UNIVERSITY
of
GLASGOW

Department of Physics & Astronomy
Experimental Particle Physics Group
Kelvin Building, University of Glasgow,
Glasgow, G12 8QQ, Scotland
Telephone: +44 (0)141 339 8855 Fax: +44 (0)141 330 5881

GLAS-PPE/2003-14
24th September 2003

UK Grid Simulation with OptorSim

David G. Cameron[1], Rubén Carvajal-Schiaffino[2],
A. Paul Millar[1], Caitriana Nicholson[1], Kurt Stockinger[3], Floriano Zini[2]
[1] University of Glasgow, Glasgow, G12 8QQ, Scotland
[2] ITC-irst, Via Sommarive 18, 38050 Povo (Trento), Italy
[3] CERN, European Organization for Nuclear Research, 1211 Geneva, Switzerland

Abstract

As the computational and data handling requirements of large scientific collaborations grow, Grid computing is rapidly emerging as a feasible solution to these requirements. Optimising the use of Grid resources is crucial, and to evaluate potential optimisation strategies it is important to simulate them as realistically as possible before they are used on real Grids. We have developed the Grid simulator OptorSim and used it to test several optimisation strategies using a set of performance metrics. In this paper we consider the effects of several scheduling and replica optimisation strategies and base our simulation environment on the UK Grid for Particle Physics (GridPP).

UK e-Science All Hands Meeting 2003
Nottingham, UK

1 Introduction

GridPP [2] is a collaboration of particle physicists and computing scientists from the UK and CERN, who are building a Grid for particle physics. It is designed primarily for the analysis of large amounts of data from high energy physics experiments such as the LHC experiments at CERN. Data is the most important resource in this Grid, where users' jobs require access to a large quantity of data distributed across geographically diverse Grid sites.

Intelligent job scheduling and data replication are key tools in maximising the overall throughput of the Grid. An efficient scheduling strategy should be able to ensure jobs are submitted to Grid sites where the time spent waiting to be executed and the execution time are minimised. The replication strategy should be able to (a) determine the "best" replica, when given a request by a job for a particular file and (b) trigger both replication and deletion of files by analysing patterns of previous file requests.

The Grid simulator OptorSim [4] was designed to test various optimisation strategies in a simulated Grid environment before they are deployed in the real Grid. Many other Grid simulators have been developed recently, including ChicagoSim [10, 11], EDGSim [1], GridSim [6], and GridNet [9]. However, these simulators generally concentrate on the problem of optimising job scheduling in a Grid environment, whereas we combine this with optimisation of replication strategies to enable the best performance from all the Grid's resources.

In this paper we present some results from OptorSim, which show the effectiveness of several scheduling and replication strategies on the simulated GridPP environment under a range of conditions. Evaluation of scheduling and replication strategies is performed using a number of metrics including mean job time and usage of computing and network resources.

2 Simulation Environment

Given (a) a Grid topology and resources, (b) a set of jobs that the Grid must execute and (c) an optimisation strategy, OptorSim simulates what would happen in the Grid if the optimisation strategy were in use. It provides us with a set of measurements used to quantify the effectiveness of the strategies.

2.1 Grid Architecture

In OptorSim we adopt a Grid structure based on a simplification of the architecture proposed by the EU DataGrid project [3].

The Grid consists of several *sites*, each of which may provide resources for submitted jobs. Computational and data-storage resources are called *Computing Elements* (CEs) and *Storage Elements* (SEs) respectively. Computing Elements run jobs that use the data in files stored on Storage Elements. A *Resource Broker* controls the scheduling of jobs to Computing Elements.

Sites without Computing or Storage Elements act as network nodes or routers. Grid sites are connected by *Network Links*, each of which has a certain bandwidth. A *Replica Manager* at each site manages the data flow between sites and interfaces between the computing and storage resources and the Grid. The *Replica Optimisation Agent* (or *Optimiser*) inside the Replica Manager is responsible for both replica selection and the automatic creation and deletion of replicas. Replica optimisation is performed in a distributed way via the interaction of Optimisers located at each Grid site. An Optimiser performs local replica optimisation; the aim is to achieve global optimisation as the emergent result of local optimisation.

2.2 Optimisation Strategies

The Resource Broker uses a scheduling algorithm to calculate the cost of running a job on a group of candidate sites. It then submits the job to the site with the minimum estimated cost. The algorithms we test are based on the estimated data access time for the job at each site, the size of the queue at each site, or a combination of both. The following scheduling algorithms are analysed:

- *Random:* Schedule randomly to a CE.
- *Shortest Queue:* Schedule to the CE with the shortest job queue.

- *Access Cost:* Schedule to the CE where the job has minimal file access cost.
- *Queue Access Cost:* Schedule to the CE where the sum of the access cost for the job itself and the access costs of all jobs in the queue is smallest.

As for replica optimisation strategies, in this paper we consider three specific strategies: a traditional LFU (Least Frequently Used)-based strategy and two economy-based strategies [8].

The LFU-based strategy will always replicate files to the Storage Element local to the job's Computing Element. Replica Selection is achieved using a Replica Catalogue look-up to locate all replicas. After examining the current network state, the replica that can be accessed in the shortest time is chosen. If the local SE is full, the file that has been accessed the least number of times in the previous time window is deleted, creating space for the new replica.

The two economy-based strategies use prediction functions, one binomial-based [4] and the other Zipf-based [7], to calculate the file usefulness used in the replication and file replacement decisions. Relative file values are calculated based on the file access history stored by each Optimiser. If the potential replica under consideration has a higher value than the lowest value file currently in the local SE, that file is deleted and the new replica is "bought". Replica Selection is based on the auction protocol described in [5] for buying and selling files.

2.3 Evaluation Metrics

In this paper we consider the following measures in the evaluation of Grid optimisation strategies.

- The *mean job execution time* is defined as the total time to execute all the Grid jobs divided by the number of jobs completed.
- We define *effective network usage* r_{ENU}:

$$r_{\mathrm{ENU}} = \frac{N_{\text{remote file accesses}} + N_{\text{file replications}}}{N_{\text{local file accesses}}},$$

  where $N_{\text{remote file accesses}}$ is the number of times the CE reads a file from a SE on a different site, $N_{\text{file replications}}$ is the total number of file replications that take place and $N_{\text{local file accesses}}$ is the number of times a CE reads a file from a SE on the same site (we assume infinite bandwidth within a site). For a given network topology, a lower value of r_{ENU} indicates the optimisation strategy is better at replicating files to the correct location.

- We define computational power usage as the percentage of time that a CE is running jobs or otherwise active. Henceforth, we use the term *CE usage*, which is the total computational power usage for all the CEs on the Grid.

3 Simulation Setup

To test the performance of these strategies we simulate the proposed GridPP 2004 testbed, which has the network topology and resources shown in Figure 1. It comprises 17 Grid sites in the UK and one at CERN in Switzerland. Each UK site has a storage capacity between 5TB and 500TB[1)] and between 40 and 1800 processing nodes. CERN has 1000TB of storage and is used to hold all the master files at the beginning of the simulation. A simulated job was defined as reading and processing sequentially a prescribed list of files. To simplify the simulation we assumed a constant time to process each file, i.e. the analysis of each file was not modelled in detail. Six high energy physics experiments are involved in GridPP; to simulate a realistic workload we used between 200 and 400 1GB files per experiment[2)] and defined 7-10 jobs per experiment. The probability of a job being submitted to the Grid was inversely proportional to the number of files required by the job (typical of most high energy physics workloads).

[1)] For simulation purposes the storage capacity of each site was scaled down by a factor of 100.

[2)] The number of files per experiment was also scaled down by a factor of 100 compared to realistic high energy physics analysis jobs.

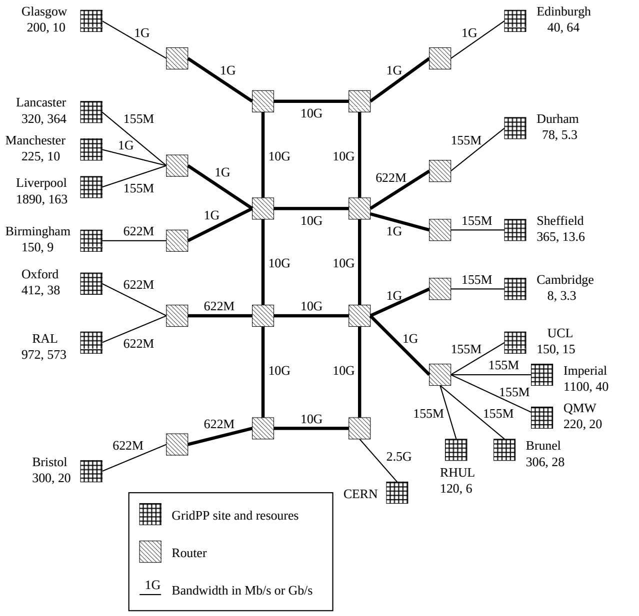

Figure 1: GridPP resources and topology in 2004. The numbers next to each site state the CPU capacity in kSI2000 and storage space in TB respectively.

4 Results

In this section we present simulation results. The measurements described in Section 2.3 are used as indicators of how well each strategy performs.

4.1 Scheduling Strategies

We start by studying the impact of the scheduling algorithm used by the Resource Broker. We ran the simulation with 1000 jobs submitted at 5 second intervals. Results showing the mean job time and CE usage for the scheduling strategies described in section 2.2 are shown in Figure 2.

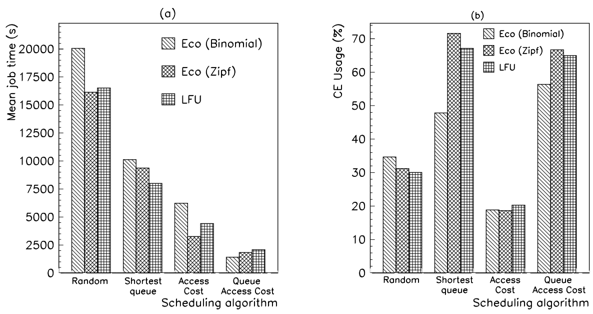

Figure 2: (a) Mean job time and (b) CE usage for various optimisation algorithms.

Overall, *random* scheduling gives the worst performance with mean job times roughly twice as high as the next worst algorithm, *shortest queue*, for all replica optimisation strategies. The *Access Cost* algorithm has a lower mean job time than these two but has the lowest CE usage, due to the fact that jobs are only scheduled to sites with high network connectivity. The mean job time is lowest and CE usage is highest when we use the *Queue Access Cost* algorithm. This gives the best balance between scheduling jobs close to the data whilst ensuring that sites with high network connectivity are not overloaded and sites with poor connectivity are not idle.

We therefore use the *Queue Access Cost* scheduling algorithm for all further tests.

4.2 Replication Strategies

We now demonstrate the scalability of each replica optimisation strategy by varying the number of submitted jobs (Figure 3).

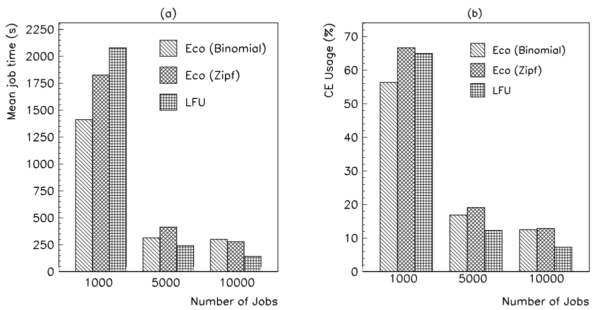

Figure 3: (a) Mean job time and (b) CE usage for different number of submitted jobs.

There is a large drop in the mean job time when the number of jobs submitted is increased, the LFU-based strategy being the most affected. The binomial economic model, which is $\sim 30\%$ faster than the LFU with 1000 jobs, is slightly slower when more jobs are included. However, the economic models still make better use of the Grid resources, with the CE usage for 10000 jobs $\sim 70\%$ higher than LFU.

OptorSim includes the simulation of non-Grid background traffic. Here, we examine the effect this has on Grid performance by comparing results with and without the inclusion of background (Figure 4). As expected,

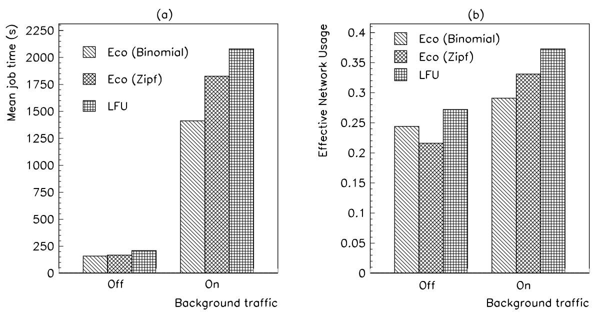

Figure 4: Effects of background network traffic on (a) mean job time and (b) effective network usage.

there is a large increase of a factor of around 7-10 in mean job time when we simulate the background network traffic; the effective network usage also increases slightly. The binomial-based economic model changes the least, showing that it is the most stable to fluctuations in the Grid environment.

5 Conclusion

In this paper we have shown that scheduling and replication strategies play a fundamental role in the optimisation of resource usage in a Data Grid. In particular, our experiments highlight that when scheduling jobs it is important to account for both the workload of computing resources and the location of the required data. For replica optimisation, we have shown that for many situations the economy-based strategies we have developed have the greatest effect in reducing job times and getting the most out of the resources available, while being robust to fluctuations in the non-Grid network traffic. The economic models were even more efficient when OptorSim was applied to a different Grid configuration in [7]. They have thus given promising results with two very different testbeds; we intend to investigate them further both with OptorSim and in a real Grid environment.

Acknowledgments

This work was partially funded by the European Commission program IST-2000-25182 through the EU DataGrid Project, the ScotGrid Project and PPARC.

References

[1] EDGSim: A Simulation of the European DataGrid. `http://www.hep.ucl.ac.uk/~pac/EDGSim/`.

[2] GridPP: The Grid for UK Particle Physics. `http://www.gridpp.ac.uk/`.

[3] The European DataGrid Project. `http://www.edg.org`.

[4] W. H. Bell, D. G. Cameron, L. Capozza, P. Millar, K. Stockinger, and F. Zini. Simulation of Dynamic Grid Replication Strategies in OptorSim. *Int. Journal of High Performance Computing Applications*, 17(4), 2003.

[5] W. H. Bell, D. G. Cameron, R. Carvajal-Schiaffino, P. Millar, K. Stockinger, and F. Zini. Evaluation of an Economy-Based File Replication Strategy for a Data Grid. In *Int. Workshop on Agent based Cluster and Grid Computing at Int. Symp. on Cluster Computing and the Grid (CCGrid 2003)*, Tokyo, Japan, May 2003. IEEE CS Press.

[6] 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*, pages 1–32, May 2002. Wiley Press.

[7] D. G. Cameron, R. Carvajal-Schiaffino, P. Millar, C. Nicholson, K. Stockinger, and F. Zini. Evaluating Scheduling and Replica Optimisation Strategies in OptorSim. In *Proc. of 4th International Workshop on Grid Computing (Grid2003)*, Phoenix, USA, November 2003. IEEE CS Press.

[8] M. Carman, F. Zini, L. Serafini, and K. Stockinger. Towards an Economy-Based Optimisation of File Access and Replication on a Data Grid. In *Int. Workshop on Agent based Cluster and Grid Computing at Int. Symp. on Cluster Computing and the Grid (CCGrid 2002)*, Berlin, Germany, May 2002. IEEE CS Press.

[9] H. Lamehamedi, Z. Shentu, B. Szymanski, and E. Deelman. Simulation of Dynamic Data Replication Strategies in Data Grids. In *Proc. 12th Heterogeneous Computing Workshop (HCW2003)*, Nice, France, April 2003. IEEE CS Press.

[10] K. Ranganathan and I. Foster. Identifying Dynamic Replication Strategies for a High Performance Data Grid. In *Proc. of the Int. Grid Computing Workshop*, Denver, Colorado, USA, November 2001.

[11] K. Ranganathan and I. Foster. Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications. In *Int. Symposium of High Performance Distributed Computing*, Edinburgh, Scotland, July 2002.