Grid Data Farm for Atlas Simulation Data Challenges

Y.Morita[1], O.Tatebe[2], S.Matsuoka[3], N.Soda[4], H.Sato[1], Y.Tanaka[2],
S.Sekiguchi[2], S.Kawabata[1], Y.Watase[1], M.Imori[5], T.Kobayashi[5]

[1]KEK, Tsukuba, Japan, [2]AIST, Tsukuba, Japan
[3]Tokyo Institute of Technology, Tokyo, Japan
[4]SRA Inc., Tokyo, Japan, [5]University of Tokyo/ICEPP, Tokyo, Japan

Abstract

Data-intensive computing and networking technology has become a vital part of large-scale scientific research projects such as LHC/Atlas. Handling of petascale data with PC clusters with thousands of nodes will impose a non-trivial challenge on effective utilization of the CPU resources and the I/O bandwidth.

Grid Data Farm is a middleware project to solve the petascale data-intentive data analysis. It is based on Grid-based RPC, in particular an extended variant of Ninf system, and also on other lower level Grid services such as Globus.

Atlas will perform a series of world-wide Data Challenges as a part of constructing and validating its software and hardware infrastructure. We have studied the feasibility and the scalability of the Grid Data Farm middleware by using Atlas simulation framework FADS/Goofy. Parallel I/O capability of the Gfarm architecture is demonstrated.

Keywords: PC Cluster, PC Farm, Grid, Data Challenge, Gfarm

1. Introduction

In the Atlas and other three exiperiments in the Large Hadron Collider (LHC) project at CERN, an order of Petabyte of raw data will be produced each year, from 2006. Distributed nature of the participating researchers around the world requires a distributed and reliable computing model for the data analysis [1].

KEK (High Energy Accelerator Research Organization) and ICEPP (International Center for Particle Physics, the University of Tokyo) will jointly build a so called "Tier-1" regional center for the Atlas in Japan. A large cluster of PC farm with several hundreds or a thousand of CPUs, as well as an order of petabytes of mass storage system will be employed in the regional center. This facility must cooperate with other participating laboratories and universities world-wide with different use-policy and system management disciplines. Grid technologies, especially the Globus package is expected to play a major role in building the world-wide distributed LHC data analysis system [2,3].

To build a computer system which is scalable in the data intensive computing of thousands of CPU node, which is also be able to to a part of the world-wide distributed

computing resource of the experiment, we have developed a middleware project "Grid Data Farm (Gfarm)".

2. Grid Data Farm

Major components of the Grid Data Farm are, Gfarm client, Gfarm server, and Gfarm file system (Fig. 1). The Gfarm file system consists of Gfarm Meta Database and Gfarm pool. Gfarm pool may consist of a thousands node PC cluster, each node may be equipped with local disks and the files may be duplicated and distributed using Gfarm and other Grid middleware.

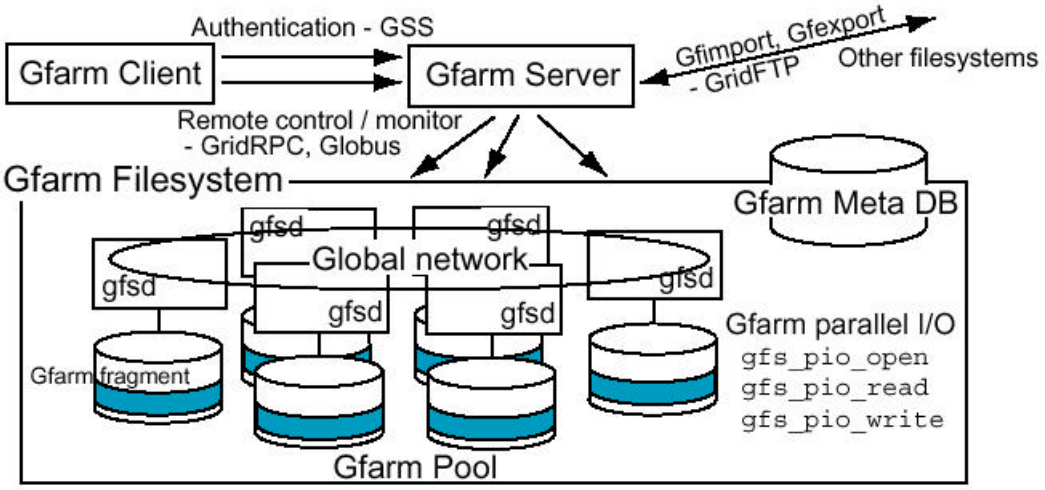

Fig. 1 : Software architecture of the Grid Data Farm

A large-scale distributed file, called Gfarm file, is divided into several fragments and distributed across the disks in the Gfarm filesystem. A Gfarm file, specified by the Gfarm file name or the Gfarm URL such as **gfarm:/path/name**, is a logical aggregation of physical file fragments distributed across multiple CPU nodes. A job accessing a Gfarm file uses the Gfarm parallel I/O library, and the job is processed at each node in parallel where the physical file fragments reside. The Gfarm filesystem daemon (gfsd) runs on each node to facilitate remote file operations with access control using a light-weight GFS RPC. Metadata of the files such as the mapping between the logical Gfarm file name and the physical path name of the file fragments is stored into the Gfarm Meta Database. Flexible configuration of physical file fragments and the CPU and disk resources allows efficient parallel I/O and dynamic load balancing of multiple jobs.

3. Gfarm with Atlas Detector Simulation

FADS/Goofy, is a software framework of Geant4-based full detector simulation written in C++ [6]. We utilize this simulation framework to study the scalability issues and functional requirements of the Gfarm. An abstract interface for the data I/O persistency mechanism is provided in FADS/Goofy so that users can switch from one persistency package to another. The simulated raw data will then stored into files or databases and

will be read by the analysis framework later.

This approach imposes a limit in utilization of object associations across multiple database federations in object databases such as Objectivity/DB. However, by limiting the object associations locally at the given data sample, one can process large amount of multiple data sample independently on each node, from the reconstruction of the raw data into event summary data (ESD), or into analysis object data (AOD) (Fig. 2).

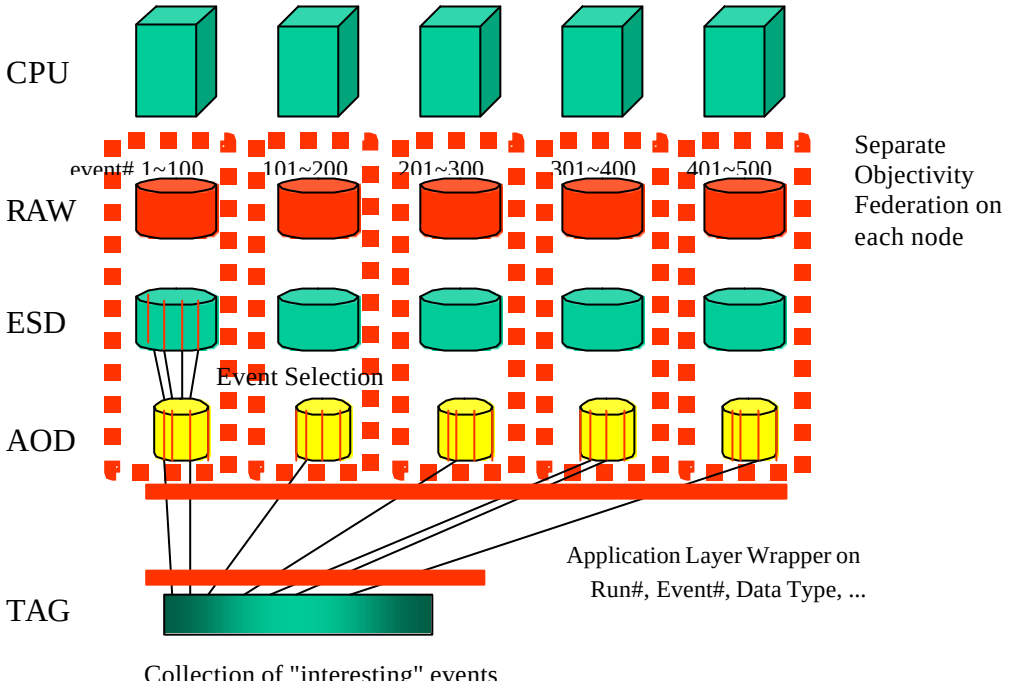

Fig. 2 : Event Database Federation mapped on the Gfarm file system

At the end of analysis cycle, users will create a "TAG", a collection of "interesting" event of their analysis, which contains the n-tuple physics quantities of the events and the logical link to the analysis objects in the chain of the analysis package from calibration, reconstruction and filtering.

Each database file in the same category is treated as a "file fragment" in the Gfarm metadatabase, and aggregated as a single logical Gfarm "filename" as the set of database. When a job is submitted into the Gfarm server, job is redistributed to the nodes which contain the "fragment" database files. Event is iterated over within each fragment database on each node. For a tag of the collection of "interesting" events across multiple node, a thin wrapper layer in the FADS/Goofy application will locate the physical path of the associated object and will returns the data member value across the network.

Since the data I/O is confined within each node in typical use case, I/O speed of the Objectivity/DB transaction is identical to the standalone database operation in a single node. Scalability issue arises when a common set of files, such as calibration data, job executable files, and collecting or re-clustering the file fragments for load balancing reconfiguration and in importing and exporting files from/to external file system. Parallel processing of file replication and relocation helps the speed up of the process.

Parallel network transfer is required in high-bandwidth, high-latency wide area networks, which is suitable for the Gfarm parallel architecture.

4. Summary and Futer Plans

A data parallel job processing middleware Gfarm is developed. A preliminary study indicates that the distributed jobs run at the same performance with a single job running standalone on a local CPU and disks. Actual scalability is expected to depend on distributing job execution files, parameters, common calibration data, and handling of the common background events in case of the Monte Carlo simulation study. Scalability test with the Atlas Full Detector Simulation based on FADS/Goofy and Geant4 up to several hundred nodes are planed by the end of this year as a part of Atlas Data Challenges.

Although Gfarm is designed to work on flat structure file for nominal I/O use case, it has been demonstrated that more complicated I/O packages such as Objectivity/DB can be fitted into the Gfarm file system seemlessly by posing a simple "local-association only" rule into the object modelling. Performance study of thin-layer wrapper to access any object accross the Gfarm pool nodes, which will provide a transparent and parallel seek of the object in each node, are being investigated and will be published in near future.

Refrences

[1] MONARC Collaboration, Models of Networked Analysis at Regional Centres for LHC experiments: Phase 2 report, Technical Report CERN/LCB-001, 2000. http://www.cern.ch/MONARC/.

[2] Ian Foster and Carl Kesselman, Globus: A metacomputing infrastructure toolkit. Intl J. Supercomputer Applications 11(2):115 - 128, 1997.

[3] EU DataGrid, http://www.eu-datagrid.org/
PPDG: Particle Physics Data Grid, http://www.ppdg.net/
GriPhyN: Grid Physics Network, http://www.griphyn.org/

[4] O. Tatebe *et al.*, Grid Data Farm for Petascale Data Intensive Computing, Electrotechnical Laboratory, Techinical Report, TR-2001-4.
http://datafarm.apgrid.org/.

[5] T. Saeki and Y. Morita, HepMC_Contrib: Persistent interface package for HepMC, submitted to this conference.

[6] A.DellAcqua *et al.*, Development of the ATLAS Simulation Framework, submitted to this conference.