

Grid Computing

Gary McHugh, *Student, GMIT*,

Abstract—Grid Computing can be defined as a large number of geographically dispersed computers connected to solve a complex problem. Servers or personal computers run independent tasks and are loosely linked by the internet or a low speed network. Grid computing allows a user to gain access to computing power that is equivalent to a mainframe or a super computer. TeraGrid is an example of a successful business that uses the grid architecture, it was mainly used in academia. Grid computing became a large area for research particularly with Resource Management, researchers focused on trying to make the allocation of resources to heterogenous systems more efficient. The Global Grid Forum created standards for Grid computing, this spurred the development of The Open Grid Services Architecture (OSGA) followed by the Globus Toolkit 3.0. These tools helped to define standard protocols for the grid architecture. Finally, we summarise the challenges faced by Grid Computing and its future as a distributed architecture.



1 INTRODUCTION

GRID Computing is a collection of many geographically dispersed computers that are connected to solve a complex problem. It is a special kind of distributed system where each node is performing a different task/process. The computers tend to be geographically dispersed not just across countries but also across continents. These computers are interlinked with each other by the internet or low speed networks. It is common that the computers are both heterogeneous (running different operating systems) and geographically dispersed. [3]

The term Grid Computing originated in the mid 1990's to describe a distributed computing infrastructure. [1] The goal of Grid Computing was to allow an easy way for consumers to access high computational power. [2] Through virtualization, grid gathers resources from dispersed locations to provide users with a transparent high performance environment that has high storage, computational power and instrument capabilities [4].

Ian Foster first put forward the concept of Grid Computing, he and others hypothesized that through standardization of the protocols used to request computational power when using grid, they could replicate a computer grid that is comparable in structure and efficiency to an electric power grid. This concept was then developed in many ways to create the modern-day Grid Computing infrastructure. [1]

2 DEFINING GRID COMPUTING

Ian Foster provided a three-point checklist in order to define what was categorized as being grid, if the project does not have these three points then it is not grid. [1] The three points in the checklist are as follows:

1) Coordinates resources that are not subject to centralized control: This ensures that we are not using a local management system, it does this by integrating and coordinating resources and users that are geographically in different control domains. For example, different companies, different departments of the same company or the user's computer and the central computer. It's also

addresses problems like security, policy, payment and other conflicts that occur.

2) Using standard open, general-purpose protocols and interfaces: A Grid is composed of versatile protocols and interfaces that address crucial points such as: authentication, authorization, resource delivery and resource access. This is to ensure that we are not referring to an application specific system.

3) To deliver nontrivial qualities of service: This means that the resources are used in a synchronized fashion to accomplish different qualities of service such as: response time, availability, security, throughput and allocation of multiple resources to meet user demands. This leads to the utility of the system to be greater than the sum of its parts. [5], [1]

An example of a large system that allows institutes to use grid computing is TeraGrid. This gives a user access to high computing power but also data and software on demand, I will discuss this further in the Business Model section of this paper. Examples of Grid Computing being applied in the real world are: Distributed Aircraft Engine Diagnostics and NEESgrid Earthquake Engineering Collaboratory. The Distributed Aircraft Maintenance Environment (DAME) project uses grid computing to address the critical issue of computer-based fault diagnosis in Plane Engines. The data can come from a range of data sources making the grid computing architecture optimal for this type of problem. [7] DAME is specifically working on diagnosing faults in Rolls Royce aircraft engines by using sensor data recorded during trans-Atlantic flights. [2]

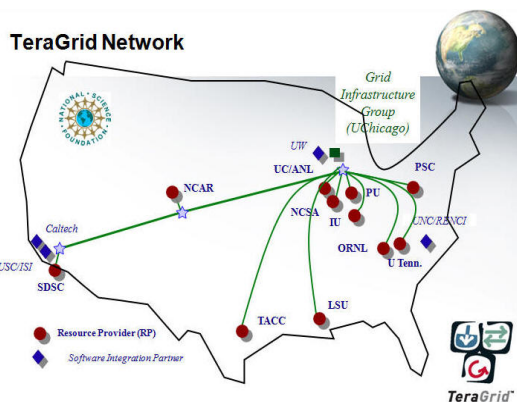
The second example is the U.S Network for Earthquake Engineering Simulation (NEES). This is an ambitious project that is attempting to "enable remote access to, and collaborative use of, the specialized equipment used to study the behaviour of structures, such as bridge columns when subjected to the forces of an earthquake." [2] The NEESgrid uses grid technologies to gain experimental access to join physical experiments with

numeric simulations and to archive, discover and analyse data. [6] [8]

3 BUSINESS MODEL

Business Models in grid computing are popular in academia as well as government labs, they are mainly used for academic research/projects and government related research. TeraGrid was a grid computing business model in which users were designated an amount of CPU hours in which they could use the TeraGrid's resources. Users had to write a complex proposal in order to increase the computational power they got to use. TeraGrid had over twelve grid sites which were hosted throughout America. TeraGrid had access to over 2 petaflops of computing capability and more than 60 petabytes storage (2 quadrillion bytes). When an institute joined the TeraGrid with a set of resources they were also allowing the community to access their resources, they became part of the TeraGrid. They also recognised that they gained access to over a dozen grid sites. This model was applied around the world and was successful in different countries. This gave the institutes encouragement to join more grids to gain additional CPU hours. [1]

Grid computing projects that had important impacts in scientific fields included: NAREGI and Open Science Grid, they both achieved admirable results that made exceptional contributions to scientific research as well as in China with the Chinese Academy of Science Computing with the National Defence University. [4] When using a batch-scheduled compute model with a local resource manager like Condor, users submit batch jobs to request resources to use. For example, if a user wanted to run an application for one hour and required 80 processors to execute the task, and then wanted to output the resulting data to a server. The user would request these resources from a central computer, the request would then be added to local resource managers wait queue. When the resources became available, they would be allocated to the application for the requested duration of the task. Because of expensive scheduling algorithms, long queue times and data staging, some people argue that grids shouldn't support interactive applications.



However, efforts are being made in the research community to lower wait times and to allow applications with many

short running tasks to run more efficiently on grids. [13], [1] Task scheduling has become one of the major areas of research as it directly influences the efficiency of grid applications. Task scheduling is responsible for the allocation of resources; it does this by using scheduling algorithms and various policies. Researches are constantly attempting to create new and optimise existing scheduling algorithms to improve applications efficiency. This area is of interest to business's who provide grid services, as it will enable them to maximise the use of their grid sites. This would allow them to increase the number of users without having to increase the number of grid sites, in return turning a higher profit. [9]

4 RESOURCE MANAGEMENT

Resource Management and Scheduling Systems are responsible for managing resources and the execution of the application. This is dependent on the resources requested by the user and the availability of those resources due to other applications using them. The Grid Computing environment is extremely complicated due to many widely distributed heterogeneous resources which dynamically change. [4] This introduces some challenging issues such as: site autonomy, heterogeneous substrate, policy extensibility, resource allocation, online control and resource trading which Resource Management attempts to solve. [10] The majority of the early development of resource management for The Grid concentrated on attempting to overcome the heterogeneity of the computers. Most of the concepts for Grid Computing existed from other computing environments, these concepts were applied to the Grid architecture. Unfortunately, Resource Management could not be applied to the Grid Architecture due to this heterogeneity problem.

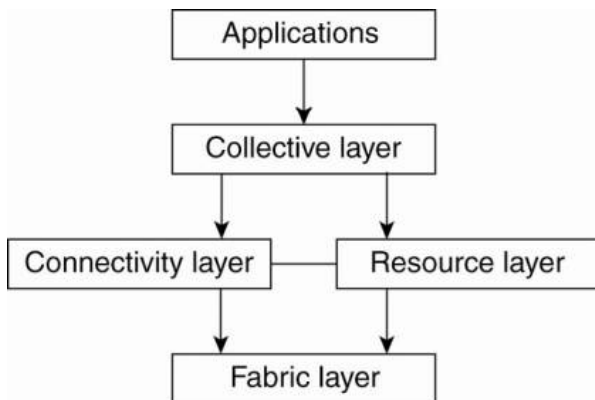
Resource Managers like batch schedulers and work flow engines worked under the assumption that they had complete control over resources and could therefore implement the policies and mechanisms that it required, this concept did not work with heterogeneous systems. Standard resource management protocols and standard mechanisms for expressing resource and task requirements were defined to help overcome the heterogeneity problem. The way in which different organisations operated was an even bigger issue. The different organisations operated their resources under different policies, also the objectives of the user and the resource provider may be conflicting. Grid Applications also frequently required simultaneous allocation of resources at a given time. This had not been an issue in previous Resource Management algorithms. This meant that grid computing needed a structure in which resources were coordinated across heterogeneous systems. The ongoing research in Grid Computing focuses on resource management, particularly in comprehending and administering different policies from both the view of the user and the service provider. The goal is to incorporate "end-to-end resource management in spite of the fact that the resources are independently owned and administered." [2]

5 ARCHITECTURE

Grid Computing implements the service-orientated architecture, whereby it provides the necessary hardware and software services as well as the infrastructure for a secure and uniform environment. This grants access to heterogeneous resources and enables the formation of virtual organisation. [10] A virtual organisation is an organisations whose members are geographically dispersed. To the end user however the virtual organisation appears to be a single organisation in a physical location rather than many dispersed members.

As interest in The Grid grew throughout the 1990's the need for standards became an issue of importance. The Global Grid Forum was a community based standards organisation and they began to work on standards for The Grid. This seen the development of The Open Grid Services Architecture (OSGA) in 2002 which is an example of a service-orientated architecture. [11] This was followed by the OSGA based Globus Toolkit 3.0 which was released in 2003. This toolkit built on the previous concepts in Toolkit 2.0 and aligns Grid Computing with industry initiatives in a service-orientated architecture. [2] To support the formation of virtual organisations, Grid defines a set of protocols, middleware, toolkits and services that are built on top of these protocols. The main issues for a Grid architecture is security as there are a lot of resources coming from separate administrative domains, and interoperability of these domains. Both issues have global and local resource usage policies, contrasting software configurations and operating systems, and they vary in capacity and availability. [1]

Grids implement protocols and services at five layers: Fabric Layer, Connectivity Layer, Resource Layer, Collective Layer and Application Layer.



The Fabric Layer: At this layer grids grant access to resource types like compute, storage and network resources. Most Grids use pre-existing fabric components such as Condor for local resource management, General Architecture for Advanced Reservation (GARA), which is a general-purpose component for the fabric layer or resource management resources like Falkon.

The Connectivity Layer: This layer is responsible

for protocols that allow communication as well as authentication of users. It is also responsible for secure network transactions. A protocol called Grid Security Infrastructure (GSI) is used for all Grid transactions. [1]

The Resource Layer: Contains protocols for the publication, secure initiation, monitoring, negotiation accounting and payment of sharing operations on individual resources. Grid Resource Access and Management (GRAM) is a commonly used protocol for the allocation of resources for monitoring and control of those resources. The ability to run the same application on different systems is dependent on the resource protocols used. The Globus Toolkit discussed above is frequently used for connectivity and source of resource protocols. [12]

The Collective Layer: This layer captures interactions between several resources. Monitoring and Discovery Service is an example of a service that is available that enables the monitoring and discover of virtual organisation resources. Condor is an example of a co-allocating, scheduling and brokering service.

The Application Layer: This is the topmost layer; Applications rely on all the other layers below them to run on the Grid. This layer encompasses what the application has built on and operates in the virtual organisation environment. An example of a grid workflow system is TeraGrid which you should be familiar with from the Business Model section of this paper.

6 CHALLENGES

One of the biggest challenges that Grid Computing faces is Parallel Computing. This has been one of the key areas of research in Grid Computing, different algorithms have been designed through research in an effort to make the simultaneous execution of processes as efficient as possible. Grid also encounters the challenge of transferring the work load to resources over a long distance. This is becoming a larger problem with the amount of data increasing, this means the I/O and network performance must also increase.

Another challenge faced by Grid is that in a multi-grid environment Grid needs to integrate heterogeneous resources and coordinate communication among grid systems. [4] The Global Grid Forum has been crucial in the development of The Grid with their contributions of the Open Grid Services Architecture (OSGA) which extends the Globus Toolkit. This has eliminated the challenge of defining a standard InterGrid protocols as the OSGA has become commonly used tool in the Grid which defines standards and protocols used in Grid. [5]

7 CONCLUSION

In this paper, we first discussed what Grid Computing was at a basic level and where the concept of Grid Computing came from. We then defined Grid Computing at a higher level using Ian Foster's three-point checklist and looked

at some real-world projects that have used The Grid Architecture. We examined the business model in The Grid using TeraGrid as an example, we looked at how a user requests resources when using the batch scheduling model while discussing the popular research area of Task Scheduling. We then went on to talk about the challenge of resource management and examined the architecture of The Grid. Finally, we looked at some of the challenges that Grid Computing faces.

What is the future of grid computing? As we have seen, Grid has evolved over time based on the power grid, I believe that we will continue to see this parallel evolution with the electric power grid and grid computing. They will move towards a mix of micro production as well as large utilities. I think there will also be some small-scale producers that will exist alongside the large-scale producers. We will continue to see this push for efficiency in grid computing through new research and newly developed algorithms. I believe that there will still be a place for grid computing even with the current push for cloud computing and its ongoing evolution.

REFERENCES

- [1] *Cloud Computing and Grid Computing 360-Degree Compared* Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu.
- [2] *The Grid in a nut shell* Ian Foster and Carl Kesselman
- [3] *Grid Computing* Wikipedia
- [4] *Grid or Cloud? Survey on scientific computing infrastructure* Bing Yu, Jing Tian, Shilong Ma, Shengwei Yi, Dan Yu
- [5] *What is the Grid? A Three Point Checklist* Ian Foster
- [6] *NEESgrid A Distributed Virtual Laboratory for Advanced Earthquake Experimentation and Simulation*
- [7] *Predictive maintenance: Distributed aircraft engine diagnostics. TheGrid: Blueprint for a New Computing Infrastructure (Second Edition)*. Jim Austin, Tom Jackson, Martyn Fletcher, Mark Jessop, Peter Cowley, and Peter Lobner, Morgan Kaufmann, 2004
- [8] *Distributed telepresence: The NEES grid earthquake engineering collaboration. TheGrid: Blueprint for a New Computing Infrastructure (Second Edition)*. C. Kesselman, T. Prudhomme, and I. Foster, Morgan Kaufmann, 2004
- [9] *Sort-Mid tasks scheduling algorithm in grid computing* Naglaa M Reda, A Tawfik, Mohammed A Marzok, Soheir M Khamis
- [10] *Market-Oriented Grid and Utility Computing* Published by John Wiley & Sons, Inc.,
- [11] *Grid Services for Distributed System Integration* Ian Foster, Carl Kesselman, Jeffrey M. Nick and Steven Tuecke
- [12] *Grid Computing: Making the Global Infrastructure a Reality* Fran Berman, Geoffrey Fox and Anthony J. G. Hey
- [13] *Falcon: a Fast and Light-weight task execution framework* Ioan Raicu, Yong Zhao, Catalin Dumitrescu, Ian Foster and Mike Wilde