CDF Run II Data Handling Resource Management and Tests

E. Buckley-Geer[1], *S. Lammel*[1], *F. Ratnikov*[2], *T. Watts*[2⋆]

[1] Fermilab, USA
[2] Dept of Physics and Astronomy, Rutgers University, USA

Abstract

The next collider run of the Tevatron, Run II, will start in March of 2001. The CDF experiment expects to record at least 1 PetaByte of data. Analysis of such a volume by a collaboration of over 450 physicists requires management of limited resources such as CPU time, disk space, tape drives, and I/O bandwidths. The resource management strategy is based on the use of batch queues and a disk inventory manager.

A prototype system was built and heavily exercised to evaluate strategy and performance. A first mock data challenge that uses most components of the data handling system has run. Results from both tests will be presented.

Keywords: data handling, prototype, resource management

1 Introduction

Run II data[1] in CDF is organized into datasets which are collections of events of particular content; examples of content type are raw detector data, reconstructed event data from a particular reconstruction version, mini-dst, ntuples, etc. The selection of events to include in a collection depends on online trigger decisions for datasets containing raw and initial reconstructed data ("primary datasets"); secondary and tertiary selections can be made on offline reconstructed quantities.

Events are collected into files of size one GB written in the ROOT[1] format. Files are grouped into a fileset; filesets into a dataset. The associations are kept in the "CDF Datafile Catalog" which is a database implemented centrally as an Oracle database and as a freeware database as needed elsewhere.

All filesets are stored on tape in the robot tape library[2], and the association of filesets to tapes is in the Datafile Catalog. Access to filesets by a user job is by staging disk and some filesets will be permanently present on staging disk. A user analysis job can specify a dataset and this translates automatically into a list of filesets to read.

The creation of datasets has two parts – a user job or data logger to write files, and data handling software to combine files into filesets and datasets and to transfer the data into the tape archive.

2 Resource Management

The management of limited resources such as CPU time, disk space, and tape drives is crucial in a central analysis computer cluster for big physics experiments and collaborations.

⋆ mailto:watts@fnal.gov
[1] http://root.cern.ch
[2] EMASS/2 AML from ADIC Corporation

2.1 Disk Management

To simplify a complicated problem, CDF data analysis and processing jobs read and write data only from or to disk; staging software transfers data between tape and disk. The amount of data in the experiment will be too large to fit all data on a disk array and the archiving of all raw and derived data will be done using tapes.

Some datasets will be accessed more frequently than others and if small in size, copies can be fixed permanently on disk. Larger datasets and less frequented datasets will be staged from tape to disk and reside in a shared array of disks for temporary use. A disk inventory manager (DIM) organizes the use of disks allocated both to the static and the temporary data. The quantum of data used for managing by the DIM is the fileset (several files). Data transfers between disk and tape move all filesets on a tape in one job.

The filesets in shared space remain in place until space is needed to fetch more filesets. The DIM uses an algorithm based on frequency of use and of time since last use to determine which filesets to remove when space is needed. in addition, filesets being processed by a user job receive a DIM use reservation until released by that job.

When a user job specifies a dataset to process, a data input software module[2] linked in the user job converts that dataset into a list of filesets using an access to the CDF Datafile Catalog. The DIM/Stager software in the job, at the completion of every fileset, then works to keep a small buffer of filesets reserved on disk ready for reading; it spawns an adjustable number of independent staging jobs to keep that buffer full. In this way the amount of shared disk space needed to support a user job will be, in general, a small fraction of the size of the dataset being read.

When a user job needs a list of filesets, a check is made to see if any are already present on disk. Thus multiple jobs reading the same dataset at the same time only generate one fetch from tape for each fileset in the dataset.

Controls and priorities on the amount of reserved and static disk space by group and/or user are plannned as a future feature of the DIM.

2.2 Batch Queues

The allocation of CPU cycles among users and groups has long been done by the use of batch queues. CDF plans to use LSF from Platform Computing[3] as a batch system on the Central Analysis Cluster.

There will be a set of queues based on the CPU requirements of analysis jobs and a second set of queues for the tape I/O staging jobs which are mostly automatically generated by the analysis jobs. There will be CPU queues for physics groups and special projects as well as general purpose use. There will be I/O queues for input, output, and for event picking, as well as for special purposes such as data logging of raw data and of reconstructed data from the reconstruction farms.

The control of priorities in the scheduling of I/O jobs will act as the management of the tape I/O resource.

Fairshare scheduling that takes account of the amount of CPU consumed by groups and users and of their quota will be used to insure that users get a fair share of the CPU and I/O resources. The CPU and I/O usage will be accounted separately and might have different quotas from each other. It also might be desirable to have the same quota apply to a group of queues so that a group/user that uses a large amount of CPU in one queue would have that counted against their use of CPU in other queues.

[3] http://www.platform.com

3 Tests of Disk Management and Staging Strategy

In Fall/Winter 1998-9, tests of the resource management strategy outlined in Section 2 were performed using a simulation prototype of the software architecture. The elements of the system were: 1) A set of basic batch queues of both classes, CPU and I/O, set up using the LSF product; 2) DIM and Staging software with all basic functions but not running in final multi-node architural form; 3) Simulated analysis jobs which read dummy events from files; 4) A set of big and small datasets (dummy data); 5) Basic CDF Datafile Catalog software with a catalog containing entries for the datasets and files of this simulation.

In order to exercise the resource management strategy, the tests were conducted in a limited computing environment: 1) Single CPU workstation (IBM, R6000, 63 MHz); 2) Staging disk of size 9 GB; 3) Two simulated tape drives, see below; 4) Four small datasets of 1 GB, and four large of 10 GB; 5) Two CPU queues, short and long, with 4 execution slots each; 6) Analysis jobs with variable (random) CPU time.

Note that the small datasets were about 10% of the available staging disk size, and the large did not fit completely on the staging disk. Tape storage was simulated by another disk drive which helps speed debugging. The number of simulated tape drives was controlled by the number of execution slots in the tape I/O batch queue and was set at two.

3.1 Simulation Scenarios

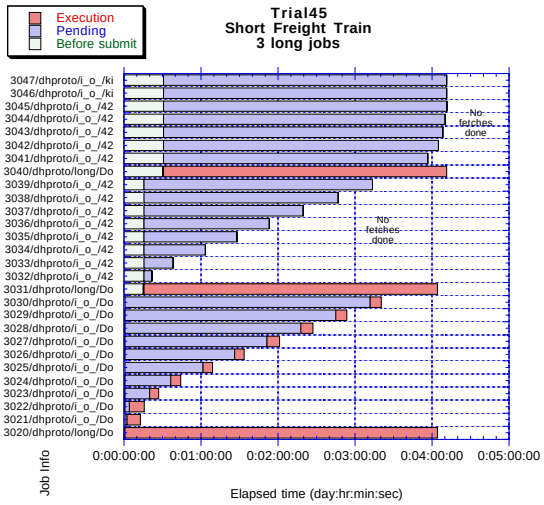

Figure 1: Three jobs (3020, 3031,3040) were submitted for batch execution on the same large dataset. The Stager software was needed to fetch filesets from tape in jobs parallel to the three analysis jobs, i.e. jobs 3021-3030, 3032-3039, 3041-3047. From the graph, it can be seen that there are no execution phases for jobs 3032-3039 and 3041-3047, so the software prevented multiple fetches of the same filesets from tape, thus conserving the use of tape drives.

The tests of resource management strategy were conducted in a series of “scenarios” each of which represented some aspect of the expected patterns of data analysis by users. Of particular interest were situations: 1) where several jobs processed simultaneously the same large dataset; 2) where a user resubmitted frequently a job on the same small dataset in order to tune plots.

The scenarios were studied for competition in reserved space on the shared disk, to check that staging jobs were executed only at the rate needed by analysis job’s speed, and to look for

unnecessary tape to disk transfers. No static filesets were used.

Some scenarios studied were:

One long job vs a stream of short jobs ("Trial29") The long job (analysis job reading one of the large datasets) did not hog the disk and no staging jobs were necessary for the stream of short jobs.

Three long jobs("Trial45") Figure 1 shows a scenario of one physics group processing a large dataset in 3 different jobs. Even though the large dataset jobs are submitted 15 minutes apart (about 1/4 day in full system), there are no extra fetches of filesets from tape to disk.

Ten long jobs ("Trial18") Ten jobs processing a large dataset were submitted at one hour intervals (equivalent to about a day for the full DH system). The substantial delay between the start of the jobs caused twice as many staging jobs as would have been necessary if all jobs started together. This is an acceptable result.

Mixed set of long jobs and users ("Trial20") This scenario had 6 long jobs by 6 different users reading 4 different long datasets, all submitted very close in time. There were 25% more staging jobs executed than the minimum, again an acceptable result.

Stream of short jobs battles 4 long jobs ("Trial40") There were four long jobs which read different datasets and were submitted close together in time. A continuous stream of short jobs alternated between two short datasets. This time, there were extra staging jobs executed for the short datasets but the rate was about 25% of the maximum potential rate; again this seems acceptable.

The Disk Inventory Manager (DIM) and resource management strategy worked well in prototype. The DIM functions were appropriate, simple to understand, and worked well. The behavior shown in the mixes of types of user jobs ("scenarios") gave good understanding for the ongoing development of the full resource management scheme.

4 Mock Data Challenge Experience

During December 1999 and January 2000, a test was made of the movement of data from the Level 3 Trigger online farm of microprocessors, through data logging into the robot tape library, through the reconstruction farm back into the tape library, and finally into a user analysis job. This was a test of connectivity of many different hardware and software components of the online and offline systems[1]. The data used was from Monte Carlo detector simulation of several physics processes of interest when Run II starts.

Data Handling components used were: 1) Central analysis computer cluster; 2) Tape library; 3) Datafile Catalog, Oracle version; 4) Disk inventory management software, prototype version; 5) Daemon to form filesets and stage data from disk to tape; 6) User analysis I/O modules which incorporated the resource management strategy of Section 2.

Results will be presented at CHEP2000.

References

1 E.Buckley-Geer, S.Lammel, M.Leininger, T.Watts, "Overview of CDF Run II Data Handling System", CHEP2000, Paper 366.

2 P.Calafiura, J.Kowalkowski, S.Lammel, M.Lancaster, E.Sexton-Kennedy, I.Sfiligoi, T.Watts, E.Wicklund, "The CDF Run II Datafile Catalog and Data Access Modules", CHEP2000, Paper 367.