Performance Analysis of DataGrid Systems for High Energy Physics Applications

Atsuko Takefusa,[†1] Golnaz Bahmanyar,[†2] Osamu Tatebe[†3] and Satoshi Matsuoka[†2,†4]

DataGrid is a Grid environment for petascale data-intensive computing. Being in the development stage, performance of DataGrid systems on large-scale and realistic applications have not been well-investigated. Using the Bricks Grid simulator, we investigate the performance of various DataGrid system models in simulation by assuming actual application scenarios.

1. Introduction

Next generation scientific exploration and research in areas such as High Energy Physics (HEP), astronomy, bioinformatics require analysis of large-scale data reaching hundreds of T-Pbytes. One example is the Large Hadron Collider (LHC) at CERN, which is a particle accelerator that will produce an order of petabyte of raw data each year, starting in 2006.

"DataGrid" is a Grid environment for petabyte-scale data-intensive computing. The aim of DataGrid systems is to establish a fleet of computational and data-intensive Grid resources for the analysis of data derived from scientific experiments. To process such large amounts of data, a global-scale Grid computing model consisting of multi-tier worldwide Regional Centers has been studied by the MONARC project[3].

Being in the development stage, performance of DataGrid systems on large-scale, realistic applications has not been well-investigated. We simulate the performance of various DataGrid system models under various application scenarios by using the Bricks Grid simulator with DataGrid extensions, comparing centralized data storage and processing vs. Monarc-style hierarchical distributed configuration, etc.

2. DataGrid Projects

DataGrid is a framework to process/manage T∼PB-scale data as well as ensuring data management, job allocation, replication, etc.

Grid Datafarm (Gfarm)[1] provides a global data parallel file system with online petascale storage, scalable I/O bandwidth and scalable parallel processing for DataGrid applications, exploiting local I/O in a grid of clusters with tens of thousands of nodes. In order to attain scalability, Gfarm basically adopts the owner-computes strategy rather than the other way to converse of staging the data over to the computation using high-performance file systems such as HPSS.

GriPhyN[2], EU DataGrid[5], and PPDG[6] are other representative DataGrid projects. In the GriPhyN project, various replication/caching policies have been simulated and analyzed using different access patterns[7]. Their results claim that the fast spread policy saved network bandwidth and the cascading policy yielded faster response for "loading data". Although such low-level findings are important, their simulation was not conducted assuming reasonable application scalability. In our simulation we attempt to simulate scalability of actual application scenarios.

3. Simulation Modeling and Setting

3.1 Job Processing

In LHC experiments, observed data (*events*) are collected from a huge number of collisions of particles and analyzed through different levels of a data processing hierarchy[3]. A typical *job* in the experiments is a collection of millions of the events. A job is handled on a DataGrid system as follows:

(1) A user (physicist) invokes a job
(2) The scheduler selects suitable servers
(3) Each server loads the data fragment required for the job
(4) The servers process parts of the job
(5) The servers send the output to specified storages (Client receives only statistical data)

The duration it takes to process this job is shown as follows:

$$T_{response} = T_{read} + T_{process} + T_{write} \quad (1)$$

†1 Ochanomizu University
†2 Tokyo Institute of Technology
†3 National Institute of AIST
†4 National Institute of Informatics

Table 1 Parameters set for simulation.

	Storage	Performance	Data	Nodes
Case1	7.6PB	12M SI95	100TBx10, 10TBx100	10000
Case2	5.7PB	9M SI95	100TBx10, 10TBx100	10000
Case3	Tier-1: 2PB	3.157M SI95	100TBx10	10000
	Tier-2: 1PBx4	1.578M SI95	(10TBx25)x4	5000
	Tier-3: 100TBx16	0.157M SI95		500

3.2 DataGrid Architectures

The MONARC project proposed the multi-tier regional center model, due to the limitation of computational and storage resources. The report assumed that could be placed at a single site at the time of its publication However, remarkable improvement in commodity technologies could allow huge clusters with petascale storage with appropriate data processing capacity. The first plausible comparison would be thus to investigate how much performance penalty we could suffer by distributed placement of storage and computational resources in the Monarch style "tier model".

We have assumed two models for our simulation. One is the *Central model* where all the jobs are processed at single site and the other is the *Tier model* where jobs are processed in different levels of the hierarchy. The advantages of the former are manageability and performance in that all of the jobs can be handled on the site; however there are limitations of cost, electric power, and aggregation performance that is achievable on a single site. For the latter, DataGrid systems must facilitate suitable scheduling and replication policies to deploy user jobs and maintain data replicas over the resources for efficient data processing.

3.3 Simulation Scenario

We have determined the following scenarios.

Case1 Data is processed at the central site where there is sufficient processing power to handle all jobs. The queue of jobs at the site, according to the queuing theory is stable (arrival rate $\ll$ service rate).

Case2 Data is processed at the central site where processing power is limited which will cause the server to be overloaded.

Case3 When the server load increases, a copy of data is created at a lower level tier and job processing is delegated to that tier.

Case1, 2 correspond to the *Central model* and Case 3 corresponds to the *Tier model*.

Parameters used in simulation are listed in Table 1. We have analyzed and typified the jobs running different levels of analysis in actual HEP experiments into two kinds:

- 100TB $\rightarrow$ 10TB (250G SI95*S) once a day
- 10TB $\rightarrow$ 1TB (25G SI95*S) 10 times a day

Table 2 Comparison of response time in Case1 and 2.

	Case1 [sec]	Case2 [sec]
T_{read}	18.168	18.131
$T_{process}$	1871.202	2725.104
T_{write}	1.814	1.810
$T_{response}$	1891.184	2745.046

3.4 Simulation with Bricks

We utilized the Bricks Grid simulation framework for evaluation purposes. Bricks is a discrete event simulator written in Java and provides canonical Grid scheduling modules and various scheduling analysis under dynamic Grid environment. For this work we have determined the following policies for user job scheduling onto compute resources. policies:

- Owner Computes Rules in Case1,2
- Greedy strategy in Case3 where it allocates a job into the server which is estimate to process the job fastest.

4. Evaluation

We have analyzed the performance of DataGrid systems when using different system architectures and scheduling algorithms; the details of the simulation results will be shown on the poster. Performance comparison of response time in Case1 and 2 are shown in Table 2;

5. Summary and Future Work

We modeled an actual HEP application scenario with different system architectures and compared performance using the Bricks system. More simulation scenarios with various scheduling policies will be performed by the time of poster presentation.

References

1) http://datafarm.apgrid.org/.
2) http://www.griphyn.org/.
3) M. Aderholz et al.: Models of Networked Analysis at Regional Centres for LHC Experiments. Monarc Phase 2 Report. (2000).
4) http://grid-team.is.titech.ac.jp/bricks/.
5) http://www.eu-datagrid.org/.
6) http://www.ppdg.net/.
7) Ranganathan, K. and Foster, I.: Identifying Dynamic Replication Strategies for a High Performance Data Grid, *Proc. of Grid Computing* (2001).