Chapter 1

Introduction

This chapter provides a high-level overview of the application of an economic-based model to Grid resource management and scheduling. It briefly presents the inspiration for computational Grids and our work. Then, it summarises the key components of economic-based distributed resource management and application scheduling and presents primary contributions of our research. The chapter ends with a discussion on the organization of the rest of this dissertation.

1.1 Inspiration for Computational Grids

Following Alessandro Volta's invention of the electrical battery in 1800, Thomas Edison and Nikola Tesla paved the way for electricity's widespread use by inventing the electric bulb and alternating current (AC) respectively. Figure 1.1 shows Volta demonstrating the battery for Napoleon I in 1801 at the French National Institute, Paris. Whether or not Volta envisioned it, his invention evolved into a worldwide electrical power Grid that provides dependable, consistent, and pervasive access to utility power and has become an integral part of modern society.

Figure 1.1: Volta demonstrates the battery for Napoleon I at the French National Institute, Paris, in 1801. The painting (by N. Cianfanelli, 1841) is from the Zoological Section of "La Specula" at the National History Museum, Florence University, Italy.

Inspired by the electrical power Grid's pervasiveness, ease of use, and reliability, computer scientists in the mid-1990s began exploring the design and development of an analogous infrastructure called the *computational power Grid* [48] for wide-area parallel and distributed computing. The motivation for computational Grids was initially driven by large-scale, resource (computational and data) intensive scientific applications that require more resource than a single computer (PC, workstation, supercomputer, or cluster) could provide in a single administrative domain. A Grid enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce.

To build a Grid, the development and deployment of a number of services is required. They include *low-level services* such as security, information, directory, resource management (resource trading, resource allocation, quality of services) and *high-level services/tools* for application development, resource management and scheduling (resource discovery, access cost negotiation, resource selection, scheduling strategies, quality of services, and execution management) [48][73][100][98][99]. Among them, the two most challenging aspects of Grid computing are resource management and scheduling. This thesis presents a distributed computational economy framework and algorithms for the management of resources and scheduling of applications that are driven by the Quality-of-Service (QoS) requirements of the users.

1.2 Economic-based Grid Resource Management and Scheduling

Grid [48] and Peer-to-Peer (P2P) [4] computing platforms enable the sharing, selection, and aggregation of geographically distributed heterogeneous resources—such as computers and data sources—for solving large-scale problems in science, engineering, and commerce. However, resource management and scheduling in these environments is a complex undertaking. The geographic distribution of resources owned by different organizations with different usage policies, cost models and varying load and availability patterns is problematic. The *producers (*resource owners) and *consumers* (resource users) have different goals, objectives, strategies, and requirements. To address these resource management challenges, we have proposed and developed a distributed computational economy-based[1] framework, called the **Gr**id **A**rchitecture for **C**omputational **E**conomy (GRACE) [98][99], for resource allocation and to regulate supply and demand of the available resources. This economic-based framework offers an incentive to resource owners for contributing and sharing resources, and motivates resource users to think about trade-offs between the processing time (e.g., deadline) and computational cost (e.g., budget), depending on their QoS requirements. We believe that this approach is essential for promoting the Grids as a mainstream computing paradigm, which could lead to the emergence of a new service-oriented computing industry.

The resource management and scheduling systems for Grid computing need to manage resources and application execution depending on resource consumers' and owners' requirements, and they need to continuously adapt to changes in the availability of resources. This requirement introduces a number of challenging issues that need to be addressed; namely, site autonomy, heterogeneous substrate, policy extensibility, resource allocation or co-allocation, online control, resource trading, and quality of service-based scheduling. A number of Grid systems (such as Globus [63]) have addressed many of these issues with the exception of *resource trading* and *quality of service-based scheduling.* The GRACE framework has been proposed to address, particularly, these two issues. The GRACE architecture leverages existing technologies such as Globus, and it provides new services that are essential for resource trading and aggregation, depending on their availability, capability, cost, and users' QoS requirements. Therefore, this thesis mainly focuses on three aspects of resource trading and quality of service-based scheduling. First, to develop a generic distributed computational economy architectural framework and strategies for resource trading using different economic models. Second, to develop an advanced resource broker and QoS driven scheduling algorithms for deploying applications on wide-area distributed resources. Finally, to develop a comprehensive Grid simulation toolkit that supports *repeatable* performance evaluation of scheduling strategies for a range of Grid scenarios.

The idea of applying economics to resource management in distributed systems has been explored in previous research to help understand the potential benefits of market-based systems. For example, Spawn [14], Popcorn [87], Java Market [144], Enhanced MOSIX [145], JaWS [125], Xenoservers [23], D'Agents

[1] The terms "economic/economy-based" and "market-based" are synonymous and interchangeable.

[55], Rexec/Anemone [8], Mojo Nation [84], Mariposa [83], and Mungi [33]. Unfortunately, many of them were limited to experimental simulations. The systems that were implemented followed a monolithic architecture, which means they are hard to extend. They expect the users to develop resource-aware applications explicitly for their platform using their new programming interface (e.g., Spawn and Popcorn). Consequently, developing applications for such platforms is difficult because programmers have to address both the application development and resource allocation issues concurrently. This problem is overcome in Nimrod-G by separating application development and resource management issues. To enable the creation of parameter parallel/sweep applications, Nimrod-G provides a simple parameter specification language. The execution of such applications is managed by Nimrod-G, where the users define their quality of service requirements such as the deadline, budget, and optimisation preference; and the Nimrod-G broker automatically handles the allocation of budget for each job and execution.

1.2.1 Assessing Wants and Needs

In an economic-based Grid computing environment, resource management systems need to provide mechanisms and tools that allow resource consumers (end users) and providers (resource owners) to express their requirements and facilitate the realization of their goals. Resource consumers need:

- a utility model—how consumers demand resources and their preference parameters, and
- a broker that supports resource discovery and strategies for application scheduling on distributed resources dynamically at runtime depending on their availability, capability, and cost along with user-defined QoS requirements.

The resource providers need:

- tools and mechanisms that support price specification and generation schemes to increase system utilization, and
- protocols that support service publication, trading, and accounting.

For the market to be competitive and healthy, coordination mechanisms are required to help reach equilibrium price—the market price at which the supply of a service equals the quantity demanded.

Numerous economic theories and models including micro- and macro-economic principles have been proposed. They include,

- commodity market models,
- posted price models,
- bargaining models,
- tendering or contract-net models,
- auction models,
- bid-based proportional resource sharing models,
- cooperative bartering models,
- monopoly and oligopoly.

A detailed discussion on the use of these economic models within the GRACE framework can be found in Chapter 3.

1.2.2 The Nimrod-G Grid Resource Broker

Nimrod-G is a computational economy-based global Grid resource management and scheduling system that supports deadline- and budget-constrained algorithms for scheduling parameter sweep (task and data parallel) applications on distributed resources [100]. It provides a simple *parameter specification language* for creating parameter-sweep applications. The domain experts (application-specific experts) can create a *plan* for parameter studies and use the Nimrod-G broker to handle all the issues related to the seamless management and execution, including resource discovery, mapping jobs to appropriate resources, data and code staging, and gathering results from multiple Grid nodes back to the home node[2]. Depending on the user's requirements, it dynamically leases Grid services at runtime based on their availability, capability,

[2] A node from which a job request originates.

and cost.

A diagram of high-level architecture and components of Nimrod-G is shown in Figure 1.2. The components of Nimrod-G are:

- A persistent task farming engine,
- A Grid explorer for resource discovery,
- A resource trading manager for establishing access price,
- A schedule advisor that maps jobs to resources using deadline and budget constrained scheduling algorithms.
- A dispatcher and actuators for deploying agents on Grid resources,
- Agents for managing execution of Nimrod-G jobs on Grid resources.

When Nimrod-G deploys an agent on the Grid node at runtime, it is submitted to the local resource manager, which then allocates a compute node[3] to it for executing the job.

Nimrod-G provides a persistent Task-Farming Engine (TFE), which supports job management protocols and APIs. It can be used to create and plug-in user-defined scheduling policies and customized problem solving environments. For example, ActiveSheets [20] uses the Nimrod-G broker services to execute Microsoft Excel computations and cells on the Grid. The TFE coordinates resource trading, scheduling, data staging, execution, and gathering results from remote Grid nodes to the user's home transparently.

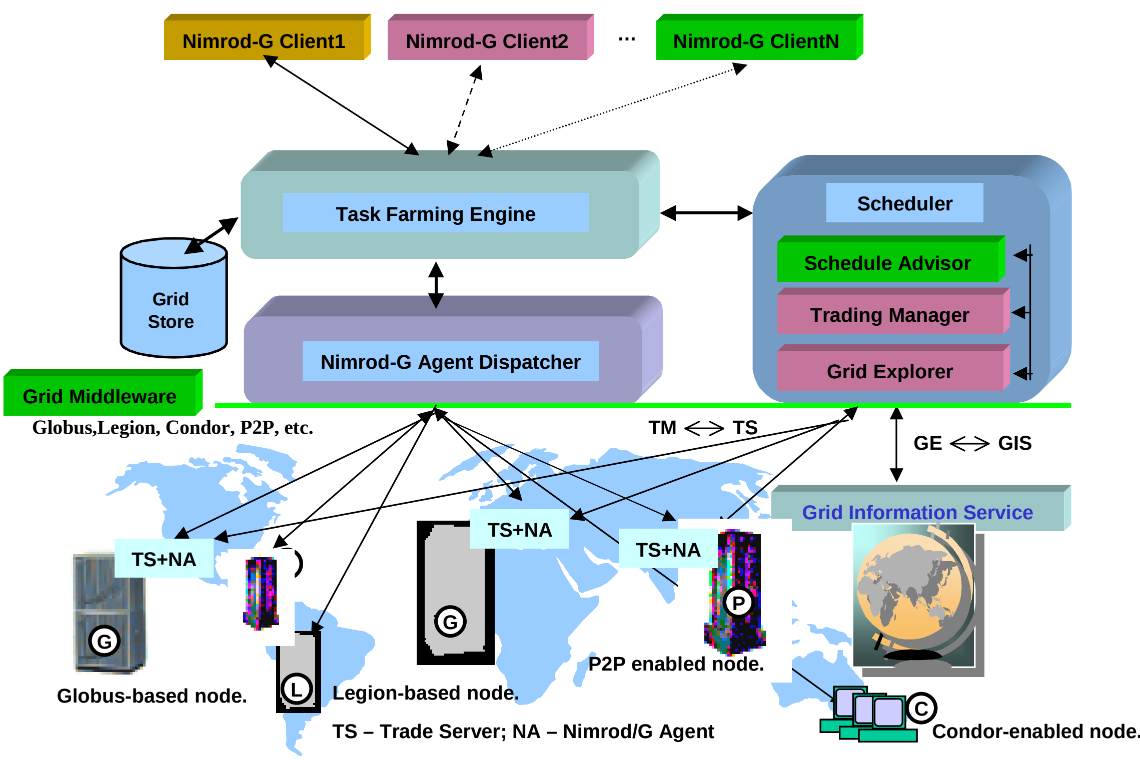

Figure 1.2: Nimrod-G Grid resource broker.

An associated dispatcher is capable of deploying computations (jobs) on Grid resources enabled by Globus [49], Legion [2], and Condor [79]. We have developed a number of deadline-based, market-driven scheduling algorithms: cost, time, conservative-time, and cost-time optimizations with deadline and budget constraints [105]. Once a price is established using a suitable economic model, depending on users' requirements, our scheduling algorithms can lease resource services depending on their availability, capability, cost, and user-level performance.

[3] A node/CPU that the local resource manager allocates to the Nimrod-G agent for job processing.

We have used the Nimrod-G broker to schedule data-intensive computational applications (such as molecular modelling for drug design) on the World-Wide Grid (WWG) testbed that has resources located in five continents. The results of our scheduling experiments can be found in Chapter 4. A complete discussion of the use of Nimrod-G tools for formulating drug design application as a parameter sweep application and processing them on the Grid can be found in Chapter 7.

1.2.3 GridSim Toolkit and Economic Grid Broker Simulator

In order to prove the effectiveness of resource brokers and associated scheduling algorithms, their performance needs to be evaluated under different scenarios such as varying the number of resources and users with different requirements. In a real Grid environment, it is hard, and perhaps even impossible, to perform scheduler performance evaluation in a repeatable and controllable manner for different scenarios—the availability of resources and their load continuously varies with time and it is impossible for an individual user/domain to control activities of other users in different administrative domains. To overcome this limitation, we have developed a Java-based discrete-event Grid simulation toolkit called GridSim. This toolkit supports modeling and simulation of heterogeneous Grid resources (both time- and space-shared), users, brokers, and application models. It provides primitives for the creation of application tasks, the mapping of tasks to resources, as well as their management. A detailed discussion of GridSim can be found in Chapter 5.

Using the GridSim toolkit, we developed an economic Grid resource broker that simulates the Nimrod-G broker. The simulator implements various "Deadline and Budget Constrained" (DBC) cost, time, cost-time, and conservative-time optimisation scheduling algorithms. We have simulated the WWG testbed resources and hypothetical task farming applications, to evaluate the performance of the scheduling algorithms through a series of simulations, by varying the number of users, deadlines, budgets, and optimisation strategies. The results of scheduling simulations can be found in Chapter 6.

1.3 Contributions

To support the thesis that economic-based Grid resource management and scheduling systems can deliver significant value to users, resource providers and consumers, compared to traditional system-centric approaches, we have made several novel research contributions. They are as follows:

1. The thesis identifies and proposes a distributed computational economy paradigm for effective management of resources in Grid computing environments by handling the large-scale heterogeneity, distribution, and decentralization inherent in them. This paradigm provides incentive to resource owners to contribute resources and motivates resource users to consider the trade-off between the time of results delivery and computational cost, depending on their quality of service requirements. It also helps in the regulation of supply-and-demand for resources. To realize this, we have developed a distributed computational economy framework called the **Gr**id **A**rchitecture for **C**omputational **E**conomy (GRACE). The architecture is generic enough to accommodate different economic models and maps well onto the architecture of modern wide-area distributed systems. Its implementation leverages many existing technologies and provides additional services for resource trading and their aggregation. Thereby, we are able to abstract away implementation details and focus on how the system is delivering value to resource owners and consumers.
2. The thesis articulates the three key functionalities that economic-based Grid resource management and scheduling systems must support in order to increase the value of the utility and quality of services delivered to users (i.e., both resource providers/owners and consumers/end-users). That is, our framework provides (i) a means to express their requirements, valuations, and objectives, (ii) scheduling policies to translate them to resource allocations, and (iii) mechanisms to enforce the selection and allocation of differential services, and dynamic adaptation to changes in their availability at runtime. Grids need to use *competitive economic models* as different resource providers and resource consumers have different goals, objectives, strategies, and requirements that vary with time. Essentially, in market-based Grid systems, resource consumers adopt the strategy of solving their problems at low cost within a required timeframe and resource providers adopt the strategy of obtaining the best possible return on their investment. The user's valuation of the utility of the same work and resources are time dependent. For example, end users with an

immediate production requirement value their computations much higher than others; the resource owners charge a higher price when there is a high demand for resources, and a lower price when the demand is low.

3. The thesis presents the design and development of the Nimrod-G prototype system that realises the system architecture by exploiting lower-level services provided by existing Grid technologies such as Globus and Legion. It uses a component-based layered architecture that enables the deployment of Nimrod-G on any emerging low-level Grid and P2P technologies with a minimal development effort. To realise the ultimate goal of delivering quality of services to users, the Nimrod-G Grid resource broker supports the deadline and budget constrained scheduling with user selected optimisation strategies, for processing large-scale task-and-data parallel (parameter-sweep) applications on globally distributed resources. The broker is capable of dynamically leasing Grid services/resources at runtime depending on their cost, capabilities, availability, and users' requirements. The broker has been deployed and used for scheduling large-scale applications (such as molecular modelling for drug design) on the WWG testbed.

4. The thesis presents four different **D**eadline and **B**udget **C**onstrained (DBC) scheduling algorithms based on cost, time, conservative-time, and cost-time optimisations. The DBC *cost-optimisation* scheduling algorithm completes application processing by the deadline and minimizes the computational cost; the *time-optimisation* scheduling algorithm strives to complete application processing earlier than the deadline and within the budget limit; *conservative-time optimisation* is similar to time-optimisation, but ensures that each job has a minimum budget-per-job allocated and surplus is moved to other jobs only after their completion; and the *cost-time optimisation scheduling* algorithm is similar to cost-optimisation and tries to optimise for time without incurring extra expenses. The first three algorithms have been implemented within the Nimrod-G resource broker and a series of scheduling experiments have been conducted on the WWG testbed for different combinations of deadline and budget parameters, application workloads, and resources with varying prices.

5. The thesis presents the design and development of a toolkit, called GridSim, that supports discrete-event based simulation of Grid environments that allows *repeatable* performance evaluation under different scenarios, which is not possible in a real Grid environment as the availability of resources and their load continuously varies with time. The toolkit supports modeling and simulation of heterogeneous Grid resources (both time- and space-shared), users, brokers, and application models. It provides primitives for creation of application tasks, mapping of tasks to resources, and their management. A Nimrod-G like economic Grid resource broker is being simulated using the GridSim toolkit to evaluate the performance of deadline and scheduling algorithms through a series of simulations. It is achieved by varying the number of users, deadline, budget, and optimisation strategies and simulating geographically distributed Grid resources that resemble the WWG testbed. The results at microscopic level reveal their impact on the application processing cost and time, depending on user's requirements and valuations. They demonstrate the usefulness of allowing users to trade-off between the timeframe and processing cost depending on their QoS requirements.

6. The thesis demonstrates the effectiveness and application of Grid technologies and the Nimrod-G resource broker for solving real-world problems by creating a virtual laboratory environment for distributed molecular docking. The molecular docking application screens compounds in the Chemical DataBases (CDBs) to identify their potential as drug candidates. Using the Nimrod-G parameter specification language, the docking application's data files have been parameterized to formulate it as SPMD (single program multiple data) model-based parallel application. The virtual laboratory enables remote access to domain-specific databases (e.g., CDB) as a network service. The two scheduling experiments with cost and time optimisations using Nimrod-G for processing molecular docking jobs on the WWG testbed, demonstrate that users can express their valuations naturally by defining deadline, budget limits, and optimisation preference. The fact that users have the option of expressing their requirements, allows them to trade-off between the time for results delivery and the cost of computations, depending on the perceived value of utility at that time.

1.4 Organization

The rest of this thesis is organized as follows. Chapter 2 presents state-of-the-art Grid technologies from areas concerned with traditional and computational economy based resource management systems. Chapter 3 proposes computational economy as a metaphor for effective management of distributed resources and application scheduling. A distributed computational economy framework, called the Grid Architecture for Computational Economy (GRACE), leverages existing technologies and provides additional services for resource trading and aggregation. It discusses the use of real-world economic models and strategies: commodity market, posted prices, bargaining, tendering, auction, proportional resource sharing, and cooperative bartering for resource management and scheduling within the GRACE framework.

Chapter 4 presents architecture and implementation of the Nimrod-G resource broker that uses a computational economy driven framework for managing resources and scheduling applications. It discusses the deadline and budget constrained scheduling algorithms and the results of scheduling parameter sweep applications on the World-Wide Grid resources using the Nimrod-G resource broker.

Chapter 5 discusses the design and implementation of GridSim, a toolkit for modelling and simulation of resources and application scheduling in large-scale parallel and distributed computing environments. Given this simulation toolkit, Chapter 6 briefly presents the development of an economic Grid broker simulator. Then, we discuss the DBC scheduling algorithms and their performance evaluation through a series of simulations by varying the number of users, deadline, budget, and optimisation strategies and simulating geographically distributed Grid resources.

Chapter 7 presents the design and development of a virtual laboratory environment that enables molecular modelling for drug design on the Grid using Nimrod-G tools. Finally, Chapter 8 presents conclusions and provides directions for future work.

1.5 Acknowledgements

During the three years of my Ph.D. candidature at Monash University, I have collaborated with a number of colleagues within the university and outside. The work resulted in joint publications from which some of the chapters are *partially* derived. Note that *"I have only used the material that I contributed"*, which is directly relevant to my research.

Chapter 2 is *partially* derived from the following joint publications.

- M. Baker, R. Buyya, and D. Laforenza, *The Grid: International Efforts in Global Computing*, International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR 2000), l'Aquila, Rome, Italy, July 2000.
- K. Krauter, R. Buyya, and M. Maheswaran, *A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing*, International Journal of Software: Practice and Experience (SPE), Wiley Press, New York, USA, May 2002. (to appear).

Chapter 3 is *partially* derived from the following joint publications.

- R. Buyya, D. Abramson, and J. Giddy, *Economy Driven Resource Management Architecture for Computational Power Grids*, International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000), Las Vegas, USA, 2000.
- R. Buyya, J. Giddy, and D. Abramson, *A Case for Economy Grid Architecture for Service-Oriented Grid Computing*, 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS 2001), San Francisco, California, USA, April 2001.
- R. Buyya, H. Stockinger, J. Giddy, and D. Abramson, *Economic Models for Management of Resources in Peer-to-Peer and Grid Computing*, SPIE International Conference on Commercial Applications for High-Performance Computing, August 20-24, 2001, Denver, USA. An extended version to appear in *The Journal of Concurrency and Computation: Practice and Experience* (CCPE), Wiley Press, 2002.

  Comments: H. Stockinger (CERN) has helped me to enhance the paper and contributed a section on Data Grid economy, which I have *not* included in the chapter.

Chapter 4 is *partially* derived from the following joint publications.

- R. Buyya, D. Abramson, and J. Giddy, *Nimrod-G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid*, The 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000), Beijing, China.
- R. Buyya, J. Giddy, and D. Abramson, *An Evaluation of Economy-based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications*, The Second Workshop on Active Middleware Services (AMS 2000), in conjunction with HPDC 2001, August 1, 2000, Pittsburgh, USA (Kluwer Academic Press).
- R. Buyya, J. Giddy, and D. Abramson, *A Case for Economy Grid Architecture for Service-Oriented Grid Computing*, 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS 2001), San Francisco, California, USA, April 2001.

  Comments: J. Giddy and I worked as members of the Nimrod-G project led by my PhD supervisor (Prof. D. Abramson). I have contributed towards the Nimrod-G architecture, distributed computational economy methodologies, scheduling strategies, and performed scheduling experiments on the World-Wide Grid testbed. J. Giddy has played a major role in implementing some of those concepts within Nimrod-G.

Chapter 5 and 6 are *partially* derived from the following joint publications.

- R. Buyya and M. Murshed, *GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing*, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, USA, May 2002.
- R. Buyya, M. Murshed, and D. Abramson, *A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids*, 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), June 24 - 27, 2002, Las Vegas, USA.

  Comments: I have collaborated with M. Murshed in developing the GridSim Toolkit discussed in Chapter 5. While developing GridSim base module, I worked with M. Murshed and explored the use of discrete-event simulation techniques with SimJava package. I have developed and simulated a Nimrod-G like economic Grid resource broker using the GridSim toolkit, designed, developed, and evaluated performance of scheduling algorithms.

Chapter 7 is *partially* derived from the following joint publication.

- R. Buyya, K. Branson, J. Giddy, and D. Abramson, *The Virtual Laboratory: Enabling Molecular Modeling for Drug Design on the World Wide Grid*, Technical Report, Monash-CSSE-2001-103, Monash University, December 2001. An extended version to appear in *The Journal of Concurrency and Computation: Practice and Experience* (CCPE), Wiley Press, May 2002.

  Comments: I have collaborated with K. Branson and developed the *Virtual Laboratory* environment for enabling the processing of Drug Design application on the Grid using the Nimrod-G broker. He has provided me with necessary application data and Nimrod scripts, which he explored on a Linux cluster.