to information supplied by the forecasting facilities e.g. from a set of identical resources, the classification could be based on the downtime-to-uptime ratio of the hosting devices.

Job Submission

Service aggregators, present a uniform interface to the underlying aggregated resources available in the cluster. The Grid only sees a single virtual resource/service. This makes job submission easier and minimizes the workload on the Grid brokers and schedulers that would otherwise be much higher due to a much bigger number of resources in the Grid. The job of scheduling and brokering resources is now delegated to a cluster community scheduling system, deployed at the proxies. The aggregator

services make use of a delegation mechanism that forwards invocations to service and resource providers in the “cluster”. This abstraction layer provides a homogeneous interface to the Grid for all underlying aggregated resources and services independent of the specific service architecture used.

Enhanced Functionality

Aggregator services and the virtualization of resources, allow us to transparently provide enhanced functionality, like collective operations on similar services available in the “cluster”. Examples of such collective operations on available “cluster” services are:

Mirrored execution of a job in many “cluster” hosts to provide increased reliability and the best possible response time for highly prioritized or critical applications.

Mathematical and statistical collective operations on groups of resources or sets of sensors. For example, we could perform merging or reporting operations on independent data sets from a distributed database or on a set of sensors that collect and store formatted data, in order to present a complete and comprehensive data set to the client. This would be done transparently and the client that requested the data set may have no idea of the distributed nature of the database.

Automatic distribution of load, whenever this is feasible and adequate resources are available in the “cluster”.

4. Conclusion and Future Plans

This project is a work in progress that has only recently started. We are developing an implementation prototype using Java technologies to ensure a certain degree of interoperability. So far, we have implemented components for service aggregation, forwarding of invocation requests and indexing of the available services/resource in the “cluster”. We have tested our platform’s aggregation and indexing functions against a wide variety of services and it has been proved to perform reasonably well –even for a very preliminary prototype. Nevertheless, there is still plenty of work to do and our plan for the near future center around dynamic discovery of services, implementation of the monitoring framework, the forecasting components and finally support for mobility and roaming between “clusters”.

The participation of the authors of this paper in the CoreGRID Network of Excellence [16] and the collaboration with partners dealing with many different aspects of Grid technologies, makes us confident that the outcome of this project will be more than just successful and will lay the roadmap for further development in the area of integrating mobile and resource limited devices into the Grid. Our vision of the Grid is to become a truly ubiquitous and transparent virtual computing infrastructure available from all types of end-user devices, whether that is a powerful server, a mid-range desktop or laptop, or a lower-end limited smart phone.

References

1. L. Cheng, A. Wanchoo, I. Marsic, “**Hybrid Cluster Computing with Mobile Objects**”, Proc. of Fourth International Conference on High-Performance Computing in the Asia-Pacific Region, pp. 909-914, 2000.
2. T.Phan, L. Huang, C. Dulan, “**Challenge: Integrating Mobile Wireless Devices into the Computational Grid**”, Proc. of the 8th Annual International Conference on Mobile Computing and Networking, pp 271-278, 2002
3. Chlamtac, J. Redi, “**Mobile Computing: Challenges and Potential**”, Encyclopedia of Computer Science, 4th edition, International Thomson Publishing, 1998
4. B. Chen, C. H. Chang, “**Building Low Power Wireless Grids**”, www.ee.tufts.edu/~brchen/pub/LowPower_WirelessGrids_1201.pdf
5. D. Bruneo, M. Scarpa, A. Zaia, A. Puliafito, “**Communication Paradigms for Mobile Grid Users**”, IEEE/ACM International Symposium on Cluster Computing and the Grid, p. 669, 2003
6. K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke, “**From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution**”, www.ibm.com/developerworks/library/ws-resource/ogsi_to_wsrf_1.0.pdf, 2004
7. The **AKOGRIMO** project: <http://www.akogrimo.org>
8. J. Hwang, P. Aravamudham, “**Proxy-based Middleware Services for Peer-to-Peer Computing in Virtually Clustered Wireless Grid Networks**”, International Conference on Computer, Communication and Control Technologies, 2003
9. D. Chu, M. Humphrey, “**Mobile OGSI.NET: Grid Computing on Mobile Devices**”, www.cs.virginia.edu/~humphrey/papers/MobileOGSI.pdf, 2004
10. J. Thiagalingam, S. Isaiadis, V. Getov, “**Towards Building a Generic Grid Services Platform: A Component-Oriented Approach**”, in V. Getov and T. Kielmann (Eds), “**Component Models and Systems for Grid Applications**”, 39-56, Springer, 2005
11. Grimshaw, A. Ferrari, G. Lindahl, K. Holcomb, “**Metasystems**”, in Communications of the ACM, vol. 41, n. 11, 1998.
12. The Globus Project, www.globus.org
13. Foster, D. Gannon, H. Kishimoto, “**The open grid services architecture**”, GGF-WG Draft on OGSA Specification, 2004
14. S. Corsava, V. Getov, “**Self-Healing Intelligent Infrastructure for computational clusters**”, SHAMAN workshop proceedings, ACM ISC conference, New York, 2002.
15. R. M. Badia, J. Labarta, R. Sirvent, J. M. Pérez, J. M. Cela, R. Grima, “**Programming Grid Applications with GRID Superscalar**”, Journal of Grid Computing, Volume 1, Issue 2, 2003
16. CoreGRID Network of Excellence, www.coregrid.net
17. Foster, C. Kesselman, J. M. Nick, S. Tuecke, “**The Physiology of the Grid**”, <http://www.globus.org/research/papers/ogsa.pdf>, 2002
18. Foster, C. Kesselman, S. Tuecke, “**The Anatomy of the Grid: Enabling scalable Virtual Organizations**”, International Journal Supercomputer Applications, 2001
19. S. Isaiadis, V. Getov, “**Integrating Mobile Devices into the Grid: Design Considerations and Evaluation**”, to appear in the EuroPar proceedings, 2005