GRID DATA MANAGEMENT: SIMULATIONS OF LCG 2008

A. T. Doyle, C. Nicholson*
Dept. of Physics and Astronomy, University of Glasgow, Glasgow, G12 8QQ, Scotland

Abstract

Simulations have been performed with the grid simulator OptorSim using the expected analysis patterns from the LHC experiments and a realistic model of the LCG at LHC startup, with thousands of user analysis jobs running at over a hundred grid sites. It is shown, first, that dynamic data replication plays a significant role in the overall analysis throughput in terms of optimising job throughput and reducing network usage; second, that simple file deletion algorithms such as LRU and LFU algorithms are as effective as economic models; third, that site policies which allow all experiments to share resources in a global Grid is more effective in terms of data access time and network usage; and lastly, that dynamic data management applied to user data access patterns where particular files are accessed more often (characterised by a Zipf power law function) lead to much improved performance compared to sequential access.

INTRODUCTION

Particle physicists are currently preparing for the Large Hadron Collider (LHC) at CERN, the European Organization for Nuclear Research, to start data-taking. 2008 will be the first full year of running, and to handle the expected 15 PB/year of raw data, the LHC experiments have adopted grid-based solutions. The LHC Computing Grid (LCG) project has been established to provide and maintain the data storage and analysis infrastructure.

LCG has adopted a tiered architecture, with CERN as a Tier-0 site where all raw data are produced and archived. Tier-1 sites are responsible for a share of permanent data storage and computational power for reprocessing and analysis. Each Tier-1 will have a number of associated Tier-2 sites, each providing computing power. The LHC experiments will use the different tiers in slightly different ways according to their own computing models; while these computing models are well-developed, the actual behaviour of LCG during LHC running remains unknown. Simulation may therefore be a useful tool to examine this. In particular, the data management components may be simulated and ways of improving grid performance investigated. The grid simulator OptorSim [2], originally developed as part of the European DataGrid (EDG) project [1], has been designed to explore the effects of dynamic data replication: replicating files between sites in response to jobs as they run. It has been used to simulate a model of LCG in 2008 and this paper presents some of the results.

OPTORSIM

OptorSim's emphasis is on simulation of the replica management infrastructure, as dynamic data replication involves automated decisions about replica placement and deletion. The architecture and implementation are described in [3] and only a brief description is given here.

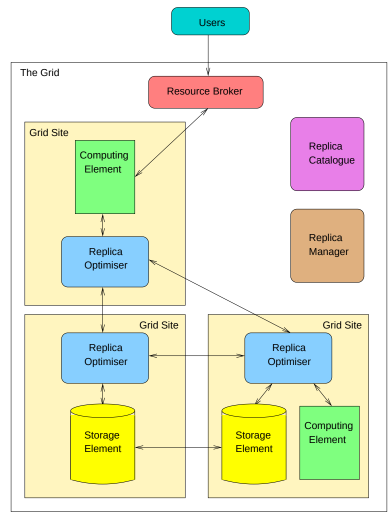

Figure 1: Grid architecture used in OptorSim.

In the OptorSim model (Figure 1), the grid consists of sites connected by network links; a site may have a Computing Element (CE), a Storage Element (SE) or both. Each site also has a Replica Optimiser (RO) which makes decisions on replications to that site. A Resource Broker (RB) schedules jobs to sites. Jobs process files, which can be replicated between sites according to the decisions made by the RO. A Replica Catalogue holds mappings of logical to physical filenames and a Replica Manager handles replications and registers them in the Catalogue.

Grid Topology. To input a grid topology, a user specifies the storage capacity and computing power at each site, and the network links between each. SEs are defined to have a certain capacity, in MB, and CEs to have a certain number of "worker nodes" with a given processing power.

Jobs and Files. A list of jobs and the files that they need is defined; a job will process some or all of the files in its dataset, according to the *access pattern*. The time a file takes to process depends on its size and on the worker nodes at the CE.

Site Policies. Different sites are likely to prioritise different jobs; a site with strong involvement in ATLAS, for example, may prefer to accept ATLAS jobs. In OptorSim, each site is given a list of job types which it will accept.

Optimisation Algorithms. There are two kinds of optimisation algorithm which may be investigated using OptorSim: the job scheduling algorithms used to decide which sites jobs should be sent to, and the data replication algorithms used by the RO at each site. The focus of this paper is on the data replication algorithms, for which there are three options available. Firstly, one can choose to perform no replication. Secondly, one can use a "traditional" algorithm which always tries to replicate files, deleting existing files if necessary. Algorithms in this category are the LRU (Least Recently Used), which deletes those files which have been used least recently, and the LFU (Least Frequently Used), which deletes those which have been used least frequently. Thirdly, one can use an economic model in which sites "buy" and "sell" files using an auction mechanism, and will only delete files if they are less valuable than the new file. Details can be found in [4]. There are currently two versions: the binomial economic model, where file values are predicted by ranking the files in a binomial distribution according to their popularity, and the Zipf economic model, where a Zipf-like distribution is used.

EXPERIMENTAL SETUP

OptorSim was set up using the predicted LCG resources for 2008 as a basis. While some simplifications were necessary for the simulation to run, the aim was to have a simulation which yielded useful information about grid behaviour.

Analysis Jobs and Files. The experiment computing model documents ([5], [6], [7], [8]) describe their analysis models. All experiments except LHCb plan to do most of the analysis at Tier-2 sites, with data storage at Tier-1s. This was modelled by assigning each experiment a dataset, which was placed at each Tier-1 site and at CERN at the start of the simulation. Six job types were defined, and each dataset divided into 2 GB files, with parameters for each job type as presented in Table 1. The jobs processed

Job	Event size (kB)	Total no. of files	Files per job
`alice-pp`	50	25000	25
`alice-hi`	250	12500	125
`atlas`	100	100000	50
`cms`	50	37500	25
`lhcb-small`	75	37500	38
`lhcb-big`	75	37500	375

Table 1: Job configuration parameters used.

a subset of files from the dataset according to the access pattern. Processing times per file were calculated for according to the expected time per event during analysis.

Storage Resources. The Tier-0 and Tier-1 sites were designated as "master sites", with SEs according to their planned capacities in [9], as presented in Table 2. Detailed

Site	Storage (PB)	Experiments served
CERN Tier-0	12.5	All
CAF	6.4	All
TRIUMF	1.5	ATLAS
IN2P3	7.7	All
GridKa	4.0	All
CNAF	7.5	All
NIKHEF/SARA	5.2	ALICE, ATLAS, LHCb
Nordic	2.8	ALICE, ATLAS, CMS
PIC	3.5	ATLAS, CMS, LHCb
ASCC	2.5	ATLAS, CMS
RAL	3.6	All
BNL	5.1	ATLAS
FNAL	5.2	CMS

Table 2: LCG Tier-0 and Tier-1 storage resources for 2008.

resource estimates are not available for all the Tier-2s, so each Tier-2 site was given a canonical value. Averaging the total Tier-2 requirements over the number of sites gave an average SE size of 197 TB. Defining a *storage metric*, D, as the ratio of average SE size to total dataset size allows characterisation of a grid in terms of the proportion of the dataset individual SEs can hold. If $D > 1$, an average SE can hold all the files, so the choice of replication strategy will have no effect. For $D < 1$, the replication strategy becomes more important, but if $D << 1$ due to a very large dataset, replication will begin to lose its advantage, as each job is likely to request new files.

The size of the simulation, however, limited the number of jobs which could be simulated, due to the available memory when running the simulation. The simulations were restricted to about 1000 jobs, so the Tier-2 SE sizes were scaled down to 500 GB, allowing file replacement to start quickly. The disadvantage is that D is then very small, so the file prediction algorithms will not perform to their best advantage. The effect of changing D by changing the size of the dataset, however, is among the tests presented.

Computing Resources. As most analysis jobs run at Tier-2 sites, the Tier-1 sites were not given CEs, except those which run LHCb jobs and were therefore given a CE equal to those at the Tier-2s. The CERN Analysis Facility (CAF) is a special case, and was allocated a CE of 7840 kSI2k. The Tier-2s were given an averaged CE of 645 kSI2k, to meet the total requirement of 61.3 MSI2k.

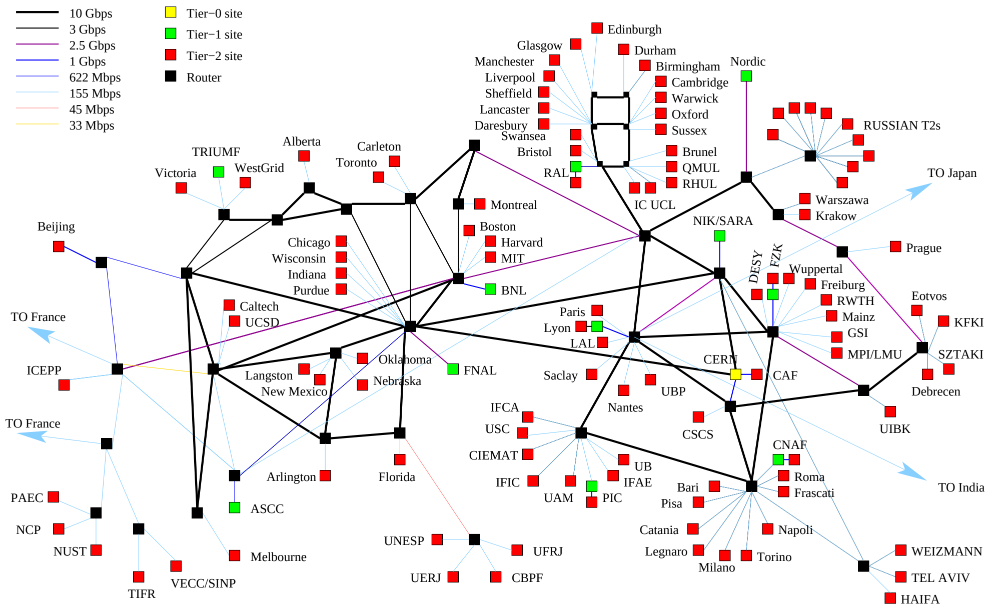

Figure 2: Simulated topology of LCG in 2008.

Network Topology. The network topology (Figure 2) was developed using the topologies of the main research networks, simplified slightly. Sites were connected with the published bandwidths if these were available and 155 Mbps otherwise.

RESULTS

For each of the results presented, a scheduler was used which combines information on data location and lengths of queues at sites. In each test, 1000 jobs were submitted. The metric used in evaluation is the *mean job time*, which is the average time a job takes from scheduling to completion.

Effects of Data Replication The first test presented examines the performance of the replication algorithms with different values of the storage metric, D. The overall dataset size was successively halved, varying D from 1.2×10^{-3} to 7.5×10^{-2} and bringing it closer to the more realistic level of $\mathcal{O}(10^{-1})$. The results of this test are shown in Figure 3. This shows, first, that for low D, dynamic data replication gives little benefit. As D increases, however, replication gives up to 20-25% gain in performance, with the simpler LRU and LFU strategies giving better performance than the economic models. Increasing the number of jobs shows a linear increase in job time, so the relative improvement in performance would hold.

Effects of Site Policies In the last section, site policies were set according to their planned usage. Here, this is compared with two extremes of policy. In the first, called *All Job Types*, all sites accepted all job types. In the second, designated *One Job Type*, each site would accept only one job type, with an even distribution of sites for each job type.

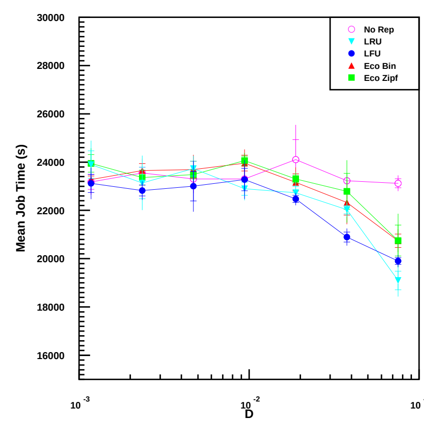

Figure 3: Mean job time with varying D.

The CAF still accepted all job types. The default set of policies is between these two extremes, and is designated in the results below as *Mixed*. The results are shown in Figure 4. These results show that the pattern of site policies on the grid have a powerful effect on performance. The mean job time with the *All Job Types* policy is about 60% lower than with the *One Job Type* policy. This is true across all the replication strategies, although the effect is strongest with no replication and with the LRU. It seems clear that an egalitarian approach, in which resources are shared as much as possible, yields benefits to all grid users.

Effects of Data Access Pattern In the previous sections, file access was sequential. Other access patterns are also possible, and perhaps the most likely of these in a

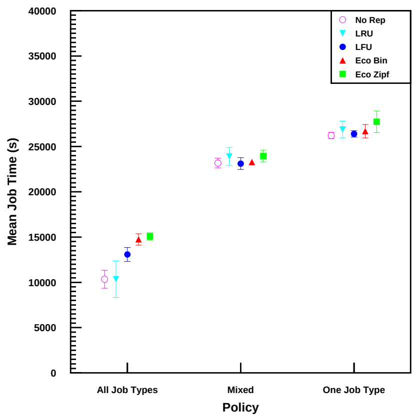

Figure 4: Mean job time for varying site policies.

chaotic analysis situation is a Zipf-like access pattern, with a probability distribution $P_i \propto i^{-\alpha}$, where i is the file rank and α is the power value. [10] showed that in the D0 experiment at FNAL, the least popular files followed a Zipf-like pattern while there were many popular files which were all accessed with the same frequency. This corresponds to the use of the sequential access pattern in OptorSim. This gives strong motivation to examine the relative effects of sequential and Zipf access patterns with OptorSim.

Figure 5 shows the results of using a Zipf-like access pattern with $\alpha = 0.85$. Although the four replication algorithms still have very similar performances, they are now about 75% faster than without replication. This is due to the way in which a few files from each job's fileset are accessed many times during the jobs, while others are accessed infrequently. This allows the access histories to predict file values more accurately than with the sequential pattern, where they may see a file only once. The presence of any Zipf-like element, even if combined with a sequential pattern, would make dynamic replication highly desirable.

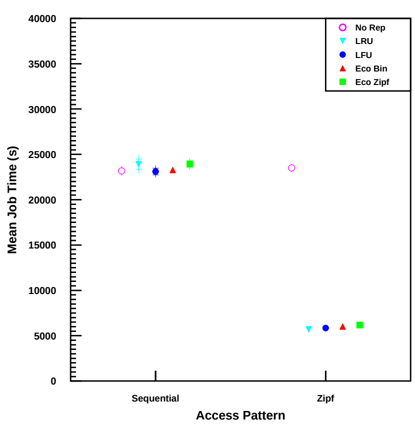

Figure 5: Mean job for varying access patterns.

CONCLUSIONS

The grid simulator OptorSim has been used to simulate a model of LCG during the first year of LHC running, exploring several different aspects. First, it was shown that dynamically replicating data between sites using a sequential file access pattern decreased the running time of grid jobs by about 20%, especially as sites' Replica Optimisers gained more knowledge of the overall dataset. While the performances of different replication strategies were similar, the simpler LRU and LFU strategies were found to perform up to 20% and 30% better, respectively, than the economic models. Examining site policies, it was found that a policy which allowed all experiments to share resources on all sites was most effective in reducing data access time. Finally, it was shown that if user data access patterns include a Zipf-like element, dynamic replication has a much stronger effect than with sequential access, with gains in performance of about 75%. It is quite likely that an analysis situation would involve Zipf-like elements in data access patterns, and so implementing such an automated file replication and deletion tool for analysis would give significant gains. In future, if sub-file level replication were implemented, these gains could be even greater.

ACKNOWLEDGMENTS

This work was funded by PPARC. Thanks to all the members of the EDG WP2 Optimisation Team, whose work allowed this research to be conducted.

REFERENCES

[1] The European DataGrid Project, http://www.edg.org

[2] OptorSim Release 2.0, November 2004. `http://edg-wp2.web.cern.ch/edg-wp2/optimization/optorsim.html`.

[3] D. Cameron et al, "Evaluating Scheduling and Replica Optimisation Strategies in OptorSim", Journal of Grid Computing 2(1):57-69, March 2004.

[4] W. Bell et al, " Evaluation of an Economy-Based Replication Strategy for a Data Grid", Int. Workshop on Agent Based Cluster and Grid Computing, Tokyo, 2003

[5] ALICE Computing Model. Technical Report CERN-LHCC-2004-038/G-086, CERN, January 2005.

[6] The ATLAS Computing Model. Technical Report CERN-LHCC-2004-037/G-085, CERN, January 2005.

[7] The CMS Computing Model. Technical Report CERN-LHCC-2004-035/G-083, CERN, January 2005.

[8] LHCb Computing Model. Technical Report CERN-LHCC-2004-036/G-084, CERN, January 2005.

[9] LHC Computing Grid Technical Design Report. Technical Report CERN-LHCC-2005-024, CERN, June 2005.

[10] A. Iamnitchi and M. Ripeanu. Myth and reality: Usage behavior in a large data-intensive physics project. Technical Report TR2003-4, GriPhyN, 2003.