Large Scale Test of D0 Reconstruction Farm

H. Schellman[1,2], *M. Albert*[2], *J. Bakken*[2], *L. deBarbaro*[1], *M. Breitung*[2], *M. Diesburg*[2], *D. Fagan*[2], *J. Fromm*[2], *L. Giacchetti*[2], *D. Holmgren*[2], *T. Jones*[2], *T. Levshina*[2], *L. Lueking*[2], *I. Mandrichenko*[2], *S. Mayola*[2], *A. Moibenko*[2], *R. Pasetes*[2], *D. Petravick*[2], *M. Schweitzer*[2], *K. Shepelak*[2], *I. Terekhov*[2], *J. Trumbo*[2], *S. Veseli*[2], *M. Vranicar*[2], *R. Wellner*[2], *S. White*[2], *V. White*[2]

[1] Department of Physics and Astronomy, Northwestern University, Evanton IL, USA
[2] Computing Division, Fermi National Accelerator Center, Batavia IL, USA

Abstract

We report on a full system test of 1/3 of the planned reconstruction farms for the D0 experiment. 50 dual-processor Pentium PC's were used to process simulated events. Data were read from a tape robot to the PC's, spooled to a large disk buffer and then written back to the tape robot. The tests were at the data rate and cpu utilization expected during real operation.

Keywords: Linux,Farms,PC,commodity

1 Introduction

The D0 experiment is expected to start taking data for its second run (Run II) in March 2001. Data rates from the detector to the reconstruction farm will be ~ 50 Hz peak with an average of 20Hz. Event sizes are expected to be 0.25 MB and CPU time/event is expected to be in the range 5-10 sec on a 500MHz Pentium processor. These rates and event sizes imply a peak input bandwidth of 12.5Mb/sec to the reconstruction farms and a need for 250-500 processors in the final system to handle the peak rates. Output bandwidths are lower due to a smaller record size for reconstructed data and event filtering but still significant.

D0 ran a reconstruction farm consisting of 100 Unix (SGI and IBM) workstations during Run I (1992-1997). Data were copied to and from 8mm tape drives attached to 4 SGI data server and control nodes. The control software for the system was largely written by D0 physicists and shipped each event to a processing node and then recovered it. Significant data loss due to problems with 8mm tape technology[1] led to severe reprocessing and book-keeping loads. Several veterans of the Run I experience were involved in the design of the Run II data handling and farm systems and that experience has led to a significant change in design philosophy. The two most important changes are the move to an integrated data handling system and a change in the granularity of the records passed to the processor nodes. In the new system, instead of individual events, each 1GB file ($\sim$ 4000 events) is sent as a whole. This has resulted in both a simpler system and far more efficient use of CPU on the processor nodes.

The Run II farm design separates data storage and book-keeping from farm operations by use of the Fermilab Sam[2] and Enstore[3] data handling systems. The Fermilab Farm Batch System[1] is used for job control. These are general products produced largely by computer professionals rather than physicists, as a result they have documentation, defined interfaces and reliable support. They are described in other contributions to this conference. [4, 5, 6] As a result of the use of these external products the farm specific coding has been reduced to a small number of python scripts and and was done by one physicist working part time.

During 1999 farm design and job control scripts were tested on a small prototype farm consisting of 5 PC's running the Fermilab version of Red Hat Linux 5.0. These tests included the pro-

[1] We were replacing $\approx$ 1 drive/day.

cessing of 90,000 Monte Carlo events and were generally successful but indicated a need for faster disk and networking as I/O throughput was limited to around 10MB/s.

In October 1999, Fermilab purchased 50 PC's as the first of several farm purchases for D0. Each PC has 2 500 MHz Pentium III processors, a 6 GB system disk, 2 16GB IDE disks for data, 512MB of memory, floppy and CD drives and a 10/100 MB ethernet card. The SPECINT95 for each of these processors was $\sim$21. The operating system is a 'farm' version of RedHat Linux 5.2 with non-essential features removed. The PC's were connected via a Cisco 6509 switch to the SAM and Enstore systems and to the D0 farm control system, a 4 processor SGI Origin 2000 with 1GB of memory and 200GB of disk (2-way striped), in early December. The Origin serves the home areas for all 50 PC's, serves as a disk buffer for output files and is the master for the FBS batch system. It has two Gigabit Ethernet cards configured to talk to the farm worker nodes and the Enstore tape system respectively. In December small-scale tests were done on a 10 PC subsystem to verify the configuration. Minor changes (such as reenabling X) were made to the default farm Linux configuration to make it easier to diagnose problems on PC's. In early January 2000 all 50 nodes were fully configured and full system tests were done, using the SAM/Enstore data handling system to deliver data to and from the PC's.

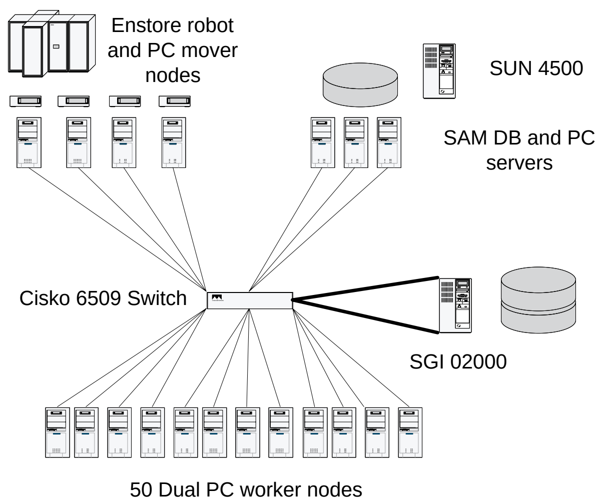

Figure 1: The 100 CPU D0 reconstruction farm. Thin lines indicate 100MB ethernet connections, thick lines are Gigabit ethernet.

2 Design Philosophy

Our farm philosophy is to use general products as much as possible and keep the farm specific D0 code as simple as possible. We use two major components.

The SAM/Enstore data delivery system consists of a tape robot with (currently) 10 Mammoth I tape drives connected via the D0 switch to the farms. The Enstore system handles tape mounts and data delivery via the pnfs file system while the SAM system provides a full file catalog and processing database. Once a connection to the SAM system has been established, any process on the farm can request 'the next file' and it will be delivered. The SAM system does not count the file as successfully processed until the farm process 'releases' it with a status code. The SAM system can thus provide a summary for any farm job of file deliveries and the processing status of any file.

The Farm Batch System allows the user to submit jobs with a simple control structure and dependencies. It also provides simple system status displays. FBS is used by many experiments at Fermilab. D0 farm jobs use only the minimal FBS functionality and consist of

A Start Section In the start section, which runs on a single node, the SAM data handling system is informed that a job required a given list of files is starting. Directories for output files are set up on the control node.

The Worker Sections Once the start section has begun, processes are started on N worker nodes. One process is run per processor due to the large size of the D0 executable and the batch system is configured to impose this constraint. When a process starts on a worker node, the environment needed to run the reconstruction, including executables, control files and stager processes is copied across the network and built from scratch. The worker process then requests data files from the SAM system, processes them and copies the output and log files to the output directory on the control node. The SAM system delivers the 'next' file in response to a request, thus if a worker machine crashes, processing of 1 (or 2 in the case of dual processors) files is lost but the system as a whole can proceed to completion. The SAM database will flag the missed files as not having been successfully processed as it will not have received the 'release' message. When a worker section can receive no more files, because all have been delivered, it terminates and the entire local D0 environment is deleted.

The End Section When all worker sections have terminated, the End Section is run. It informs the SAM system that no more data are needed and initiates a dump of the output files back to the tape robot if needed.

This system was designed to be robust against hardware failures and easy to maintain. For example, once a 10 node system had been debugged, extension to the full system took less than a day and went flawlessly. No D0 specific modifications needed to be made to the farm nodes except for the mounts of pnfs and the d0farm home areas. In addition, no human intervention is needed to change the software versions running on a given worker node as it is downloaded for each job.

3 Description of the test

$\sim$421 files of digitized simulated D0 events[7] ranging in size from 200-700 MB were processed on the farms as a part of this test. The files were divided into 6 'projects' each with a different physics process (Z$\rightarrow ee$, Z$\rightarrow \mu\mu$, ttbar, bbbar and low and high pt jets. The files each took between 3 and 9 hours to process through the D0 reconstruction code. These file sizes and reconstruction times are similar to those expected during real data taking. Six farm jobs, one for each physics 'project' were submitted at the same time and ended up running on 95 worker processors (because the jobs submitted allocated processors in multiples of 5 and one dual machine was down, leaving 3 idle nodes.) The input data were on 15 different tapes. A maximum aggregate transfer rate of 15 MB/s was observed from the tape drives to the farm worker nodes, consistent with the five 3MB/s tape drives observed to be in use. This exceeds the peak rate expected during data taking.

Data files were processed through the full D0 reconstruction code on the worker nodes and written into directories on the SGI control node. After the job had terminated all files from one job were written back to tape via the SAM/enstore system. The whole test took approximately 24 hours to complete and showed no failures attributable to farm hardware or software. Some files were not delivered by the SAM system due a server problem near the end of the test but the database tracking system successfully identified the missing files for resubmission. After repeating similar tests, we concluded that, over time scales of several days, the farm system is highly reliable and can handle a load which emulates real data processing. Because the SAM data access system was used for these tests, we have complete information on the processing status of each data files including lineage from the event generation, through the simulation step, to reconstruction.

4 The future

These tests have reassured us that the basic hardware and software configuration is easy to install and maintain and can handle data rates and CPU loads expected in real data processing. For full data taking, where 3-5 times as much CPU power will be needed, the existing system could be cloned with little increase in complexity. As a result, we are reasonably confident that the hardware configuration will operate under full load. However, much of the job creation and control was done by hand for these tests. Significant work remains to be done in automating the job submission scripts and handling reprocessing of files which fail due to hardware or software errors.

Acknowledgements

We would like to thank the Fermilab Computing Division for their invaluable assistance and advice in contructing this system. We would also like to thank the CDF farms and data handling groups for many useful discussions.

References

1 `http://www-hppc.fnal.gov/fbs/fbs.html`
2 `http://d0db.fnal.gov/sam`
3 `http://d0ensrv2.fnal.gov/enstore/`
4 L. Lueking *et al.*, The Data Access Layer for D0 Run II: Design and Features of SAM, Abstract 241, these proceedings.
5 D. Petravick *et al.*, Fermilab Computing Division Systems for Scientific Data Storage and Movement, Abstract 176, these proceedings.
6 I. Mandrichenko *et al.*, Farms Batch System and Fermi Inter-Process Communication toolkit, Abstract 191, these proceedings.
7 G. Graham *et al.*, The D0 Monte Carlo Challenge, Abstract 311, these proceedings.