XtreemOS: Towards a Grid Operating System with Virtual Organisation Support

Ian Johnson, Amit Lakhani**, Brian Matthews**, Erica Yang

e-Science Centre, STFC Rutherford Appleton Laboratory, Chilton, Didcot, OX11 0QX, UK

Christine Morin

IRISA / INRIA Rennes research center - Bretagne Atlantique, France

Abstract

XtreemOS is a European project which aims to develop an open source Grid operating system based on the existing Linux operating system. A key goal of XtreemOS is the native support for Virtual Organisations. In this poster, we present the overall motivations, aims and components of XtreemOS and give an initial view of the support for Virtual Organisations.

1. Introduction

Grid middleware has become an integral part of e-Science. Projects like EGEE[1], are dependant on Grid middleware to support collaborations and share resources. Facilities providing high-performance computing resources, such as the Barcelona Supercomputing Center[2], use the Grid to enable secure distributed access to users of its facilities. Services such as the UK National Grid Service[3] provide access to Grid infrastructure to communities of scientists in academic institutions.

Traditionally, Grid middleware such as Globus and its variants, is developed on top of existing operating systems such as Linux. Little has been done to extend the underlying operating systems to support Grid computing, for example, by embedding some important basic services directly into the operating system kernel. Consequently a number of European researchers got together to explore the concept at a workshop in March 2005 [1].

XtreemOS[4] is a European project which has the objective to design, implement, evaluate and distribute an open source Grid operating system (named XtreemOS) which supports Grid applications, and capable of running on a wide range of underlying platforms, from clusters to mobiles [2]. The goal is to provide an abstract interface to its underlying local physical resources, as a traditional operating system does for a single computer.

This approach can be seen to have some advantages over conventional Grid Middleware. Application programmers spend a good deal of time managing the services which are provided by the middleware toolkit, which may have differing programming interfaces and lack of a unifying model. By making the Grid support native to the operating system, a common interface can be provided to simplify the task of the application developer on the Grid, and also by removing layers of abstraction, leading to higher dependability of services.

The approach being investigated is to base XtreemOS on the existing Linux OS. A set of system services, extending those found in the traditional Linux, will provide users with all the Grid capabilities associated with current Grid middleware, but fully integrated into the OS.

2. General Architecture of XtreemOS

The overall schematic architecture of the components of XtreemOS is given in Figure 1. The XtreemOS architecture is divided logically into two layers.

The XtreemOS Foundation layer, XtreemOS-F, provides a modified Linux Kernel in the kernel embedding native Virtual Organisation support. This will be provided in three major variants: a version aimed at PCs and workstations; a version aimed at providing a Single-System Image for cluster computing, based on the existing Kerrighed system [3]; and a version which can be deployed on small mobile devices.

[1] http://public.eu-egee.org/
[2] http://www.bsc.es/
[3] http://www.grid-support.ac.uk/
[4] https://www.xtreemos.org/

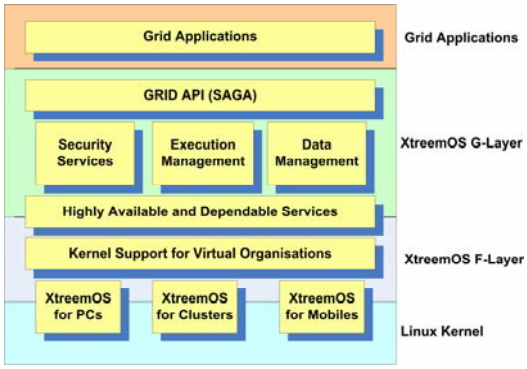

Figure 1: XtreemOS Schematic Architecture

The XtreemOS Grid support layer, XtreemOS-G will provide Grid OS distributed services to securely manage computation and data resources. The main services supported in the G-Layer are execution management, a Grid file store, and application and Virtual Organisation level security. In order to deploy the XtreemOS-G layer over a number of participating nodes, an infrastructure of highly available services will manage these nodes, by providing for example support for publish/subscribe services, node virtualization, and node directory services.

XtreemOS will provide a common API to grid applications; this will be based on the emerging Simple API for Grid Applications (SAGA) standard currently under development within the OGF [4].

3. Virtual Organisations in XtreemOS

A key feature of the XtreemOS design is native support for Virtual Organisations (VOs).

Virtual Organisations are now considered a key aspect of Grids which allow them to scale [5]. A VO can be seen as a temporary or permanent coalition of geographically dispersed entities (individuals, groups, organizational units or entire organizations) that pool resources, capabilities and information to achieve common objectives. The resources can be physical equipment such as computing or other facilities, or other capabilities such as knowledge, information or data.

The exact definition of VO differs from project to project. Some approaches concentrate on the legal or contractual arrangements between the participating entities. Others take a task-oriented approach, emphasising the workflow required to achieve a goal. VOs can range from long-lived collaborations with many users, typically found in large-scale scientific applications, to short-lived, dynamic ventures set up to achieve one task between a small number of participants, more typically of commercial scenarios. A general purpose Grid Operating system should take a flexible approach to accommodate as wide a range of applications as possible; indeed the use cases in XtreemOS reflect this diversity.

Thus the approach in XtreemOS is to define a minimal definition of the features of VOs, and to provide a toolbox which can be configured to the needs of the application. Key components of a VO are:

- an administrator of the VO; a set of users in different domains;
- a set of resources in different domains;
- a set of roles which users and resources can play in the VO;
- a set of policies on resource availability and access control;
- an expiry time of the VO.

VO Goals or Workflows are not modelled explicitly, though XtreemOS tools should allow these to be supported at the application level. This will typically require enforcement of policies, event notification of the completion of processes, and monitoring of exceptional events, such as jobs still executing at VO expiration. Similarly, we would not expect kernel support of contractual arrangements, but require monitoring and enforcement of policies which can be derived from contracts.

3.1 Challenges of Supporting Virtual Organisations in XtreemOS

XtreemOS aims to provide native support for the management of VOs in a secure and scalable way, without compromising on flexibility and performance. Several key challenges are identified from both the requirement analysis and investigation of most state-of-art grid VO solutions.

Interoperability with diverse VO frameworks and security models: different VO management frameworks and security models have been developed so far and new ones keep on emerging. The diversity of their implementations is embodied in their adoption of different user identities (e.g. X.509 end user certificates, Shibboleth handles, etc.), different message sequences (e.g. push, pull and agent protocol described in RFC 2904), different places to convey security attributes (e.g. proxy certificates or SAML tokens) and different

policy models (e.g. role-based access control). XtreemOS must be able to interoperate with, rather than replace these existing solutions and even traditional local security mechanisms (e.g. Kerberos). It is a challenge that the operating system level abstraction of VOs in XtreemOS allows for integration of various existing VO structures.

Flexibility of policy languages: XtreemOS takes the view that both scientific and enterprise business applications are equivalently supported. Users from these two representative application domains have different views of policies in a VO, in terms of subjects (users), objects (resources), access rights, Service Level Agreement (SLA) and Quality of Service constraints. Therefore VO policies in XtreemOS have to be expressive and flexible enough to accommodate various levels of resource access rules.

Scalability of management of dynamic VOs: In order to also support large numbers of users in a dynamic environment (dynamicity of resources and of users) while still providing accurate isolation of these users, solutions such as rather static files containing user information must be avoided. For example, when VOs are dynamically changed, it is impractical for the VO manager to update grid-map files on all resources, due to the heavy admin burden caused as well as the difficulty to maintain data consistency.

Strong isolation, access control and auditing: Some applications request for strong isolation of user applications on the grid: hiding user identities, protecting files and processes, strict division of performance load, and so on. These requirements are typical for most of the industrial applications. In some environments this ability to generate strong isolated execution environment could even be used to isolate individual processes on a single resource. Implementing such requirements is difficult without operating system support. Furthermore, a secure grid system must provide strict access control from the service level down to the system object level (files, sockets, etc). In all cases, it must be possible to monitor and log operating system service usage as well as system object accesses. The audit log must contain references to user credentials (security ticket) and be securely provided to the resource owner as well as the VO manager.

3.2 VO Management (VOM)

We use the concept, VO Management (VOM), to cover all the services that are needed to manage the entities involved in a VO and ensure a consistent and coherent exploitation of the resources, Virtual Organization Management in XtreemOS capabilities, and information inside the VO under the governance of the VO policies. A VO policy is defined as an authorization statement that describes what activities a subject (e.g. an entity in a VO) is allowed to perform on an object (e.g. resources) with certain constrains (e.g. time, location), if there is any.

There are several stages of VO lifecycle: VO identification, VO formation, VO operation, VO evolution, and VO dissolution. VOM plays a different role in different stages of this lifecycle.

During the identification stage, VOM is mainly responsible for user management (e.g. registration, attribute management) and VO policy specification (e.g. constraints on resource usages). During the formation stage, VOM involves in the processes of resource matching, negotiation and establishment of Service-Level Agreements (SLAs) by applying VO policies. The operation stage leverages the information made available during the previous stages. In this stage, VOM coordinates logging, accounting, auditing operations on nodes and ensures the availability of such information, if needed. For jobs that require interactive sessions, VOM also provides authorized users with facilities (e.g. credentials) to access the sessions of runtime applications. The evolution stage takes place when the VO is altered during its lifespan, for example, by a change in the participating entities or in their conditions of use. During the last stage, VOM ensures the deletion of non-persistent information (e.g. temporary files and accounts) and the reclamation of credentials.

4. Conclusions

We have described some of the requirements and challenges in providing VO support within a Grid operating system being addressed within XtreemOS. These have been the outcome of the first stage of the XtreemOS project which has concentrated on developing requirements and initial architectural design of the virtual organisation support, at both the kernel level and within the Grid support services of XtreemOS.

This work is ongoing. The component parts are under design and implementation, and may be available early for independent release and testing in the wider community. The initial integrated prototype of XtreemOS which will provide the basic instantiation of this

architecture is planned for a first public release in mid 2008. The initial Virtual Organisation support will be based on a VOMS system [6], modified to use Linux Pluggable Authentication Modules (PAMs). This will then be tested on a variety of use cases and further refined.

Further extensions to the basic VO support are planned. These would include: mechanisms for federated authentication; trials of expressive policy languages; the role of virtualisation to support highly secure commercial VOs; the integration of trust domains. Further, we regard it an essential for a practical system that there should be some assurance provided that the systems does meet recognised security criteria. Work is ongoing to derive a systematic analysis of threats to the XtreemOS system, with a view to validating the integrity of XtreemOS.

Acknowledgements

XtreemOS (IST-033576) is a Project co-funded by the European Commission within the Sixth Framework Programme, running from June 2006 to May 2010. We would like to thank our partners in the XtreemOS consortium, and our colleagues at the National Grid Service for their help and advice.

References

[1] CoreGrid Workshop on Network Centric Operating Systems, Brussels 16-17 march 2005. http://coregrid.cetic.be/NCOSworkshop

[2] C. Morin. XtreemOS: a Grid Operating System Making your Computer Ready for Participating in Virtual Organizations, *Proc. of ISORC 2007, 10th IEEE International Symposium on Object / component / service-oriented Real-time distributed Computing, May 2007*.

[3] Kerrighed Main Page http://www.kerrighed.org/wiki/index.php/Main_Page

[4] Simple API for Grid Apps Research Group (SAGA-RG) http://forge.gridforum.org/projects/saga-rg/

[5] I. Foster, C. Kesselman, and S Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.

[6] DataGrid VOMS (release v0.7.1). http://edg-wp2.web.cern.ch/edgwp2/security/voms/