

Grid Computing: the Current State and Future Trends (in general and from the University of Canterbury's perspective)

Andrew Roxburgh Krzysztof Pawlikowski

Department of Computer Science and Software Engineering

and Donald C. McNickle

Department of Management

University of Canterbury
Christchurch, New Zealand

TR-CSSE 01/04

1 Introduction

The term 'Grid Computing' is relatively new and means a lot of different things to a lot of different people[19]. It has been used as a buzzword for any new technology to do with computing, especially computer networking, and therefore it has been over-hyped as the solution to just about every computing problem. One of the goals of this paper is to give a clear definition of what Grid Computing is and why it is required.

Grid Computing, or Network Computing, is intended to provide computational power that is accessible in the same way that electricity is available from the electricity grid - you simply plug into it and do not need to worry about where the power is coming from or how it got there. The idea of Grid Computing is the same - if more computing power is required, spare cycles on other computers are used. This means that super-computer type power is accessible without the huge costs of super-computing, and that CPU cycles that would otherwise be wasted are put to good use. In fact, one of the major researchers into Grid Computing, Ian Foster from the University of Chicago says "grids are above all a mechanism for sharing resources", [13]. This means primarily sharing CPU time but also other things such as data files. Although this description sounds simple there are a number of problems with creating Grid systems - how do you access computers with different operating systems, how do you find those computers to access and how do you make sure that you can trust others to run code on your machine? In fact, how do you encourage people to let others run code on their machines in the first place? These questions, and many others, need to be answered for Grid Computing to succeed and they are also discussed in this paper.

Grid Computing is no longer just a concept to be discussed but is something that is actually used every day. There are many Grids around the world, and many researchers investigating how to do Grid Computing better. These current Grids and the some of the current Grid research topics are also discussed in this report.

There is also significant potential for Grid Computing to be used at the University of Canterbury. There are several projects which are very well suited to Grid Computing and it is likely that others would emerge were a Grid system available. The potential for Grid Computing and some of the tools that could be used for this are discussed below as well. The layout of this paper is as follows: Section 2 discusses why Grid Computing is needed at all. Section 3 discusses what makes up a Grid system, and Section 4 discusses some current Grids and Grid technologies. Section 5 discusses some of the current issues that need to be addressed in Grid Computing, Section 6

discusses the possibility of Grid Computing at the University of Canterbury, and finally section 7 concludes.

2 Why Do We Need Grid Computing?

Although the amount of computing power available to both researchers and businesses is growing at an amazing rate, and has been growing quickly for some time, the demand for computing power is never satisfied. New projects in business and in the sciences require unprecedented amounts of computing power that, even given Moore's Law, will not be fulfilled in the near future. The rate of increase in network bandwidth is increasing at a rate faster than that of processor speed which means that the way to make best use of computing power is to network many computers together in an efficient fashion[17]. Grid Computing is currently seen as the best way to do this.

The New York Times recently published an article which argued that "All Science Is Computer Science". This claim was made because every traditional science - Physics, Chemistry, Mathematics, Biology, Astronomy, and many others - is relying more and more on computers and computational power. Although new insights are still needed to generate new research in these fields, the limiting factor in many of the experiments is computational power. Grid Computing is therefore seen as the computing technology enabling the advancement of all sciences.

2.1 Advantages Over Traditional Supercomputers

Based on the above arguments, it could be said that all that is needed are bigger supercomputers. However, Grids have several advantages over traditional supercomputers. When purchasing a supercomputer it is hard to know how much power to purchase. If too much is purchased it will cost more but the supercomputer will be under-utilised. If not enough supercomputing power is purchased the advantages of supercomputing will not be fully realised. If the supercomputer will be used at its peak rate only for short periods of time, it will be even harder to decide what to buy because either the programs will run too slowly when they are running or you will pay more when most of the time the extra power is not being used. Grid Computing is better than a supercomputer because its size can be changed dynamically - computers can be added or removed once the Grid has been deployed - and because dynamic sharing is available with other organisations. Whilst with a supercomputer you can allow others to use it, with a Grid they can use it while you are not but then when you do want to access it you can do so straightaway. This is because you could set up your computing resources so that your Virtual Organisations (see section 3.2) have higher priority than others.

3 What Makes Up A Grid?

Before the problems and solutions of Grid Computing can be discussed, it is important to have a clear definition of what constitutes a Grid. The following sections attempt to do this, and introduce some of the important Grid terminology.

3.1 Components of a Grid

Grid Computing can be defined as *the seamless provision of access to possibly remote, possibly heterogeneous, possibly untrusting, possibly dynamic computing resources*. Analysed piece by piece, this definition means that Grid Computing provides seamless access to:

1. Possibly Remote Computing Resources
Means that local resources, which are on the same LAN, and remote resources, which are geographically distant, can be accessed in exactly the same way on the Grid.

2. Possibly Heterogeneous Computing Resources

Some computers on the Grid can run different Operating Systems on different types of machines. Accessing them via the Grid should be possible without making any special allowances for this.

3. Possibly Untrusting Computing Resources

Means that the owner of a computing resource on the Grid might not know or trust other users but should still be confident that they cannot access any non-shared data and cannot make malicious system calls on their computer. The Grid should handle this security checking without any specific instruction from the user or from the sharer.

4. Possibly Dynamic Computing Resources

One of the major selling points of Grid Computing is that it makes use of otherwise wasted CPU cycles. The problem with this is that the availability of computers to the Grid changes rapidly as computers become busy and then idle as their owner's usage varies. The Grid system should ensure that this dynamism is hidden from users so that they do not have to program explicitly to take account of this.

Seamless provision means that Grid users can access such seemingly un-accessible resources easily without having to worry about all these complications. Altogether, this definition leads to four main things that any Grid system must provide seamlessly in order to be considered a Grid, [15]

1. Authentication
2. Authorization
3. Resource Access
4. Resource Discovery

3.1.1 Authentication

Authentication means that each user has an identity which can be trusted as genuine. This is necessary because some resources may be authorised only to certain users, or certain classes of users (see Section 3.1.2). Authentication of a user should happen only once when they start using a Grid - they should not have to sign on separately to each of the many machines that their computation may use.

3.1.2 Authorisation

Authorisation means that each resource - be it the spare computing power on a computer of an organisation or a set of astronomical data - will have a set of users and groups that can access it. The Grid needs to first authenticate that the users are who they say they are and then ensure that they are allowed to access the resources that they are requesting. Having groups authorised to access certain resources leads to the idea of *Virtual Organisations*, which are discussed further in Section 3.2.

3.1.3 Resource Access

Resource Access means that remote resources can be accessible to Grid users. These resources could mean anything from CPU time to disk storage, to visualisation tools and data sets. As discussed, not everyone should be able to access all resources but the Grid must provide a way to access those that are allowed. This means that some sort of virtual machine is required so that machines with different operating systems, etc. can be accessed in a uniform way.

3.1.4 Resource Discovery

Being allowed to access thousands of different CPUs is useless without being able to find out where they are. Resource Discovery means that users can find remote resources that they can use. This process should be automated by the Grid so that a user's task can automatically be run remotely without them having to go through the process of finding CPUs that they can use. The automation of resource discovery is complicated hugely by the dynamic nature of Grid resources - what is available at one instant of time may no longer be available a while later. Added to this complication is the desire to avoid a single central point where all data is stored because the failure of it would bring the whole system down and one single point of control is not a scalable solution - if the Grid becomes really large this central point would be badly overloaded.

3.2 Virtual Organisations

The idea of a Virtual Organisation (VO) is that on, say, a university campus-wide Grid, members of the Physics and Biology departments could be working on a project together so they could form a Virtual Organisation for that project where they could all access the data for that project and each other's computing resources. However, those who are not members of the research group would not be members of the VO so would not be able to access the resources. Members of the Computer Science department - who would not be part of the other VO - may be working on a different project however could have separate projects running with separate access rights for a different set of resources. Note that different projects within the same departments could also have separate Virtual Organisations so keep some of their data separate but allow projects from both VOs to use the compute resources.

4 Current Grids and Grid Products

There are a number of tools available to help create Computational Grids, both free, open-source ones and commercial products. There is also a standards body which seeks to put forward 'recommendations' about how best to do Grid Computing. This section gives an overview of these, and details about several of the many Grids in existence today.

4.1 Tools and Standards

4.1.1 Globus

The Globus Toolkit[25] designed by the Globus Alliance contains a set of free software tools - services, APIs and protocols - to facilitate constructions of Grids. It is the most widely used toolkit for building of Grids and is frequently referred to as the *de facto* standard; see e.g. [18], [6]. It includes tools for, among other things, security, resource management and communication. The Globus Alliance also researches various issues related to Grid Computing, especially issues relating to the infrastructure of Grids. Almost every Grid which has its details published was constructed using the Globus Toolkit.

4.1.2 The Global Grid Forum

The Global Grid Forum (GGF) performs a similar role to the development of Grids as the W3C does toward the development of the World Wide Web, [26]. It is a conglomerate of interested parties including universities, research institutes and industry. It is not an official body so it does not put forward standards but just 'best practices' for Grid developers. It is important because it provides a forum for new ideas to be discussed by all interested parties. There are strong links between the GGF and The Globus Alliance - ideas put forward by the GGF are often implemented by Globus.

4.1.3 Condor and Condor-G

Condor is a software tool for distributing computationally intensive jobs over Grids. It works by using spare CPU cycles on other computers. It provides a way of doing resource discovery using ‘ClassAds’ which matches job requests to unused resources. From the Condor product Condor-G has been created [9]. Condor-G is an enhanced version of Condor which can be used to make Grids. It uses Globus tools to provide “security, resource discovery, and resource access in *multi-domain* environments” with Condor’s “management of computation and harnessing of resources within a *single* administrative domain.” There has also been work on making separate Condor pools “self-organising, fault-tolerant, scalable, and locality-aware” which has proved to be a successful way for automatic management of larger groups of Condor pools, [1].

4.2 Some current Grids in development and deployment

There are many Grids currently in use and in production; in this section we examine several of them in detail. These are not claimed to give a representative sample of all current Grids, but are only to give insight into a few of them. The huge Euro Grid project and the United States National Fusion Collaboratory are discussed.

4.2.1 European Data Grid

The European Data Grid [24] is a European Union funded project which aims to create a huge Grid system for computation and data-sharing. It is aimed at projects in high energy physics, led by CERN, biology and medical image processing, and astronomy. It is being developed using and extending the Globus Toolkit. In building the Grid new tools and systems have been developed in many areas useful for the extension of Grid Computing. For example, a method of enabling secure access to databases in Grid environments has been developed [18]. New techniques for searching for patterns in genomic data using the European Data Grid have also been developed [10].

4.2.2 The National Fusion Collaboratory

The National Fusion Collaboratory[7] project exists to help research magnetic fusion. Magnetic fusion experiments operate on pulses of plasmas which are produced approximately every 15 minutes. The data generated from each measurement must be analysed within the 15 minutes so that changes can be made to the set up in time for the next pulse [16]. This time limit means that it would be very useful for the researchers to be able to analyse the data quickly so that more time can be spent reconfiguring the experimental set up. For this reason, the National Fusion Collaboratory constructed a Computational Grid. This project was also built using the Globus Toolkit and the main research focus is on ‘advanced reservations of multiple resources’ - this means that resources such as computational cycles can be reserved in advance if it is known that they will be required sometime in the future.

4.3 Commercial Grid Products

There are several Grid products currently listed on various websites; see for example [31] and [29]. They claim to easily enable Grid Computing within organisations but it is hard to tell how much they actually do because they do not publish refereed papers - most of the information available about them is probably marketing hype and not a verifiable fact. When the NorduGrid was being constructed in Scandanavia they chose to develop their own Grid system because nothing existing was suitable, [11]. This shows that at this stage at least commercial products were not of a high enough standard for real use.

5 Current Issues In Grid Computing

Grid Computing is still very much in its development stage and there are a number of issues that must be addressed or resolved before it can be considered as a stable technology. Some of these issues are discussed below.

5.1 The Grid versus Many Grids

A distinction must be made between the idea of a single, worldwide, ubiquitous grid and the idea of many separate grids located in businesses and on university campuses. The original intention of Grid Computing was that it would follow the same architecture as the electricity grid. This means that whenever and wherever you needed compute power you would simply “plug in” to *The Grid* and the processing would be done. There would be no need to know where the computing was being done - just as there is no need for me to know where the power that is lighting this room is coming from - only that it was being done. In the same way that I don’t need to know whether the electricity lighting this room is coming from a hydro-electric power plant in Fiordland or a wind turbine in Wellington, I wouldn’t care if my complicated simulation were being run on a spare machine next door or on an idle server somewhere on the other side of the world. In fact, *The Grid* could be viewed as a Grid of Grids, in much the same way as the Internet is a network of networks. Although work is still being done toward creating a single Grid, it is already the case that there are many disparate grids worldwide that are all completely isolated from each other. Having many separate Grids makes issues like authentication and Virtual Organisations much simpler, which is one of the reasons that *The Grid* has not emerged. It also eliminates the need for some sort of global billing system, which is discussed further in Section 5.3.

Some progress toward creating a single worldwide grid has been made, however. The PlanetLab project [30] is a distributed testbed for testing new networking protocols, planetary scale file sharing, and many other ideas which can benefit from having a huge distributed testbed. It involves hundreds of computers at different locations around the world, mostly within academic institutions, on which researchers at the institutions can run experiments. It is not an initiative aimed at creating a global Computational Grid but it does provide some of the things that a Grid must provide, such as authentication and authorisation. It currently has 361 nodes (as at 20 February 2004)[30] connected to it so it is far short of being a worldwide Grid but it is certainly an important step toward it, both in the new research initiatives that it has allowed and in demonstrating that world-wide distributed computing projects are feasible. It has been expected that PlanetLab will have over 1000 nodes distributed over the world by the end of 2004. Its only node in New Zealand is under care of the Network Research Group in the Department of Computer Science and Software Engineering at the University of Canterbury in Christchurch. So far the only Australian node of PlanetLabs is located at the University of Technology in Sydney.

5.2 No-one wants to share

One of the biggest problems facing Grid Computing is not a technological one but a social one. Even when the technology exists for Grid Computing to work easily and flawlessly, people are still required to donate their spare CPU cycles or Grid Computing will not work at all. Although one of the major points of Grid Computing is that only spare cycles will be used, it still goes against human nature to allow others to access their computers and run programs on them. A fear of viruses is no doubt a valid threat as what has been viewed as a secure system in the past has been shown not to be so, so much work must go into developing a security infrastructure that can be completely trusted.

In the SETI@home project [3], and others like it, work by volunteers around the world allowing their computers to be used for scientific research shows that some people at least are willing to share for no direct benefit to themselves but it is unlikely that everyone would allow this. Within single businesses or university departments it is likely that it could be official policy that every

computer must be part of the organisation's Grid, but this would probably not work for *The Grid* without some sort of global billing system.

5.3 Grid Economics

Before all the separate grids can be connected into one 'supergrid' some sort of billing system must be established that is accepted and trusted by everyone. It is unlikely for a worldwide Grid to take off and make use of almost all spare CPU time without some incentive for people to make their computers available. However, in order for a world-wide billing system to work, there will need to be some way of accurately keeping track of the CPU time used, the CPU time provided by each user and a way of transferring payment between users. The development of such a system in a way that is scalable and trusted by everyone is necessary before a global Grid can become the reality.

The development of such a system could lead to some sort of global bidding system for compute power - which would fluctuate like the stock market. The value of CPU time would vary over time according to supply and demand. Daytime hours in the North America during the working week would probably have the highest demand so would cost more, but could make use of the servers in Europe and Asia that are not handling their peak capacity. The analogy of the Computer Grid with the electricity grid can be expanded further - just like it is possible to feed power back into the electricity grid - it will be possible to feed computing power back into the Computer Grid.

In order for a stock-market like Grid billing system to succeed, several obstacles must be overcome. Local resources must be able to be used first, otherwise a company could incur costs from using The Grid that they wouldn't have otherwise. This includes stopping non-local users from using the local resources in order to run local Grid applications. In order for a stock-market system to work it must also be made sure that businesses or universities do not incur charges that are more than the gain they would have made. If running an application on The Grid saves several seconds but costs \$100 then, it is probably not worth it. The ISP charges as well as the Grid charges must be taken into account when calculating how much it will cost to run on The Grid, which further complicates the issue. These problems mean that although *The Grid* certainly can come into effect sometime, it is likely that in the next few years at least the development of Grid Computing will focus mainly on the simpler task of creating separate Grids at separate organisations.

5.4 Performance Forecasting

One of the problems with scheduling resources on a Grid is that it is hard to know how long a resource will be available for or how good its performance will be if it is used. Researchers have implemented a tool known as *EveryWare* which contains, amongst other things, a performance forecasting mechanism [21]. With accurate forecasting, scheduling becomes simpler because it is known that a given resource will react fast to requests or process data quickly. Without accurate performance forecasting a scheduler could schedule a remote set of CPUs to try and speed up processing but actually make it slower because those CPUs do not perform as well as expected. There is still work to be done in this area, however, as the performance forecasting needs to be incorporated into scheduler algorithms and the accuracy of performance forecasting can no doubt be improved.

5.5 The No-Defined Problems Problem

A vital step in solving problems is identifying what they actually are. With any new technology it is hard to know what the key problems to be solved for that technology to work are - there are no forums for putting problems forward to be solved and no systematic attempts by various researchers to solve them [2]. To encourage the formulation of specific problems and solutions, The authors of [2] propose several problems that they see as holding back the progress of Grid Computing and challenge other researchers not only to solve those problems but to supply more.

Although Grid Computing has reached a state when a common vocabulary has been formed of Grid Computing terms and various components of any Grid Computing system have been spoken of, there is still inconsistency of what the different terms mean and when they are used. When basic terms related with Grid Computing and components of Grid systems are agreed upon, research into Grid Computing will be in a much better shape.

5.6 Security

As mentioned, one of the reasons that people may not want to make their computer available on a Grid is that they do not trust other users to run code on their machines. Within small scale Grids this is not too much of a problem as Virtual Organisations at least partially eliminate the fear of malicious attacks. This is because in a Virtual Organisation you can authorise only those from within a certain trusted organisation to be able to access your computer. However, there could potentially be problems with the authorisation systems and it is possible that someone from within the organisation could act in a malicious way. With larger scale Grids it will be impossible to know and trust everyone who can access a single computer so the Grid infrastructure will have to provide guarantees of security in some way.

The Java Sandbox Security Model [14] already provides an environment in which untrusted users are restricted from making certain system calls which are not considered safe, and from accessing memory addresses outside of a certain range. Any Grid system will have to provide a similar mechanism, so that users will be happy to let others access their computer.

5.7 Supercomputing Power For Everyone?

In the past, supercomputing power has been available only to very few people - certain people in research institutions and some businesses. If *The Grid* is ever created, though, supercomputing power will be available to anyone who wishes to access it, although probably at a fairly large cost. This means that, amongst other things, anyone can do huge password searches or can try and crack public/private keys. With the creation of *The Grid*, these issues will have to be addressed either by somehow restricting users from being able to do such searches or by using even larger keys and passwords. As [5] shows, what is considered to be an unbreakable key one year can be inadequate a few years later, and with the advent of *The Grid*, this situation will be re-enforced further. There are no doubt many other social issues that will arise when everyone can have access to supercomputing power, and they will have to be addressed as well.

5.8 The Need Not To Centralise

Any Grid system must have some knowledge of what resources are available in order to provide Resource Access and Resource Discovery. The logical way to do this would be to have a central repository listing all resources currently available and who is allowed to access them. The problem with this centralised solution is that it is not at all scalable and means that the entire Grid system is subject to a single point of failure. For these reasons, another way of providing Resource Discovery is required.

If there were a central repository containing details on all Grid resources for a large Grid, the speed at which it would need to operate would be immense. The dynamic nature of Grid resources would mean that the list of resources available would need to be constantly updated. Because the availability of resources is dynamic, they can be taken away from the users at any time which means that users may have to be constantly requesting access to further resources. In a Grid of world-wide scale, a single server to handle this would not be possible. As well as the problem of making the central server fast enough, it must also be so reliable that it can never break down. If it did stop working then the whole Grid would also have to stop - and even if some of the communication channels between it and certain sections of the Grid broke, that whole section would have no other server which it could access. Some distributed form of providing Resource Discovery is required for large Grids to operate reliably.

To solve this problem, the authors of [21] say that they have created distributed, dynamic ‘State Exchange Services’ system called *Gossips* which manage resource access and discovery and create and destroy themselves automatically. However, as stated there, not every Grid can use that system so more work is required in this area. Other current Grid systems do not address this problem at all (see, e.g. [1] and [20]) - but rely on centralised managers - so could not be scaled past a certain point.

5.9 Grid Programming Environments

Current Computer Aided Software Engineering (CASE) tools and programming languages have not been designed to facilitate the creation of Grid applications. What is considered to be high level in standard software development situations - Java, Message Passing Interfaces (MPI) - are referred to as low level in Grid publications [17]. This is because Grid Computing uses the abstractions provided by what are currently referred to as high-level layers - Virtual Machines, etc. - and extends them. For example Grid programmers should be able to treat a network as one huge computer and not have to worry about the individual virtual machine computers that make it up. This extra layer of abstraction should lead to new development environments and possibly things like new programming keywords - ‘remote’, ‘local’, ‘secure’, etc. The current trend toward component based development will continue with Grid applications being made up of different components at different sites. This could mean that huge data sets are stored at one place, analysis is done on the Grid, and visualisation is done somewhere else. The component based structure leads to the need for standard ways of storing and exchanging data, which current tools like XML provide, [8].

6 Grid Computing at the University of Canterbury

Grid Computing is not currently employed at the University of Canterbury (UC), but there are several research teams who would like to work on projects that could make extensive use of Grid Computing. This section outlines details of some of those projects and then the ways in which they could be activated.

6.1 Research Teams and Projects

These are projects of research teams from Physics and Astronomy (Prof. Philip Butler and Associate Prof. Lou Reinisch), Forestry (Dr. Hamish Cochrane), Biological Sciences (Associate Prof. Jack Heinemann) as well as from HIT Laboratory (Dr Mark Billingham). Their projects are considered to be so heavily computational that they are not suitable for desktop processing. In particular, the following projects have been planned:

- **Medical Imaging**

The Department of Physics and Astronomy is hoping to purchase a PET/CT scanner in the near future which would be used for Medical Imaging. Currently running the PET/CT software on a high-end desktop computer means that only about 10% of time is spent doing the scanning and the other 90% of the time is spent waiting for results. It is hoped that this ratio of scanning to processing time could be increased greatly using a Grid, with reduction of processing times at least ten times.

- **Bioinformatic Analysis and Genetic Data**

Researchers in the New Zealand Institute of Gene Ecology (NZIGE), which includes staff from the Department of Forestry and the School of Biological Sciences, as well as others, would also be ready to make use of a computational grid. The research that would use the grid would mostly involve (in very simple terms) searching for certain patterns on large data sets. This is a very slow process on standard workstations and any increase in speed would

be considered useful, with a speedup between 2 to 24 times being regarded as good, but anything further better, of course.

As well as these, it is envisaged that other projects would use the grid if it were available. Some other potential users are:

- Proteomics research.
- Processing data about imported foods on behalf of MAF. This looks for certain features of the foods but is currently a very slow process.
- Processing astronomical data from the several telescopes that the Department of Physics and Astronomy has access to.

6.2 Potential Grid Tools For UC

There are several tools that could be used to facilitate Grid Computing at the UC. All of the projects mentioned above have a focus on data processing rather than data access or any other Grid function, so this section will focus only on the data processing side of Grid Computing. Note that although most of these tools are not Computational Grids as defined earlier in this article they can still provide useful amounts of computing power (and fall into the realm of what is commonly called Grid Computing).

6.2.1 XGrid

XGrid is a distributed computing system that is currently installed on all Apple Macintoshes at UC. It claims to automatically detect the presence of other Apple Macs and to be capable of distributing processing to them without any explicit programming [23]. The degree to which this works would have to be investigated further, but although most of the computers on campus are not Macs, enough of them are for a fairly significant amount of processing power to be available from them if the *XGrid* system is effective.

6.2.2 Globus

As mentioned earlier, the Globus Toolkit [25] is often referred to as the *de facto* standard for creating Computational Grids. It is therefore logical that if a Grid is to be created at UC, the Globus Toolkit be used. The Globus Toolkit is not simply plugged in and used, however, unlike *XGrid*, but is used to *create* Grids [4]. For this reason, if the Globus Toolkit were to be used to create a Grid at UC, specialist programmers would have to be employed to put it all together. The advantage of the Globus Toolkit is that it is widely used and well understood and, compared to other tools, it is at least known to work and work well.

6.3 The Akaroa Project

Akaroa2 is an automated controller of stochastic discrete-event simulation developed at the University of Canterbury by the Simulation Research Group (the group led by Prof. K. Pawlikowski from Computer Science and Software Engineering, and Associate Prof. D. McNickle from Management), [22]. When Akaroa2 was designed at the University of Canterbury in 1992, it was one of the first software packages enabling grid processing. In 1993, it received an international commendation (in Science category) in the Computerworld Smithsonian Award for Achievements in Information Technology, USA.

Akaroa2 speeds up simulation experiments by performing multiple replications of the experiment in parallel (MRIP) on multiple computers of a LAN, with a simulation being stopped when the overall results have reached the desired level of statistical precision. It runs the different replications on different machines acting as simulation engines. Akaroa2 has been designed for

working on local area networks consisting of UNIX/Linux machines. Thus, the degree of its distributiveness is limited by the number of workstations in a given LAN. Currently, students of Computer Science and Software Engineering at the University of Canterbury can use Akaroa2 for distributing simulations utilizing about 250 workstations. Launching Akaroa2 on a Grid system would certainly be very desirable, since access to many more hosts could be possible. The next section investigates how this could be done.

6.3.1 PlanetLab

As mentioned, PlanetLab is not a Grid Computing system but is a global testbed for distributed computing systems [30]. The Department of Computer Science and Software Engineering at UC has maintained a node on PlanetLab, so any Grid projects conducted there could use the PlanetLab testbed. This could form a very good way of extending the Akaroa2 project - multiple simulations could be run on different parts of the world instead of on different machines in the same lab, although issues such as the effect of the increased time propagation delay and unreliable access to machines would need to be investigated. PlanetLab would also provide access to another several-hundred machines which could further increase the speed of simulation studies, and allow more complicated simulations to be carried out.

6.3.2 MPICH-G2 and Globus

MPICH-G2 is a grid-enabled implementation of the MPI standard [28]. MPI is a library specification for message-passing [27] which can be used for constructing portable parallel programs. Its goals are to provide portability and performance across many platforms [12] and, because it is aimed at being portable, it could be a good tool to use to modify Akaroa2. MPICH-G2 implements the MPI standard and extends it using tools from the Globus Toolkit, allowing the creation of Grid applications that run on multiple machines of potentially different architectures [28]. If Akaroa2 were extended using MPICH-G2, it could be run on multiple environments at once (ie. not just UNIX or Linux). This would greatly increase the potential processing power available to simulation applications. MPICH-G2 has C and C++ bindings which make it ideal for use with Akaroa2.

7 Conclusions

Grid Computing means sharing computing resources in order to create super-computing capabilities out of desktop computers by using their idle CPU time. It also involves sharing other computing resources such as data sets and disk storage. It has been around for several years and has reached the stage when there are tools available so that experts can create Computational Grids and use them to solve problems in many fields.

There are four vital issues which must be resolved in a distributed computing system before it can be called a Grid. These are Authentication, Authorisation, Resource Access and Resource Discovery. They lead to the idea of Virtual Organisations of collaborators who share resources over a Grid.

There are currently several tools available to help developers create Grids. The most widely used of these is the Globus Toolkit, but there are others. There are also several commercial companies which claim to provide Grid systems to clients.

Despite all the progress that has been made with Grid Computing, a number of challenges still exist. They must be faced now or in the future if Grid Computing is to succeed as a technology. These include the issue of many separate Grids versus a single world-wide Grid, addressing social issues of resulting from sharing computing resources (the idea of Grid Economics), security issues (allowing untrusted others to run code on your machine), problems with allocating resources (forecasting the performance of resources and creating a way of discovering resources without using a single central repository), and many others.

Grid Computing is well suited to some of the research that is being done, or is intended to

be done, at the University of Canterbury. Projects in Physics and Astronomy, Biological Sciences and Forestry especially could benefit from Grid Computing. There are several ways that a Grid could be constructed using different tools. As well as this, the **Akaroa2** project, a significant achievement in distributed computing, could be further enhanced by making it working in a Grid environment.

It was not long ago that very few people had even heard of the Internet, much less considered using it for their businesses. Many of those same companies could not operate without it now. Once a technology matures, it does not necessarily take long for it to become widespread. This same idea could be applied to Grid Computing - if some of the fundamental issues holding it back are addressed then computing power could truly become as widespread and easily accessible as electricity is now.

Acknowledgement

This work was supported by the University of Canterbury Summer Scholarship (Grant No. U1029).

References

- [1] Rongmei Zhang Ali Raza Butt and Y. Charlie Hu. A self-organizing flock of condors. In *Proceedings of the ACM/IEEE Supercomputing Conference*, page 42, November 15-21 2003.
- [2] Matthew S. Allen and Rich Wolski. The livny and Plank-Beck problems: Studies in data movement on the computational grid. In *Proceedings of the ACM/IEEE Supercomputing Conference*, page 266, November 15-21 2003.
- [3] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public-resource computing. *Communications of the ACM*, 45(11), November 2002.
- [4] Mario Cannataro and Domenico Talia. The knowledge grid. *Communications of the ACM*, 46(1):89–93, January 2003.
- [5] A.J. Lenstra D. Arkins, M. Graff and P.C. Leyland. The magic words are squeamish ossifrage. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology*, pages 263–277. Springer-Verlag, Asiacypt 1996.
- [6] Sambit Sahu Debanjan Saha and Anees Shaikh. A service platform for on-line games. In *Proceedings of the 2nd workshop on Network and system support for games*, pages 180–184, May 2003.
- [7] D.P. Schissel et al. Building the u.s. national fusion grid: Results from the national fusion collaborative project. *4th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research*, July 21-23 2003.
- [8] Ewa Deelman et al. Grid-based galaxy morphology analysis for the National Virtual Observatory. In *Proceedings of the ACM/IEEE Supercomputing Conference*, November 15-21 2003.
- [9] J. Frey et al. Condor-g: A computation management agent for multi-institutional grids. In *10th International Symposium on High Performance Distributed Computing*, pages 55–66, 2001.
- [10] Ken-ichi Kurata et al. Evaluation of unique sequences on the european data grid. *ACM Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003*, 19:43–52, 2003.

- [11] P. Eerola et al. Building a production grid in scandinavia. *IEEE Internet Computing*, 7(4):27–35, 2003.
- [12] William Gropp Ewing. Goals guiding design: Pvm and mpi.
- [13] Ian Foster and Carl Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Number 1558604758 in ISDN. Morgan Kaufmann, 1 edition, July 1998.
- [14] Li Gong. Java security: present and near future. *IEEE Micro*, 17(3):14–19, May-June 1997.
- [15] Carl Kesselman Ian Foster and Steven Tuecke. The anatomy of the grid - enabling scalable virtual organizations. *The International Journal of Supercomputer Applications*, 15(3):200–222, Fall 2001.
- [16] K. Keahey, T. Fredian, Q. Peng, D. P. Schissel, M. Thompson, I. Foster, M. Greenwald, and D. McCune. Computational grids in action: The national fusion collaboratory.
- [17] D. Laforenza. Grid programming: some indications where we are headed. *ACM Parallel Computing Special Issue: Advanced environments for parallel and distributed computing*, pages 1733–1752, 2002.
- [18] Marko Niinimaki and Vesa Sivunen. Applying Grid security and virtual organization tools in distributed publication databases. In *Proceedings of the 1st international symposium on Information and communication technologies*, pages 543–548, September 2003.
- [19] Jennifer M. Schopf. Grids: The top ten questions. In *International Symposium on Grid Computing*, March 2003.
- [20] Y. M. Teo, S. C. Tay, and J. P. Gozali. Distributed geo-rectification of satellite images using grid computing. In *International Parallel and Distributed Processing Symposium*, page 15b, April 22-26 2003.
- [21] Rich Wolski, John Brevik, Chandra Krintz, Graziano Obertelli, Neil Spring, and Alan Su. Running EveryWare on the computational grid. In *Supercomputing Conference on High-performance Computing*, 1999.
- [22] V. Yau and K. Pawlikowski. Akaroa: a package for automatic generation and process control of parallel stochastic simulation. In *Proceedings of the 16th Australian Computer Science Conference*, volume A, pages 71–82, February 1993.
- [23] <http://www.apple.com/acg/xgrid/>
- [24] <http://www.eu-datagrid.org>
- [25] <http://www.globus.org>
- [26] <http://www.gridforum.org>
- [27] <http://www.mcs.anl.gov/mpi/>
- [28] <http://www.niu.edu/mpi/>
- [29] <http://www.parabon.com>
- [30] <http://www.planet-lab.org>
- [31] <http://www.ud.com>