

CHRISTOS FILIPPIDIS
YIANNIS COTRONIS
CHRISTOS MARKOU

DESIGN AND IMPLEMENTATION OF THE MOBILE GRID RESOURCE MANAGEMENT SYSTEM

Abstract

We present a design of a Mobile Grid Resource Management System aimed at integrating mobile resources to an existing Grid infrastructure, while creating a platform for interoperability with telecommunications providers. In order to reduce the load on an existing infrastructure, we separate responsibilities and run additional features on mobile resources. Finally we present the Ikaros-EG implementation, IkarosM android application and the mobile-Grid concept. The system used for our implementation was the NCSR “Demokritos” ZEUS computer Cluster, where the Ikaros-EG framework consisting of a meta-data collector and the Ikaros-EG plug-in has been installed. The ZEUS cluster has local and Grid access to data and meta-data from numerous scientific experiments. Our objective was to incorporate the mobile resources using the NCSR Demokritos campus wireless infrastructure in order to run applications concerning accounting, statistics, or further data/meta-data formatting.

Keywords

grid, mobile resources, android, resource management system

1. Introduction

In the last few years we have witnessed an exponential increase in the amount of data produced by large scale collaborative science applications, which now also applies to an enterprise level and everyday user applications. These types of applications require a continuous increase of available resources. Nowadays the necessary resources are mainly provided by HPCs, scientific clusters, DesktopGrids, or by infrastructures constructed by volunteers' computers. We also observe an intense effort for coexistence and cooperation between data networks and telecommunications networks expressed by producing hardware like smart phones and tablet PC's. It is clear that telecom providers are trying to enter the field of data networks by using these devices as gateways.

The proposed architecture embraces mobile resources to Grid, leverages the existing global Grid infrastructure capabilities by integrating these devices in a transparent way while creating a platform for interoperability with telecommunication providers. We separate responsibilities, in order to reduce the load on the existing infrastructure and run additional features on mobile resources.

It is obvious that mobile resources such as a smart phone or a tablet PC are not capable of satisfying the needs of an WLCG [1] job which may demand a device capable of providing 2GB of RAM and 2 GHZ of CPU. Of course, if one considers that nowadays a typical tablet PC can have an 1 GHZ CPU and a typical smart phone a 500 MHz CPU, this may be feasible in the near future. These types of devices introduce two types of constraints, application constraints mainly due to smart-phones' low battery capabilities and architecture constraints due to mobility that these devices introduce.

The paper is organized in the following way. In section 2, we present the capabilities and goals of the traditional batch systems which is a very important set of considerations while building a resource management system. In section 3, we introduce the Ikaros-EG architecture and the Mobile Grid concept while presenting the requirements and complications which led us to adopt this framework. Finally, we present the Ikaros-EG implementation, the IkarosM application, and the system which was used in our implementation.

2. Traditional batch systems

As distributed systems scale to ever larger sizes, they become more and more difficult to control or even to describe. International distributed systems are heterogeneous in every way: they are composed of many types and brands of hardware; they run various operating systems and applications, they are connected by unreliable networks and they change configuration constantly as old components become obsolete and new components are powered on. Most importantly, they have many owners, each with private policies and requirements that control their participation in the community. Flexibility is the key for surviving in such a complicated environment [2].

Traditional batch systems implement a Matchmaking procedure consisting of the following steps. In the first step, agents and the resources advertise their characteristics and requirements. In the second step, a matchmaker scans the known “characteristics and requirements” of resources and creates pairs that satisfy each other’s constraints and preferences. In the third step, the matchmaker informs both parties of the match. The responsibility of the matchmaker then ceases with respect to the match. In the final step, the matched agent and resource establish contact, possibly negotiate further terms, and then cooperate to execute a job [3].

3. Ikaros-EG Architecture

In our effort to embrace mobile resources, such as smart-phones and tablet PCs, to the existing Grid infrastructure we realized that it would not be efficient to adopt a traditional batch system architecture for our resource management system. All existing systems such as Condor, SGE, and BOINC do not support android platforms. Moreover, even if this happens in the future, these systems will not be suitable for mobile integration, because the matchmaking procedure of such a system introduces a complexity level which a mobile device cannot handle. In addition, we believe that the most critical consideration in the attempt to incorporate telecommunication networks and mobile devices to an existing Grid in a transparent way is to build a bridge between the different infrastructures, i.e. Grids/data and telecommunication networks and not try to extend the Grid architecture to the telecom networks. We have built the Ikaros-EG resource management system which acts as a plug-in for an existing Grid infrastructure and as a content provider for the mobile devices, a very familiar concept for the telecommunication providers and networks. Content providers are one of the primary building agents of Android applications, providing content to applications. They encapsulate data and provide it to applications [4].

3.1. Ikaros-EG plug-in

The Ikaros-EG plug-in is designed as an Apache Dynamic Shared Object (DSO) [5] which supports HTTP and GridFTP. We built Ikaros-EG as an Apache DSO using the HTTP protocol as the basic mechanism for data transfer for the following reasons:

- HTTP is the basis for most Web and Grid Services.
- HTTP has an excellent interaction with Firewalls, Network Address Translation, and Application Proxies.
- HTTP can, in practice, be as fast as other TCP-based protocols like GridFTP[6].

By using the Ikaros-EG plugin, we intend to use our infrastructure only for the core services which must be implemented by the resource management system. All the additional features, such as accounting, statistics, data searching utilities or further data/meta-data formatting in order to connect the system with other utilities or applications, run on smart phones and tablet PCs. Ikaros-EG, creates jobs which can run on these devices. By building Ikaros-EG we don’t want to substitute the existing

resource management system of a Grid infrastructure. We only use it as a resource management system for the mobile devices and the telecommunication networks.

3.2. Ikaros-EG framework

It is quite common to visualize Grid architecture as a series of successive layers, namely the network, resources, middleware, and application layers (Fig. 1). In such a view, the Ikaros-EG framework is part of the middleware and application layers. Using a data or a meta-data collector the Ikaros-EG plugin creates suitable android jobs for mobile devices implementing the additional features for our architecture.

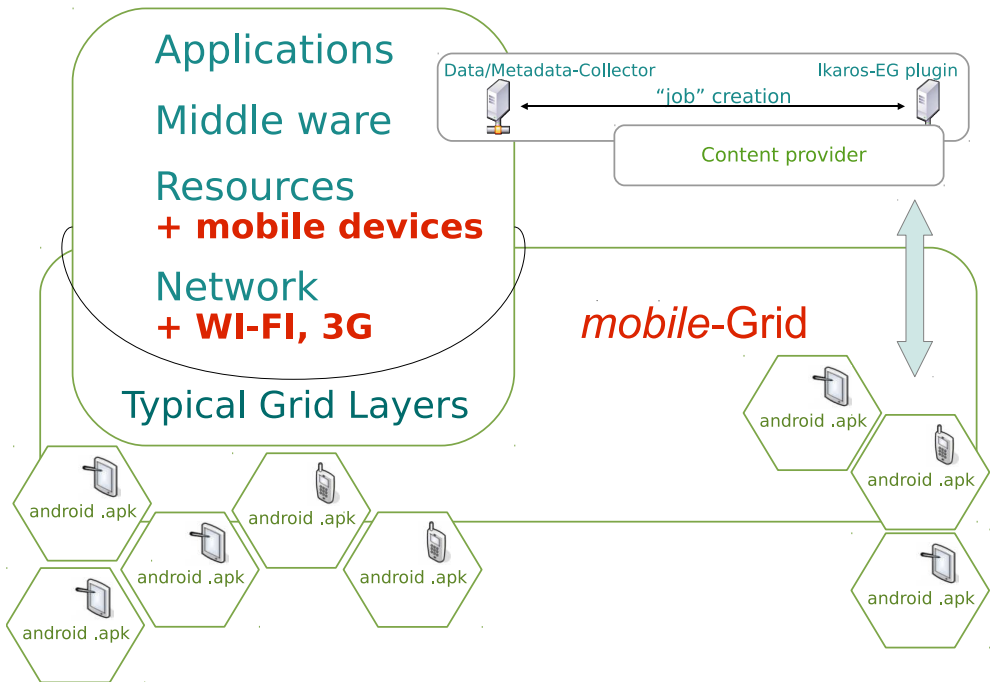


Figure 1. mobile Grid concept

In this way the plug-in acts as a content provider for the mobile devices. These mobile devices run an android application which communicates with the Ikaros-EG framework from which they are capable to download available “jobs” for execution and return the results to the system. The job execution process is not affected if the device changes its configuration, its network setup, or whether it operates on a private or public network. In this way we introduce the “mobile-Grid” concept. It is obvious that we add mobile devices to the resources layer and new interfaces, such as WI-FI and 3G, at the network layer.

With this architecture we do not need the matchmaking procedure which demands steps such as the advertisement of resources. In our case, we built two different

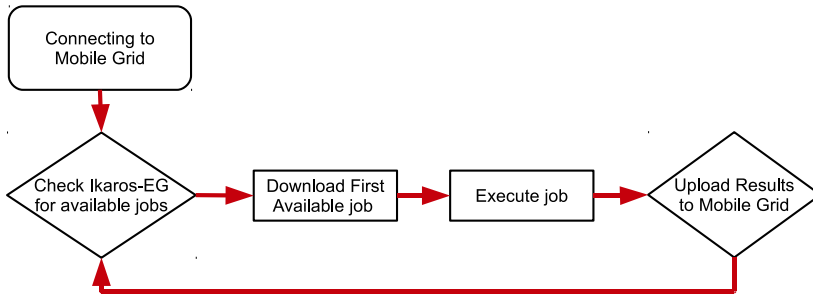


Figure 2. IkarosM android application flow chart

types of jobs, two different types of android applications. In Figure 2 we present the IkarosM android application flow chart. The IkarosM.apk acts as a state machine which checks Ikaros-EG for available jobs. It's obvious that the resources management is actually done by the android application and that is why we do not need a matchmaking procedure at the Ikaros-EG. In this way job allocation and resources reliability is not an important factor. This concept seems more efficient for these type of devices due to the fact that the hardware characteristics do not vary much, between smart-phones or between tablet PCs. For these type of devices it is more important to build jobs which will not consume much of the smartphones' battery, something which will irritate the device owner, rather than build a matchmaking procedure. We have to build light "jobs" for these devices and it seems that "jobs" concerning accounting and statistics for example can be suitable. Although a single job of this type does not require much CPU time, this type of job can be thousands in a Grid environment. A huge amount of requests can lead to overloading our infrastructure. Additionally, if, as we perceive mobility as the freedom to change, in an ad-hoc manner, the configuration of a HPC or a campus IT Infrastructure participating in a Grid, in order to cover new demands, without affecting the job execution process then Ikaros-EG architecture can offer much even to traditional computing infrastructures.

4. Ikaros-EG implementation

In this section, we present the Ikaros-EG implementation, the IkarosM android application and the mobile-Grid concept. The system used for our implementation was the ZEUS computer Cluster of the Institute of Nuclear Physics of NCSR "Demokritos", where we installed the Ikaros-EG framework consisting of a meta-data collector and the Ikaros-EG plug-in. The National Center for Scientific Research Demokritos runs High-Performance scientific applications due to its participation in experiments such as CMS [7] at CERN and KM3NeT [8]. The ZEUS cluster has local and Grid access to data and meta-data from these experiments. Our objective was to incorporate the

mobile resources using the NCSR Demokritos campus wireless infrastructure. In this way we managed to reduce the load to our existing computing infrastructure while adding new features to our system.

The android application which we present is a raw meta-data to XML parser. This type of applications is suitable for smart-phones because of the low CPU and network connections demands which results to low battery consumption. Low battery consumption is our goal in order to keep the mobile device owner satisfied. On the other hand the requests for using this type of applications can be thousands resulting to a huge load and finally overloading of the infrastructure. By distributing these requests to mobile resources we manage to reduce the load to our main infrastructure. The ZEUS cluster acts as a content provider for the IkarosM android application runs on the mobile devices.



Figure 3. IkarosM-Connecting to mobile Grid



Figure 4. IkarosM-Getting Job



Figure 5. IkarosM-raw data



Figure 6. IkarosM-upload results

In Figures 3–6 we present snapshots of the application running on a smart-phone, showing various distinct steps, namely that the device is connecting to the mobile Grid, getting the job, processing it and finally uploading the results to the ZEUS Cluster. In Fig. 5 the IkarosM application running on the android device downloads raw meta-data and the parser runs at the android application converts them to an

XML DOM tree. The android application finally uploads the XML DOM tree (Fig. 7) to the ZEUS Cluster.

```

-<FileMetadata>
  <FileSize value="125628107"/>
  <FileParts value="3"/>
-<FilepartsInfo>
  <PartInfo Filelocation="http://pvfs2-io-0-0:8888/azov_amo_2006315_250m_geo9.tif" Partbuffer="41876035"
  Partseek="0"/>
  <PartInfo Filelocation="http://pvfs2-io-0-1:8888/azov_amo_2006315_250m_geo9.tif" Partbuffer="41876035"
  Partseek="41876035"/>
  <PartInfo Filelocation="http://pvfs2-io-0-3:8888/azov_amo_2006315_250m_geo9.tif" Partbuffer="41876037"
  Partseek="83752070"/>
</FilepartsInfo>
</FileMetadata>

```

Figure 7. XML DOM tree

5. Conclusion

We have presented the design of a Mobile Grid Resource Management System which aims to incorporate mobile resources to an existing Grid infrastructure while creating a platform for interoperability with the telecommunications providers. We separate responsibilities in order to reduce the load on the existing infrastructure and run additional features on mobile resources. Finally we present the Ikaros-EG implementation, IkarosM android application and the mobile-Grid concept. Ikaros-EG can be categorized as an architecture between a traditional Grid infrastructure, such as a gLite [9] and an infrastructure which implements the volunteer computing concept, such as BOINC [10]. With the mobile-Grid concept, we tried to attract new resources to the Grid in order to leverage the infrastructure capabilities while trying to fully benefit by the characteristics of these resources. At an enterprise level and in everyday life the mobile Grid concept could incorporate telecom providers to a Grid infrastructure, in a transparent way. These providers have a huge base of users with smart phones and tablet PCs and can provide paid services to enterprises and scientific experiments while their users will have the opportunity to gain from this process with reductions in their bills.

Acknowledgements

We acknowledge the partial financial support of the Special Account for Reserach Grants of the University of Athens.

References

- [1] <http://lcg.web.cern.ch/lcg/>
- [2] Thain D., Tannenbaum T., Livny M.: *Distributed Computing in Practice: The Condor Experience* Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pp. 323-356, February-April, 2005.

- [3] Livny M., Raman R.: *High-throughput resource management*. [in:] I. Foster, C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.
- [4] <http://developer.android.com/reference/android/content/ContentProvider.html>.
- [5] <http://httpd.apache.org/docs/2.0/dso.html>.
- [6] <http://www.gridsite.org/talks/>
- [7] <http://cms.web.cern.ch/>
- [8] <http://www.km3net.org/>
- [9] Cristina A. *et al.*: *Design and Implementation of the gLite CREAM Job Management Service*, Future Generation Computer Systems, December 24, 2009.
- [10] Anderson D. P., Fedak G.: *The Computational and Storage Potential of Volunteer Computing*, IEEE/ACM International Symposium on Cluster Computing and the Grid, Singapore, May 16–19, 2006

Affiliations

Christos Filippidis

University of Athens, Ilissia Attiki Greece; NCSR Demokritos, Agia Paraskevi Attiki Greece, filippidis@inp.demokritos.gr

Yiannis Cotronis

University of Athens, Ilissia Attiki Greece, cotronis@di.uoa.gr

Christos Markou

NCSR Demokritos, Agia Paraskevi Attiki Greece, cmarkou@inp.demokritos.gr

Received: 8.12.2011

Revised: 27.01.2012

Accepted: 30.01.2012