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

The particle physics community is currently preparing for the Large Hadron Collider (LHC) at CERN, the European Organization for Nuclear Research, to start data-taking in 2007. 2008 will be the first full year of data-taking, and to handle the expected 15 PB/year of raw data, plus secondary data produced in reconstruction, analysis and simulation, 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 four-tiered grid architecture. CERN is a central Tier-0 site where all raw data are produced and archived. First-pass reconstruction will also take place there. Tier-1 sites are responsible for permanent storage of the data which they have been allocated, and providing computational power for reprocessing and analysis. Each Tier-1 will have a number of associated Tier-2 sites, each providing computing power for analysis and serving some geographical area. The Tier-3 layer then consists of the computing facilities at universities and other LHC-related institutions, which will be used for processing and analysis but are not directly part of the LCG project. The LHC experiments will also use the different tiers in slightly different ways according to their own computing models.

While these computing models are already well-developed, the actual behaviour of LCG during LHC running necessarily remains unknown. Simulation may therefore be a useful tool to investigate system behaviour. In particular, the data management components of LCG 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, has been designed for such simulations and especially 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; this paper presents the results of these simulations. First, a brief description of OptorSim is given, followed by the experimental setup. The results of experiments investigating different data replication algorithms, site policies and data access patterns are then given before drawing some conclusions.

OPTORSIM

OptorSim is an event-driven simulator, written in Java. As dynamic data replication involves automated decisions about replica placement and deletion, the emphasis is on simulation of the replica management infrastructure. The architecture and implementation are described in [3] and so only a brief description is given here.

Architecture

The conceptual model of the OptorSim architecture is shown in Figure 1. In this model, the grid consists of a number of sites, connected by network links. A grid 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) handles the scheduling of jobs to sites, where they run on the CEs. Jobs process files, which are stored in the SEs and can be replicated between sites according to the decisions made by the RO. A Replica Catalogue holds mappings of logical filenames to physical filenames and a Replica Manager handles replications and registers them in the Catalogue.

Simulation Inputs

Grid Topology. To input a grid topology, a user specifies the storage capacity and computing power at each site, and the capacity and layout of 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. Sites which have neither a CE nor an SE act as routers on the network. Background traffic on the network can also be simulated.

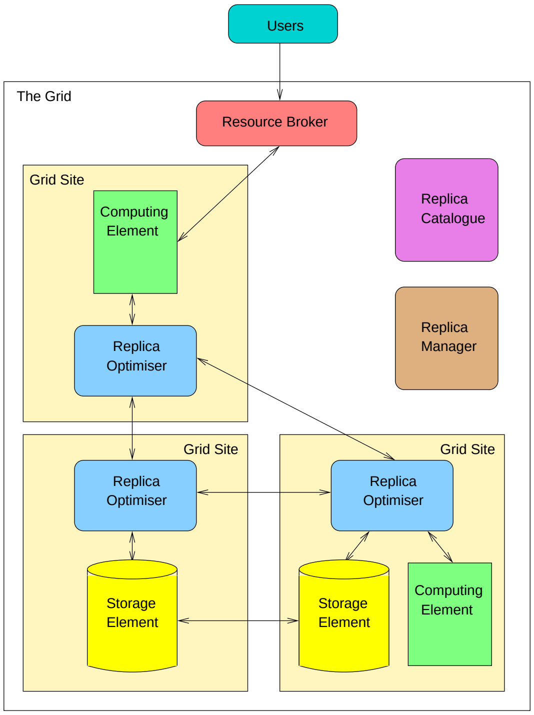

Figure 1: Grid architecture used in OptorSim.

Jobs and Files. A physics analysis job usually processes a certain number of files. This is simulated by defining a list of jobs and the files that they need; a job will process some or all of the files in its dataset, according to the *access pattern* which has been chosen. The time a file takes to process depends on its size and on the number and processing power of worker nodes at the CE. It is assumed that the output files from a physics analysis would be small enough to ignore compared to the input datasets, and also that these are likely to be stored at local sites rather than on the grid, and so no simulation of output files is required.

Site Policies. Different grid sites are likely to prioritise different kinds of job. A university with strong involvement in the ATLAS collaboration, for example, may prefer to accept ATLAS jobs, whereas a regional Tier 2 centre may be contracted to serve all experiments. 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 by the RB to decide which sites jobs should be sent to, and the data replication algorithms used by the RO at each site to decide when and how to replicate files. The focus of this paper is on the data replication algorithms, and so the job scheduling algorithms are not described here.

There are three broad options for replication strategies in OptorSim. Firstly, one can choose to perform no replication. Secondly, one can use a “traditional” algorithm which, when presented with a file request, always tries to replicate and, if necessary, deletes existing files to do so. 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 in the recent past. 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 of the auction mechanism and file value prediction algorithms can be found in [4]. There are currently two versions of the economic model: the binomial economic model, where file values are predicted by ranking the files in a binomial distribution according to their popularity in the recent past, and the Zipf economic model, where a Zipf-like distribution is used instead (a Zipf distribution is one in which a few events occur very frequently, while most events occur infrequently).

Evaluation Metrics

For evaluating grid performance, different users may have different criteria. An ordinary user will most likely be interested in the time a job takes to complete. The owners of grid resources, on the other hand, will want to see their resources being used efficiently. The evaluation metrics used in this paper are *mean job time*, which is the average time a job takes to run, from the time of scheduling to completion, and *effective network usage (ENU)*, which is the ratio of file requests which use network resources to the total number number of file requests.

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 LHC experiment computing model documents ([5], [6], [7], [8]) describe the planned analysis models and the roles played by different tiers. All the experiments plan to do most of the analysis at Tier-2 sites (except LHCb, which plans to use Tier-1s), with primary 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. Assuming that a “typical” analysis job runs over 10^6 events (with the `lhcb-big` jobs running over 10^7 events) and taking the AOD event sizes from the computing models gave the parameters for each job type as presented in Table 1. The total size of the dataset for each job type is the approximate size of a single copy of the AOD for a year’s worth of data taking.The simulated jobs processed a subset of files

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 in the LCG 2008 configuration.

from the dataset, from a random starting point, according to the access pattern. When a job ran on a site, it retrieved its files and processed them according to the computing resources available at that site. Processing times per file were calculated for each job according to the expected processing time per event during analysis, and the probability of a particular job being run on the grid was modelled by the relative number of expected users for the different experiments, taken from the computing model documents.

Site Resources

The resource requirements for LCG in the first few years of LHC data-taking were drawn from [9], and the Tier-1 and Tier-2 sites participating in LCG in 2008 were taken from [10]. Using this, the analysis model was used to allocate appropriate resources to each site as follows.

Storage Resources. The Tier-0 (CERN) and Tier-1 sites were designated as “master sites”, and were given SEs according to their planned capacities, as presented in Table 2. As OptorSim does not differentiate between types of storage, the tape and disk capacities were summed to give a total capacity for each site. Table 2 also shows the experiments served by each site.

Detailed resource estimates are not available for all the Tier-2s, and so each Tier-2 site was given a canonical value. Averaging the total Tier-2 resource requirements over the number of Tier-2 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. The size of D indicates the likelihood of replication occurring. If $D > 1$, an average SE has more than enough capacity to hold all the files, so the choice of replication strategy will have little effect. For $D < 1$, the replication strategy becomes more important, as the SE is not capable of holding all the files, but if $D << 1$ due to a very large dataset, replication will begin to lose its advantage, as each job is likely to request files which have not been requested before. An SE size of 197 TB gives a value for D of 0.47.

The number of sites in the simulation, however, limited the number of jobs which could be simulated, due to the physical limitation of available memory when running the simulation. This meant that the simulations were restricted to the order of 1000 jobs, and so the Tier-2 SE sizes were scaled down to 500 GB. These then hold 250 files, allowing file replacement to start when at most 10 jobs have been submitted to a site. This has the disadvantage that the storage metric 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.

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.

Computing Resources. As most analysis jobs run at Tier-2 sites, the Tier-1 sites in the simulation were not given CEs, except those which run LHCb jobs and were therefore given a CE equal to those at the Tier-2s. In reality, of course, the Tier-1s have large computing resources, but as the focus here is on analysis, they are assumed to be reserved for reconstruction and thus unavailable for analysis. The CERN Analysis Facility (CAF) is a special case, and was allocated a CE of 7840 kSI2k as well. The Tier-2s were given an averaged CE compute power of 645 kSI2k, to meet the total requirement of 61.3 MSI2k over the 95 simulated Tier-2 sites.

Network Topology

The network topology between the sites was developed using the published topologies of the main research networks, simplified slightly to give the network backbone. Sites were connected to their closest router node, with the published bandwidths used if these were available and a default of 155 Mbps otherwise. As the simulation was geared towards the user analysis view of the grid, where resources are available via the standard research networks rather than the dedicated paths which will be available for initially transporting data from CERN to Tier-1 sites, this is not inappropriate. Sites with both a Tier-1 and a Tier-2 facility had the Tier-2 attached directly to the Tier-1 by a 1 Gbps link. The resulting topology is shown in Figure 2.

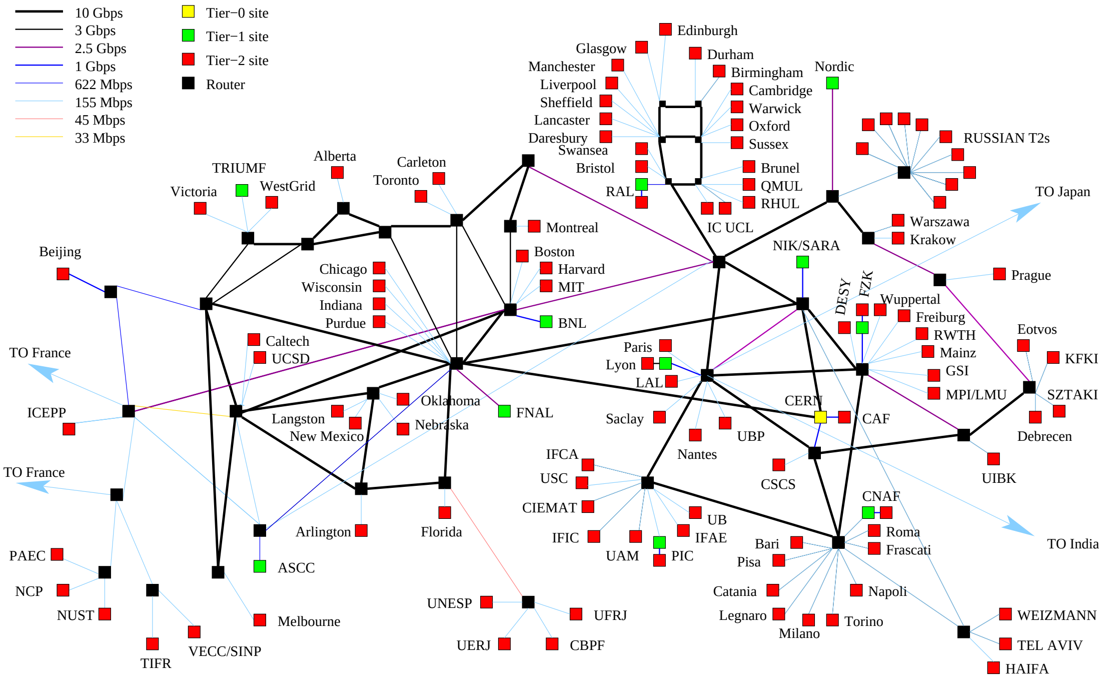

Figure 2: Simulated topology of the LCG 2008 grid. CERN, as the Tier-0 site, is shown in yellow, while Tier-1 sites are green, Tier-2s are red and router nodes are black. Network links have the values shown in the key.

EFFECTS OF DATA REPLICATION

For each of the results presented in this and the following sections, a job scheduler was used which combines information on data location and lengths of queues at sites; this has been shown in previous studies to give the best grid usage [3]. In each test, 1000 jobs were submitted to the grid, and the test repeated 3 or more times to give an average.

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, thus increasing the fraction which could be stored by a Tier-2 site. D was varied from 1.2×10^{-3} to 7.5×10^{-2}, 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. There is also improvement in the network usage as D increases, as more files are available locally.

Although these results were gained with 1000 grid jobs, Figure 4 shows the variation in job time with an increasing number of jobs. This shows a linear increase in job time with number of jobs. Extrapolating to $\mathcal{O}(10000)$ jobs, which is a more realistic number, this linear relationship is expected to hold. This means that with realistic values of D, and higher numbers of jobs, the relative improvement in performance would hold. Replication is therefore an important way of reducing job times and network usage, and the relatively simple LRU and LFU strategies are the most effective.

EFFECTS OF SITE POLICIES

In the last section, site policies were set according to their planned usage. Here, the effect of site policies on the overall running of the grid are investigated. This was done by defining 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. The CAF, being a special case, still accepted all job types. The default set of site policies is therefore in between these two extremes, and is designated in the results below as *Mixed*. The results are shown in Figure 5.

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. *All Job Types* also gives a lower ENU (about 25% lower than the others) 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, jobs accessed their files sequentially. Other access patterns are also possible, however, and perhaps the most likely of these in a chaotic analysis situa-

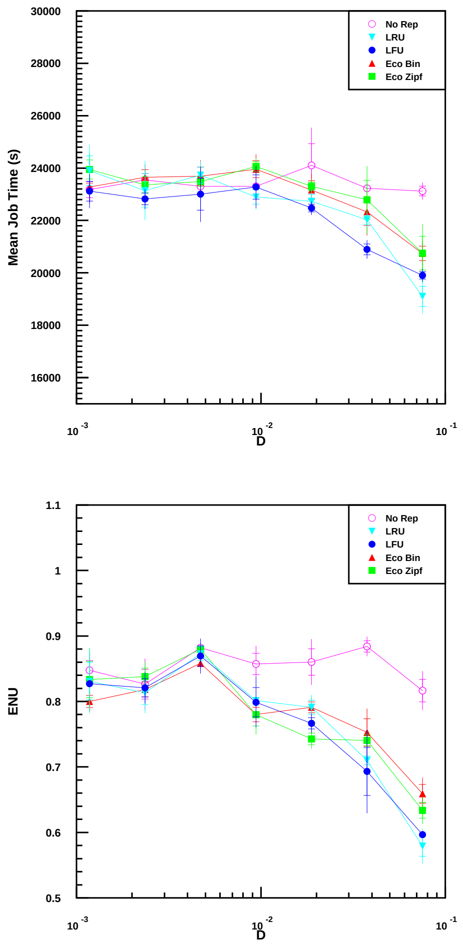

Figure 3: Mean job time (top) and ENU for replication strategies, varying D.

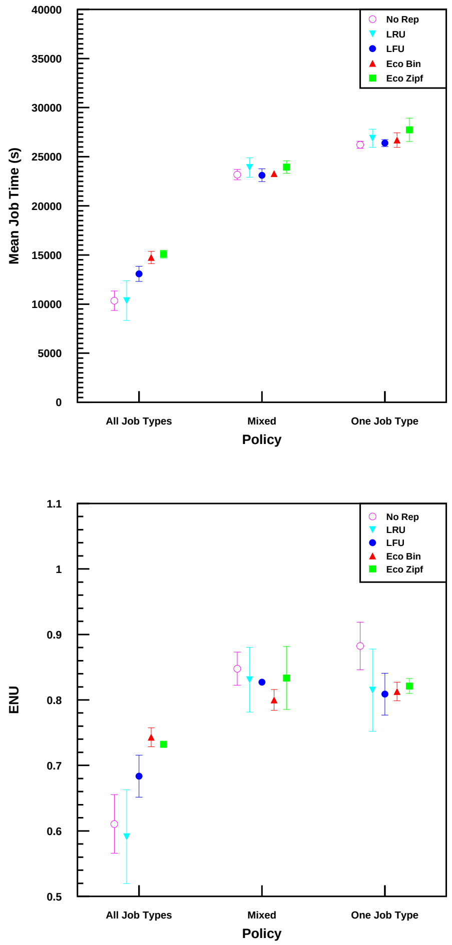

Figure 5: Mean job time (top) and ENU for replication algorithms, with different site policies.

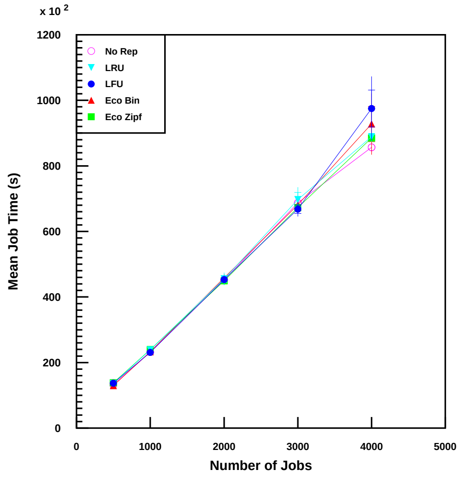

Figure 4: Mean job time with varying number of jobs.

tion is a Zipf-like access pattern. Such a distribution has been observed for web page access patterns. A particle physics example could be files containing data from a set of possible Higgs events, which would attract a great deal of attention from LHC physicists.

In [11], examination of access patterns for the D0 experiment at FNAL showed that although the least popular files followed a Zipf-like pattern, there were a large number of popular files which were all accessed with the same frequency, which corresponds to the use of the sequential access pattern in OptorSim. This observation may be specific to the D0 sample studied, or may be applicable to HEP experiments in general, but gives strong motivation to examine the relative effects of sequential and Zipf access patterns with OptorSim.

Figure 6 shows the results of using a Zipf-like access pattern rather than a sequential access pattern. This is quite different from the previous results. Although the four replication algorithms still have very similar performances, they are now about 75% faster than without replication. The

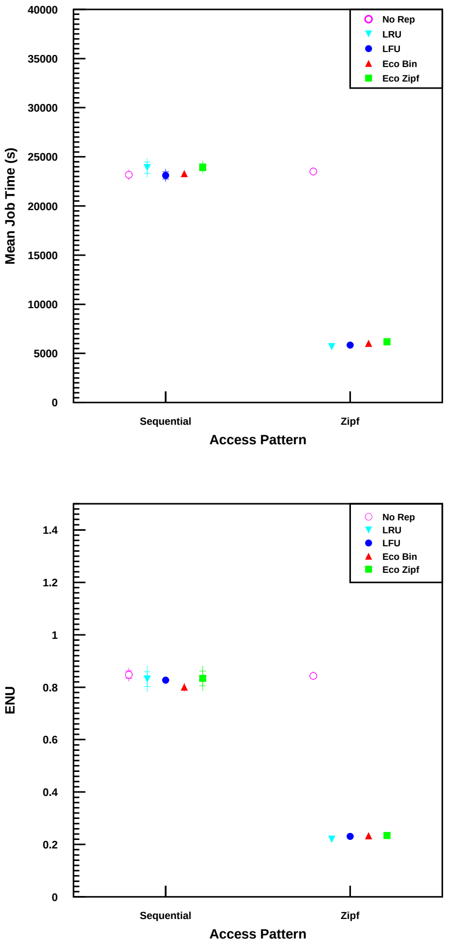

Figure 6: Mean job time (top) and ENU for replication algorithms, with Zipf-like access pattern.

ENU is correspondingly lower. 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. As the number of jobs and the proportion of the whole dataset seen by an individual SE increases, however, the results with sequential access should tend towards a similar pattern as for the Zipf access. This is borne out by the results from varying D with sequential access. The presence of any Zipf-like element, even if combined with a sequential pattern, would make dynamic replication highly desirable.

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% and reduced usage of the network by about 25%, 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 and network usage. Finally, it was shown that if user data access patterns include a Zipf-like element, with some files much more popular than others, 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/downloads/v2_0/`.

[3] D. Cameron, R. Carvajal-Schiaffino, P. Millar, C. Nicholson, K. Stockinger and F. Zini, "Evaluating Scheduling and Replica Optimisation Strategies in OptorSim", Journal of Grid Computing 2(1):57-69, March 2004.

[4] W. Bell, D. Cameron, R. Carvajal-Schiaffino, P. Millar, K. Stockinger and F. Zini "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] Memorandum of Understanding for Collaboration in the Deployment and Exploitation of the Worldwide LHC Computing Grid. Technical Report CERN-C-RRB-2005-01, CERN, September 2005.

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