# Grid Operating System

Arshad Ali[1], Richard McClatchey[2], Ashiq Anjum[1, 2,] Irfan Habib[1]
1 NUST Institute of Information Technology, Pakistan
2 University of West England, Bristol, UK
Email: ashiq.anjum@niit.edu.pk

**Abstract**
Grid computing has made huge advances during the last decade, most notably the deployment of tera-scale grids. However not all users of computers have been beneficiaries of the advances in grid computing. One of the most neglected segments has been the common user, who still predominantly uses operating systems which do not support resource sharing and the scale of collaboration possible in grid environments. Current grid middleware and associated software is extremely complex to set up and use for the common user. As a result of this, desktop PCs are still not a major part of existing grid infrastructures. This paper introduces a concept for a grid operating system targeted at the common user. Its aim is to bring grid computing to the desktop, and bring the desktop PC to the grid. The paper also discusses related research challenges such as grid enabling interactive applications and associated scheduler and resource brokers, developing a grid enabled file system, peer-to-peer discovery etc. These issues must be tackled before a Grid operating system can be successfully developed and its benefits realized.

## 1 Introduction

Grid computing has become increasingly pervasive in a short span of time. However most of the emphasis in grid computing effort has been directed towards science and engineering and deploying grids in large scale computing facilities. As a result grid middleware and its associated software have become far too complex for the common user to install on his desktop computer, besides not being able to run common user applications.

Many Grid computing projects such as SETI@HOME [1], have used desktop PCs to good effect, however little effort has been made to introduce standard interfaces to the desktop OS, to make the PC a major player in the existing grid infrastructure. Contemporary projects related to Grid operating systems [2,3] have focused on providing operating system functions by emulating them in the application layer, however these systems have failed to simplify Grid computing to the level of a common user and grid enable his desktop.

A desktop user should be able to do with his desktop whatever scientists have been able to do with grid-linked clusters. A user should be able to access information on a data farm the world over, run applications which can be transparently migrated to powerful systems and collaborate with a group of users on a particular project using shared resources.

## 2 Potential benefits

Assume a scenario where a user runs an interactive program on his PC. In contemporary operating systems such as MS Windows and GNU/Linux all user interaction, I/O and eventual processing is handled at the user's PC. However we believe that more productive use can be made of neighbouring as well as world wide computing resources.

In a grid enabled operating system the user would feel just like as if he is using a conventional operating system, however at the kernel level all aspects of

- Resource brokering and grid-wide scheduling, discovery of resources;
- mapping of "parts" of local processes to global processes , and process status tracking;
- security and confidentiality of data;

would be handled in a peer-to-peer fashion with other machines, transparently form the user. Hence leading to a huge increase in the response time of the user application and decreasing the complexity of grid computing software dramatically.

On the flipside of this coin, Grid computing itself can benefit from the introduction of the desktop to the infrastructure. It should be possible for any organization including governments to create *ad-hoc* grids, and have desktop users subscribe to them and share their resources, according to a sharing policy defined by the user either on a non-profit or profit basis. This has great opportunities for organizations which do not have the resources to purchase and setup clusters or supercomputers; with grid enabled desktops they can utilize existing commodity PCs.

It is important to separate the dual objectives that are to be achieved: bringing Grid computing to the desktop means that the benefits of Grid computing such as resource sharing, and enhanced collaboration can be introduced into the domain of desktop computing. The other aim of bringing the desktop to the grid entails introduction standard interfaces such as Open Grid Services Architecture (OGSA)[4] to the grid operating system to make it part of the existing grid infrastructure.

## 3 Research Challenges for building a desktop user oriented grid OS

The greater part of the last decade research in grid computing was primarily directed towards science and engineering applications [5], which have

significantly different requirements than desktop computers, hence there are significant challenges in each major component of an operating system

- How can we define a mechanism where "parts" of a program can be outsourced to other machines to grid enable interactive applications, and how to track their status?
- How will the nodes of a desktop grid be linked and identify each other? And what kind of networking paradigm would be suitable?
- What kind of protocols must be defined to enable such collaboration between users and how to grid enable the file system?
- How can we implement security and enable control of resources?
- How can we implement standard interfaces to the desktop to make it interoperate with the existing grid infrastructure?

These are many questions to answer, amongst others. In the following sections we will have a more detailed look at some major questions.

### 3.1 Grid Enabling Interactive Applications

One possible solution of grid enabling interactive applications is the creation of a new threading methodology: threads with introspection and metadata capabilities. In such a system, desktop application could be developed in a multithreaded fashion and the user interaction could be handled by one thread, and the processing parts could be handled by another thread.

The operating system scheduler will know with the metadata what kind of thread it is about to schedule and communicate with peer systems, to know if this thread can be moved to some other machine for more high speed processing. The system would shield user which are executing other users threads from the privileged instructions of the operating system, and any system call would be mirrored back to the system where the thread originated.

### 3.2 Resource management

Resource brokers are a vital component in a grid system. Most existing grid infrastructure use job specific or history based resource brokers, which are unsuitable for a general purpose environments, thus a generic resource broker is needed.

### 3.3 Networking paradigm

Contemporary grids mostly implement the client server paradigm; this is not workable for the grid enabled desktops, as there would be no centralization in such a system. The most popular paradigm which has proved useful for a loose knit community of users has been the peer-to-peer paradigm. However in order to support existing grid infrastructures, the grid enabled operating system must also support the client-server paradigm. Any peer-to-peer grid operating system must also inculcate a discovery service to transparently discover neighboring nodes.

### 3.4 Security

There are also major questions regarding security which need to be answered by the system: How can we discourage any single malicious user from intently abusing the system? How can we ensure confidentiality of the data of the peers, integrity of the data, and ensure maximum availability?

### 3.5 File system

The peers of the grid must be able to access each others' data transparently; all issues of data transfer, fault tolerance and scalability should be handled by the file system. Building such a file system is a considerable research challenge.

### 3.6 Standard Interfaces

The system should be compatible with the current infrastructure. Standard interfaces would be provided by the operating system so that users can easily interconnect with the existing grids.

## 4 Conclusion

As discussed in this paper, most research in grid computing has been targeted to some segments of computer users. No conscious effort has been made to bring the benefits of Grid computing to the desktop, and bring the desktop to the existing grid infrastructure. Thus some significant challenges must be overcome to bring grid computing benefits of resource sharing, and enabling unprecedented levels of collaboration between common users. This paper has identified some of the most important issues to be resolved in providing a Grid OS. Making the desktop part of the grid infrastructure, would provide existing grids with millions of new nodes, and implementing standard interfaces required to make the desktop ready for the grid, will reduce the cost of creating and maintaining a grid, which has potentially huge benefits for companies, and governments around the world.

## 5 References:


[1] SETI@HOME site : www.seti.org

[2] Klaus Krauter etc, *Architecture for a Grid Operating System*, International Workshop on Grid Computing, 2000

[3] Pradeep Padala etc, *GridOS: Operating System Services for Grid Architectures*, International Conference on High Performance Computing 2003

[4] I. Foster et al, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, OGSI WG, Global Grid Forum, June 2002.

[5] F Berman, G Fox & A. Hey. *Grid Computing: making the Global Infrastructure a Reality*. Wiley Publishers, 2003.