CDF/DOC/COMP_UPG/PUBLIC/7900
Draft 0.0
October 24, 2005

Evaluation of the Monte Carlo Data Processing Using JIM for CDF
Intermediate Project Report

Valeria Bartsch[1]
FNAL/UCL, USA/UK

Gabriele Garzoglio[2]
FNAL, USA

Anoop Rajendra[3]
Texas Arlington, USA

Abstract

The SAM-Grid is a distributed meta-computing infrastructure tailored for high-energy physics applications. The system is composed of two main components: JIM, the Job and Information Management services, and SAM, the Data Handling services.

Starting in January 2005, the CDF collaboration and the SAM-Grid team have developed and deployed a SAM-Grid test bed tailored for Monte Carlo (MC) production for CDF. The goal of the project was evaluating whether the SAM-Grid data processing model for generating MC events would fulfill the CDF needs. The prototype was made available at the end of May and maintained functional for about 4 months. During this period, because of other higher priorities in the collaboration, only a partial evaluation of the system was produced.

This document describes the principal design and deployment choices and the feedback on the system by the CDF user community.

[1]bartsch@fnal.gov
[2]garzoglio@fnal.gov
[3]anoopr@fnal.gov

Contents

1 Introduction

Adopting Grid technologies, high-energy physics experiments are able to decentralize their computing resources to remote collaborating institutions. At the beginning of 2005, the CDF collaboration was in the process of evaluating how different grid technologies would address the CDF computing needs. Two such technologies were the "Glidein-CAF" and the JIM component of the SAM-Grid system.

JIM was an attractive technological choice for CDF for the following reasons:

- It is fully integrated with the SAM system, the data handling system of the experiment.
- It provides a user-friendly/easy-to-user user interface.
- It provides a lightweight client, easy to install.
- It uses standard grid middleware services and protocols for job and data management.
- It interfaces to non-dedicated clusters by deploying JIM-specific services on a single gateway node at the cluster; this leads to a low-cost increase of the available computing power to the experiment.
- The DZero experiment at Fermilab relies on the SAM-Grid for its production activities, in particular for data reprocessing and MC generation.

Because of these reasons, CDF and the SAM-Grid team have started a project of integration of the CDF MC production application with the JIM framework. Goal of the project was providing a system prototype that could be used to evaluate whether the data processing model of JIM could fit the CDF needs for MC production.

In the next section, we review the data processing model that was implemented within the JIM framework to produce MC events for CDF. In section 3 we describe the deployment of the test bed system. Such system has been maintained for four months, before cluster reconfigurations made it non-functional. In section 4 we comment on the phase of evaluation of the system. We conclude in section 5.

2 Design and Implementation

The SAM-Grid job management component is based on a multi-tiered architecture. Users interact with the grid through a thin client interface (first tier). The client interface interacts with the "submission" service (second tier), which is responsible for maintaining the queue of grid jobs and for scheduling them to the remote clusters (third tier). The submission service interacts with a resource selection service, which recommends what cluster should receive each job. This infrastructure is implemented around Condor-G services.

Jobs enter the remote cluster through a gateway service. This service is responsible for authenticating/authorizing users and for interfacing the jobs to the fabric services. Fabric services include the local job scheduler, the SAM data handling system, the local monitoring infrastructure, and the environment preparation service.

The implementation of the grid-to-fabric interface is different for the different types of applications supported by the SAM-Grid. Introducing a new application type consists in developing a new module for the grid-to-fabric interface.

The design decisions of the module for the CDF montecarlo application (CDF MC) are discussed hereby. The data processing model implemented for CDF MC is the result of these decisions.

- The CDF MC module has been implemented using the DZero MC module as reference. The DZero module and the relative data processing model, in fact, has been proven sufficiently robust to be used in production. The model relies on a central request database. Users request the production of MC events describing them with MC-specific metadata. Each new request is granted a unique identification number, used when submitting jobs to the grid. The request system allows the central coordination of MC production and facilitate the bookkeeping of the production activity.

- A MC request is executed by submitting one grid job. The JIM grid-to-fabric interface splits the grid job in multiple parallel batch processes. The multiplicity of these processes depends on the speed of the worker nodes and the total number of events to be produced for the given request. In case of failure of some of the batch processes, users can recover by resubmitting the same grid job, until the number of events specified by the request has been produced. The small number of grid jobs needed to execute a request improves the monitoring and bookkeeping of the production activity.

- To optimize the use of Mass Storage, the output from MC batch processes is merged to produce files of 1 GByte in size. We provide two mechanisms to run the merging application: (1) it can be run at the end of the MC production batch process i.e. on the same worker node; (2) output from each MC production batch process can be temporarily stored in volatile storage via SAM, then merged running a the merging application after the MC request has been executed.

In order to present the CDF users with a user friendly interface, the SAM-Grid makes available on the grid the typical CDF MC environment. This is how the environment is provided and used

- In general, the CDF MC environment cannot be pre-installed on a grid cluster. The SAM-Grid system makes it available to the job by dynamically deploying it at the worker nodes, before the job passes control to the application. The software is distributed in the form of a compressed archive (few hundreds MBytes in size) and deployed using SAM.
- A CDF MC request typically simulates events for a range of data acquisition runs. Each run is characterized by a different detector geometry and beam luminosity. The CDF MC application produces an output file of simulated events for every run. We considered two models to parallelize the production of events: (1) for every batch process, produce events for the whole range of runs in the request; (2) for every batch process, produce events for a subset of the runs in the request. Model (1) creates a large number of small intermediate files, as the CDF MC application produces an output file per run. Model (2) requires central coordination of what run numbers each batch job should produce, in order to avoid overlaps; on the other hand, it produces less intermediate files. We opted for model (2), as it results in a lower load to the data handling. The grid-to-fabric interface coordinates the preparation of the job environment for each batch processes, providing each with the appropriate non-overlapping range of runs. As an additional optimization for CDF MC, SAM-Grid provides the option of merging the output at the end of each batch process, thus further reducing the data handling load.
- The user can submit jobs by providing the following information: the request id (points to a description of the characteristics of the MC events to be produced); the version of the CDF MC software environment; the mapping between run number and luminosity (used to produce the correct relative abundance of events for different run numbers); the size of the merged file which should be produced; the number of events produced for each segment (parameter that influences the parallelization of the job onto several worker nodes).

3 Deployment

The deployment of the SAMGrid software is logically divided in the deployment of the following components: SAM station (data handling), JIM execution site software suite (grid-to-fabric interface, local monitoring, job environment preparation), JIM submission site software (robust grid queuing service), and JIM client site software (user interface).

In order to facilitate the evaluation of the system, we deployed the software redundantly, in order to increase the availability of the services. In particular, the SAM-Grid

(JIM execution site) has been deployed at two clusters at the university of Oxford and Glasgow. At both clusters SAM stations v6 were already available. The JIM software interacted directly with the SAM database via a single SAM DB Server v5. The submission site services were installed at Oxford.

Different ups databases have been used for the installation of the SAM and JIM software. This has the advantage that any changes of the SAM station installation do not affect the JIM installation. The JIM execution and submission site installations have been done with the help of the "samgrid" script, which allows the aggregated installation and easy tailoring of the underlying products. Comparing to earlier deployment attempts, the installation has been simplified by the samgrid product. Privileged access to the gateway machine is needed only once when configuring the security infrastructure (GSI for the gatekeeper).

To facilitate the deployment, we have developed a test script to check that all required services at the clusters are appropriately instantiated. The steps for the installation have been summarized at http://home.fnal.gov/~bartsch/jim/. For the client site installation a script has been developed in the manner of the init_sam script which implement version control. The script requires as input the experiment (DZero or CDF) and the SAM station name and takes a few minutes to run on a modern commodity CPU. As part of the project, we have written a user guide targeted to CDF users on job submission and the SAM-Grid job description language.

We conclude summarizing some relevant problems encountered during the installation.

- In order for the intra-cluster data transfer to function, the head and worker nodes of the cluster must know their fully qualified domain name. This can be achieved, for example, by writing these name in the /etc/hosts files of each node.
- The batch system can impose restrictions on the use of the resources at the worker node. These limitation include usable disk space, CPU wall clock time, etc. The limitation on the use of the disk space prevented the jobs from running. The batch system configuration should be checked at installation time, in order to facilitate a functional deployment.
- Because of the intensive I/O during the environment preparation at the worker node, the file system used for local scratch should be local. NFS areas have resulted unstable when used for this purpose.
- /etc/grid-security files and the configuration of the SAM batch adapters should be readable by user sam.

4 Evaluation

During the phase of design of the system, the CDF experiment has done an initial evaluation of the proposed data processing model (sec. 2), giving feedback to the

SAM-Grid team at the CDF GRID workshop in March 2005. These are the major points discussed.

- General questions about the architecture of the SAM-Grid infrastructure. In particular, we discussed the grid-to-fabric interfaces (job-managers) of SAM-Grid.
- Concerns about the version of the SAM API used by JIM. JIM currently uses SAM version 5 to interact directly with the SAM database server. Problems of maintainability of the infrastructure were discussed.
- Concerns about the ease of deployment of the infrastructure. In particular, some users favored an infrastructure where no middleware was to be installed at the execution sites. This concern is been addressed in the context of the LCG/OSG interoperability systems, but was not available at the time.
- Users suggested to implement file merging immediately after the phase of event production. This was implemented, as discussed in section 2.

After deployment, the JIM system for CDF montecarlo production was maintained operational for a period of four months. Due to other priorities in the experiment, during this time the system could not be completely evaluated.

After the four months of evaluation period, the deployment became non-functional due to system problems and hardware/system upgrades at the execution sites. These problems were not addressed as the experiment did not request to maintain the system operational for a longer time.

As requested by the team, a complete evaluation would include answers to the following questions, which currently remain open.

- Can you evaluate the data processing model and, especially, the use of the montecarlo request system?
- Can you evaluate the distribution of software via SAM as compared to the distribution via user-provided archives?
- Can you comment on the event production model? In particular, can you comment on the fact that each batch process produces events for a subset of the runs in a number proportional to the run luminosity?
- Can you comment on the output file metadata?
- Is the monitoring infrastructure adequate to to gather information about the running jobs and services?
- Any general feedback from the experience of running test jobs?

5 Conclusions

The SAM-Grid team and the CDF experiment have developed a prototype for producing montecarlo events via the JIM infrastructure. The prototype has been maintained functional for four months. During this period, because of other priorities for the experiment, a full evaluation could not be conducted. If CDF decided to evaluate the system at a future time, a new installation of a testbed would need to be made. The time estimate for this is several weeks. Also it is possible that recent interface changes in the JIM software may cause the MC prototype to malfunction. Fixing these malfunctions would probably take a few days for an expert.