3B2v7.51c GML4.3.1 NIMA : 14216 Prod.Type:COM pp.1–5(col.fig.:NIL) ED:Vijaya PAGN: nmanjunatha SCAN: PREETHI

ARTICLE IN PRESS

Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Nuclear Instruments and Methods in Physics Research A ▮ (▮▮▮▮) ▮▮▮–▮▮▮

NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

www.elsevier.com/locate/nima

Computing for Run II at CDF

M.S. Neubauer

Physics Department, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

On behalf of the CDF Collaboration

Abstract

With Run II of the Fermilab Tevatron well underway, many computing challenges inherent to analyzing large volumes of data produced in particle physics research will need to be met. We present the computing model within CDF designed to address the physics needs of the collaboration. Particular emphasis is placed on current development of a large O(1000) processor PC cluster at Fermilab serving as the Central Analysis Farm for CDF. Future plans leading toward distributed computing and GRID within CDF are also discussed.

1. Introduction

Run II at the Fermilab Tevatron began in March 2001 and will continue to probe the high energy frontier in particle physics until the start of the LHC at CERN. The accelerator facility underwent a major upgrade for increased energy ($\sim$10%) and instantaneous luminosity ($\times$10) over that attained in Run I. With a goal of attaining 15 fb^{-1} of integrated luminosity over Run II (2 fb^{-1} during Run IIa) for each experiment, a very rich and exciting physics program [1] at Fermilab is expected over this decade.

In order to operate at the upgraded Tevatron and to exploit the physics potential of the new beam conditions, the CDF detector also underwent a major upgrade [2]. By the end of Run II, it is expected that the CDF collaboration will write up to 10 Petabytes of data onto tape. Providing efficient access to such a large volume of data for analysis by hundreds of collaborators world-wide will require new ways of thinking about computing in particle physics research.

2. Overview

A broad overview of the data acquisition and analysis flow in CDF is shown in Fig. 1. Nearly one million channels of electronics process detector data resulting from $p\bar{p}$ interactions in the Tevatron. An increasing amount of detector information is passed on to a series of trigger subsystems (levels) which make the final decision as to whether or not the event is interesting enough to record. Raw data is logged to tape (via an intermediate cache disk) at an average rate of around 40–75 Hz or $\sim$20 MB/s. The raw data is reconstructed and validated on a 169 dual CPU (equivalent to 300 1 GHz Pentium III (P3)) PC cluster and then written to an STK tape robot. Data in the tape robot is accessible from the

E-mail address: msn@fnal.gov (M.S. Neubauer).

0168-9002/03/$ - see front matter © 2003 Published by Elsevier Science B.V.
doi:10.1016/S0168-9002(03)00449-2

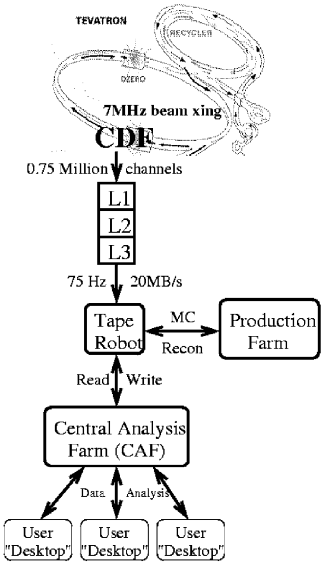

Fig. 1. Overview of CDF data and analysis flow among several of the major offline subsystems.

production farms as well as the Central Analysis Farm (CAF). The Enstore software system developed at FNAL provides the interface layer to network-attached tape drives in the robot. In cases where fast, frequent access to large amounts of tape data is required (e.g. CAF), significant read and write cache disk is employed. The dCache software system from DESY provides a scalable network-attached front-end disk cache to Enstore. The dCache system is currently under beta-testing within CDF.

SAM [3,4] is also under intensive evaluation for use in CDF data handling. SAM provides distributed data access, flexible dataset history and management, and optimizations for limited fabric resources (e.g. network or tape bandwidth), and has been demonstrated to work with Enstore and the CDF analysis software framework.

An Oracle-based database system is used within CDF to store and provide access to metadata and calibrations for variety of different consumers—online monitoring, production farms, and offline analysis jobs. The online and production farm database servers are Sun Enterprise 4500 machines. This database is replicated to an Intel 4-way SMP server running Oracle on Linux which serves the CAF and remote offline analysis needs.

3. Computing requirements

It is important to understand the data and software characteristics involved before one sets out to solve an analysis computing problem. In the context of CDF data analysis, we are trying to process a very large number (10^7 or more) of relatively small (hundreds of kBytes) independent data elements. As such, we have the relative luxury of speeding up a typical analysis job through parallel processing of independent subsets of the job. Some additional characteristics of the CDF data and analysis software:

Data characteristics:

- Root I/O as the persistent data format.
- Typical raw (reconstructed) data size of 250 (50–100) kB/event.
- Typical Run IIa dataset size of 10^7 events.

Analysis software characteristics:

- Typical analysis jobs run at 5 Hz on 1 GHz P3, corresponding to a few MB/s input rate.
- Analysis jobs are CPU rather than network I/O bound over Fast Ethernet.

Based upon Run I experience, we expect to have roughly 200 users simultaneously running analysis jobs on the central system at a given time. Our goal is to provide sufficient computing resources to allow each of these users to process a typical secondary dataset (e.g. produce standard ntuples) in one day. The computing requirements to achieve this goal given our data and software characteristics are shown in Fig. 3 for a projected Tevatron luminosity schedule. As can be seen from Table 1, we expect to need $\sim$700 TB of disk and $\sim$5 THz of CPU by the end of FY'05 to meet our computing needs. At present, we require a large amount of inexpensive commodity hardware to feasibly attain this level of computing power. This leads us to a central analysis computing model of a large batch farm of linux-based PC's with fast access to a substantial amount of IDE disk in a RAID configuration (for redundancy and speed) with hot-swap capability for ease of maintenance. This disk serves as a cache disk layer (dCache) to the tape robot and the Enstore system. It can also

Table 1
CDF Computing needs for Run II. "THz" is relative to 1 GHz P3 performance

Fiscal year	Integrated luminosity (fb^{-1})	Batch CPU (THz)	Farm CPU (THz)	Static disk (TB)	Read Cache (TB)	Write Cache (TB)	Disk I/O (GB/s)	Archive I/O (GB/s)	Archive volume (PB)
2002	0.3	0.5	0.37	82	26	9	0.8	0.07	0.3
03	0.9	1.0	0.33	98	28	8	0.9	0.19	0.4
04	1.3	1.4	0.06	160	46	13	1.4	0.13	0.4
05	1.6	1.8	0.54	200	60	16	1.8	0.48	0.6
Total	4.1	4.7	1.3	540	160	46	4.9	0.87	1.7

be used for static file export of standard datasets to the CAF or other remote clients.

4. Central analysis farm (CAF)

The computing model for CDF central analysis is shown schematically in Fig. 2. Users develop and debug their analysis jobs on their desktop[1] and then submit these jobs to the CAF via a custom interface to the FBSNG batch system.[2] Output is then sent back to the user's desktop or to another remote machine for later retrieval (note that the latter does not require continuous network connectivity of the submission desktop).

4.1. Hardware details and performance

The CAF functionally comprised three types hardware—"workers" where users' jobs are executed, "servers" which serve data to analysis jobs running on worker nodes, and "infrastructure" nodes which provide important utilities for the CAF (databases, code servers, etc.).

The CAF is currently composed of 134 worker CPUs, with an additional 468 CPUs incorporated in the farm by the end of this year (2002). The present worker nodes are a mixture of 1U and 2U dual Athlon MP 1600+ (1400 MHz) and dual Intel P3 1266 MHz machines. Each has 2 GB of

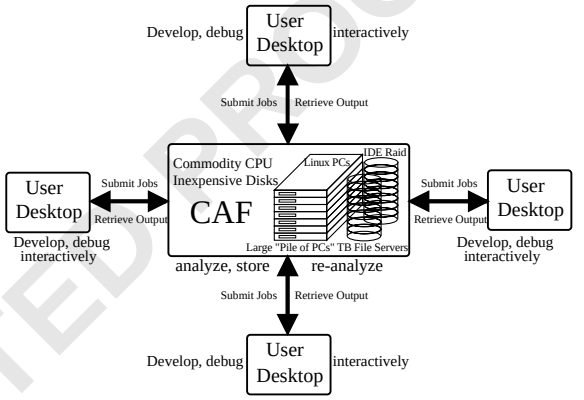

Fig. 2. Computing model for CDF central analysis

RAM, single Fast Ethernet connection to a central Cisco 6509/13 switch, and 80 GB of job scratch disk.

Large-scale deployment of self-contained IDE RAID 4U server units (disk, CPU, network device, etc.) satisfies the substantial disk space and I/O bandwidth requirements of the system. Each unit serves 2.2 TB of disk (16 160 GB IDE drives connected internally to two separate 3ware Escalade 7850 8-port IDE Raid controllers and configured as RAID50) via Gigabit Ethernet (SysKonnect 9843) connection to the central switch.

Currently, the primary task of the file servers is to export static files comprising secondary datasets to the worker nodes via automounted NFS(v3) and rootd[3] protocols. Fig. 3 shows the aggregate

[1] In this context, "desktop" refers to a linux-based PC with access to the CDF software environment.

[2] Farms Batch System (Next Generation) developed at FNAL-CD.

[3] Remote access to ROOT files using the rootd daemon and

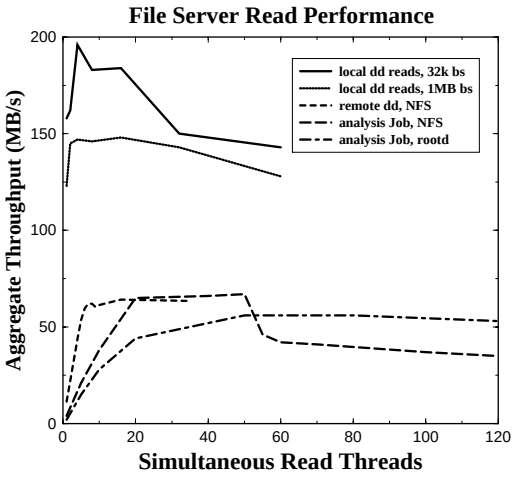

Fig. 3. File server throughput performance.

server throughput versus number of simultaneous client requests for both local disk reads and reads over the network. In terms of server performance, we make the following general observations from these tests as well as in situ system monitoring information:

- Local disk reads between 130 and 200 MB/s, depending on block size and # of simultaneous reads.
- Reads over the network of up to 70 MB/s (aggregate), with the server CPU utilization the limiting factor for throughput.
- Similar performance for NFS and rootd, with agreement between in-situ monitoring and dd tests.
- Under typical farm usage conditions, average aggregate file server I/O bandwidth is 4-8 TB/day.

4.2. Software implementation

The design goal for the CAF software is to provide users with secure access to CAF resources (batch CPU, scratch disk, data handling system) from their desktops anywhere in the world. To successfully implement such a system, we need to work within several somewhat contradictory design constraints and desirables:

- *FNAL security policy*: Kerberos authentication lab-wide.
- *Job scheduling*: proven batch system, configurable, fair share capability, local support → FBSNG.
- *Administrative ease*: no user accounts, jobs run under single 'cafuser' UID.
- *User identity*: Unique privileges for batch jobs and disk resources.

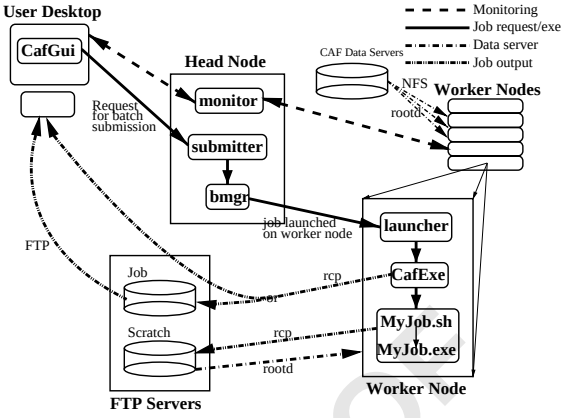

Fig. 4. CAF software implementation.

Fig. 4 depicts the CAF system flow, with the job request/executable, data, job output, and monitoring paths explicitly shown. To submit a job, the user inputs required information about the job (shell script to run, local path to executable(s) and shared libraries, etc.) into a kerberized client interface. The interface authenticates the user (i.e. presents a valid kerberos ticket) to a "submitter" server daemon which receives the job information and a tarball containing the shell script + executable + any required control files, shared libraries, etc., and performs the actual job submission to the FBSNG Batch Manager (bmgr). Once a user's job is scheduled and sufficient CPU resources become available, a standard executable (CafExe) common to every worker node is launched with

- a common 'cafuser' UNIX user ID.

(*footnote continued*)
the TNetFile class provided by CERN's ROOT package.

- appropriate parameters to completely specify the user's job.
- a kerberos principal unique to the user, generated from a single "service" principal on the farm.

CafExe copies over and unpacks the user's tarball from the head node, sets up the proper environment for CDF analysis software, and runs the user's shell script which in turn executes whatever the user has specified. Job output is sent to the location requested by the user for later retrieval or for input into a subsequent CAF analysis job. The user-specific kerberos principal is used for unique access privilege to the output location(s). Several file servers within the CAF are used for job output scratch disk, with users able to access their scratch space through a custom graphical interface to standard kerberized FTP.

Another software design goal was to provide the user monitoring and control capabilities as if the job was running on their local machine rather than a remote, non-interactive batch farm. Through a set kerberized client-server pair utilities, users are able to

- Submit, kill, and monitor their jobs.
- Generate file listing in their job's relative path (e.g. to see what is being generated) or the absolute path on any CAF node.
- Tail any file (e.g. log file) generated in their job's relative path.
- Monitor the worker node resource utilization (e.g. CPU, memory) of their jobs.

In addition, a web interface to FBSNG provides a wealth of monitoring information to the user.

After a six month period of design, prototyping, and commissioning, the CAF system was put into production use by the CDF collaboration in May 2002. Since May 2002, the CAF has been under continuous use for physics analysis by a base of nearly 300 users within the collaboration.

5. Toward the grid

Although Run II will not have nearly the amount of data or number of collaborators as the upcoming LHC experiments, it should be recognized that CDF and DO represent a testbed for distributed computing and GRID.[4] If some of the challenges of distributed access to data and CPU resources could met in such a way as to improve the physics output of the experiments, this would be a success for the particle physics community.

Additional remote farms are presently being installed with CAF infrastructure software. Initially, the user is required to choose between farms at submission time, while data movement is transparent, and controlled by SAM. In the future, we envision building a global job scheduling tool on top of the existing infrastructure (see Fig. 5). The global job scheduler minimizes job execution time by co-locating data and CPU resources.

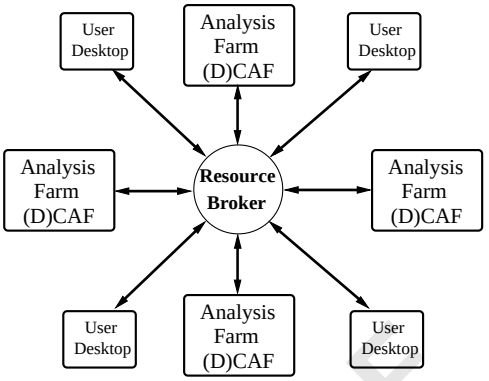

Fig. 5. Conceptual picture of peer-to-farm job brokering.

References

[1] P. Bhat, Nucl. Instr. and Meth., these proceedings.
[2] CDF Coll., The CDF II Technical Design Report, FERMILAB-Pub-96/390-E, 1996.
[3] G. Garzoglio, Nucl. Instr. and Meth., these proceedings.
[4] I. Terekhov, Nucl. Instr. and Meth., these proceedings.

[4] We remark that Moore's Law considerations may make some aspects of the CDF/D0 computing problem even more difficult than those to be encountered at the LHC.