Improvement of Grid Service deployment using virtualisation technologies.

Research Proposal for MPhil/PhD enrolment

Jack Wootton

27th October 2006

1 Introduction

1.1 Overview of Grid

Grid computing involves the coordination of multiple distributed processes such that they function as one system. Computational Grids are used to achieve this. Grid computing operates through the use of remote machines accepting executable code, which they process. Grid computing defines far more than the organisation and architecture of computational networks. How Grid is used, the policies under which it may be used, access control and the even distribution of computational resources are all within the remit of Grid Computing. The multifaceted nature of Grid ensures that it remains a hotbed of research. As the technologies underpinning Grid computing mature, so Grid becomes more usable and the applications of Grid broaden; an example being the relatively new commercial interest in Grid computing [5].

1.2 Grid Architecture

Grid architectures fall into one of two categories:

1. Resource oriented.
2. Service oriented.

Service Oriented Architecture (SOA) is a paradigm for making computational resources available to consumers. Consumers can be realised in many ways: other services, humans and programs can all be service consumers. The adoption of SOA by Grid has resulted in the Open Grid Services Architecture (OGSA). OGSA is a layered architecture, with each layer presenting some functionality with the aim that higher level services and applications use lower level layers to become part of a grid and therefore share resources [6].

SOA consist of three primary actors:

1. Service provider – the business logic of the service being offered.
2. Service registry – a method to locate services based on criteria provided by the potential service consumer.
3. Service consumer – user of the service being offered.

Web services realise SOA through *stateless* services, whereas Grid computing realises SOA through *stateful* services implemented using the Web Service Resource Framework (WS-RF) [2]. The WS-RF is supplied as part of the Globus Toolkit 4 (GT4). The latest stable release from Globus is version 4.0.3 [3]. A grid service at its most atomic is simply be a web services implementing the WS-RF.

1.3 The problem

The research proposed in this document presents a solution to a problem that has two causes. The causes being:

C1. Demand for new Grid services.
C2. High demand for existing Grid services.

Cause C1 can lead to physical grid resources becoming underutilised since new services, that consumers of Grid (primarily Virtual Organisations) require, are not currently dynamically deployed. Cause C2 can result Grid services becoming unavailable to VOs since the service has too much demand placed upon it. The problem can be defined as:

"The underutilization of physical Grid resources due to either, the static nature in which services are currently deployed, or, the unavailability of existing Grid services due to over utilisation of those services."

A potential solution is to allow for the dynamic deployment of Grid services as they are required. There are two candidate implementations of this solution.

1. The cloning of an operating system and its environment such that it may be deployed to an available physical resource.

2. The detection and selection of operating system environment and requirements such that a Grid service environment can be created and subsequently used to host Grid services.

2.0 Virtualisation

Virtualisation of resources can be achieved using a number of technologies. These technologies are now outlined:

2.1 Virtual workspaces

Grid computing involves the sharing of heterogeneous Grid resources between VOs, each with potentially conflicting resource requirements. Such shared use often results in under-utilisation of Grid resources caused by a mismatch between resources offered and the application requirements [4] of the shared users'. However sharing is not the only cause of resource under utilisation. [10] Draw attention to the potential incompatibility of a cluster's installed libraries across VOs as another difficulty with current Grid use. Virtual Workspaces (VW) are an attempt to address the following three problems with Grid clusters:

- Lack of performance isolation.
- Little control over resource sharing.
- Fine grained usage difficult to enforce.

VWs act as an abstraction layer from physical Grid resources and allow execution environments to be dynamically created by VOs such that the execution environment will suit, exactly, the specific needs of each VO. A Grid client may then define a VW driven by their own specific requirements [7]. Importantly distribution of Grid resources between multiple VOs is maintained.

The Globus Alliance [10] developed the Workspace Management Service (WMS) which provides an interface to common VW functionality. Since it is a service developed for use with Grid computing, naturally it complies with the GT4 implementation of the Web Service Resource Framework (WSRF) model. Through the WMS VOs can create, manipulate and destroy workspaces.

Virtual Workspaces result in Grid 'jobs' (executables to be run on Grid resources) being mapped to a workspace rather than a resource. Providing the job maps successfully to a workspace and the workspace resource description can be honored by a resource, the job will run.

VWs can be realised in two ways, firstly through Dynamic Accounts, secondly using virtualisation technologies.

2.2 Dynamic (Unix) Accounts

Implementing the WMS through Dynamic Accounts (DA) works either by adding user accounts on a Unix system (using the Unix `useradd` command), or requesting an account from a pool of accounts. Independent of the method chosen, the objective is to map a Grid identity onto a local Unix account. Once the account is in use, the Unix operating system provides:

- Control of resource allocation through the use of Unix commands: `setrlimit`, `quota` and `chroot`.
- Control over account lifetime.
- Auditing.
- Isolation through user groups and user accounts.

The 'backend' of the WMS implementation uses Unix commands, accounts, groups and tools to create or load a workspace configuration, as well as customise it once it is loaded.

2.3 Virtualisation

Dynamic Accounts provide a relatively coarse grained implementation of VW and do little to provide a custom execution environment for the Grid client. Virtual Machines (VM) can be used to implement VWs through the WMS by installing Grid resources with a Virtual Machine Monitor (VMM). Images containing representations of RAM, disk and other resource attributes can be loaded by the VMM. The custom (Grid) execution environment can then be used. VMs provide excellent isolation and offer fine grained control over resource customisation, as well as providing the Grid client with the perception of using a single isolated machine. One of the most appealing attributes of VMs is the ability to serialize the state of the VM, such that it may be reloaded on a different physical resource and resume processing. [11].

Significant barriers remain in the adoption of virtualisation technologies in Grid, namely staging of the VM image and any data and resources required during job execution. However, assuming staging of resources has been completed, use of VMs cause only small performance degradation (approximately 5%) [7].

2.4 Virtual Workspace deployment

It has been described how the WMS can be implemented using either DAs or Virtualisation. But how does a Grid client describe its resource requirements such that the execution environment may be created to realise the resource requirements? A number of methods have been experimented [8], [10], [4], [1], [9].

An accepted [7], [9] method is to create an XML schema that describes the workspace, how it should be deployed and the quota of resources required. Workspace description schema are called *workspace metadata schema*.

A VM image that fulfills the workspace described by the workspace metadata schema (above) is retrieved from a pool of suitable VM images, and staged onto the resource.

Alternatively an existing VM image is cloned; the new image is then staged onto the physical resource.

2.5 Virtual clusters

So far, only atomic workspaces implemented using VMs or DAs mapped onto a single Grid resource has been discussed. [4] propose extensions to the workspace metadata schema and changes to the WMS, such that virtualisation of entire clusters may be achieved. Virtual clusters work by means of:

- Collections of atomic workspaces.
- A method of describing collections of atomic workspaces.
- A method of managing collections of atomic workspaces.

The workspace metadata schema is extended to include aggregations of atomic workspaces. Workspaces now become *aggregate workspaces*. An aggregate workspace contains sets of workspaces; each set contains homogeneous workspace configurations. It is through the combination of sets within an aggregate workspace, that a heterogeneous cluster may be described using a single workspace metadata schema. Importantly the ability of aggregate workspaces to define heterogeneous clusters allows both service and worker nodes to be described.

In order to manage aggregate workspaces, the WMS is extended to handle local IP address assignment to each node. In addition, the service is extended to utilise a database which records the availability and state of physical resources that may be used to deploy a workspace. The database may then be queried so that suitable physical resources are selected for the deployment of the workspace.

3.0 Objectives

The following research issues will addressed in the development of an automated Gird service deployment system.

1) Dependency detection: Develop a method for detecting dependencies and requirements of a Grid service, examples of requirements and dependencies include libraries, other services or simple additional files. All of the aforementioned may be required for the correct functioning of a Grid service and as such should be included in any Grid service descriptor.

2) Virtualisation: Virtualisation involves copying or cloning images containing representations of RAM, disk and other resource attributes. There are two approaches. Firstly an entire operating system and environment image can be cloned and run on an execute machine with a suitable virtual machine manager installed. Secondly, components of the virtual machine can be built up to form a

highly customised and suitable execution environment for a grid job. The difference being: rather than cloning an entire operating system and environment, using dependency detection, only the required environment attributes, settings and resources are cloned and run on an execute machine with a virtual machine manager.

3) Automatic deployment tool: Dependency detection and virtualisation can be used in combination to build a highly suitable grid execution environment, however the environment is useless unless a grid service can be deployed to it. An automatic deployment tool would be developed to deploy a grid service to a virtual execution environment. It can be assumed that the execution environment can be organised to suit the requirements of the grid service through dependency detection.

4.0 References

[1] A.Matsunaga et al. 2005. *On the Use of Virtualisation and Service Technologies to Enable Grid-Computing*.

[9] Globus Aliance Web Service Resource Framework Reference - http://www.globus.org/wsrf/

[3] Globus Toolkit 4 download - http://www.globus.org/toolkit/downloads/4.0.3/

[4] I.Foster et al. 2006. *Virtual Clusters for Grid Communities*.

[5] I.Foster, D.Iamnitchi, 2003. On *Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing*.

[6] J.Joseph et al. 2004. *Evolution of grid computing architecture and grid adoption models*.

[7] K.Keahey et al, 2005. Virtual *Workspaces in the Grid*.

[8] K.Keahey et al. 2003. *Dynamic Creation and Management of Runtime Environments in the Grid*.

[9] K.Keahey et al. 2005. *Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid*.

[10] X.Zhang et al. 2005, *Virtual Cluster Workspaces for Grid Applications*.

[11] The Xen virtual machine monitor - http://www.cl.cam.ac.uk/research/srg/netos/xen/