

# High Throughput Linux Clustering at Fermilab

Steven C. Timm, Troy Dawson, Karen Shepelak, Lisa Giacchetti,  
and Dane Skow

*Fermilab, Batavia, IL, USA.*

## Abstract

Computing in experimental High Energy Physics is a natural problem to use coarse-grained parallel computing structures. This report will describe the development and management of computing farms at Fermilab\*. We will describe the hardware and configuration of our 1000-processor cluster, its interface to Petabyte-scale network-attached storage, the batch software we have developed, and the cluster management and monitoring tools that we use.

## 1. Introduction

High-energy experimental particle physics has four major uses for massive amounts of computing. These are simulation, data acquisition, event reconstruction, and theory. High-throughput Linux clusters are now regularly used at Fermilab and other laboratories for all of these purposes. All four phases of computing have as a common feature the idea of an “event.” In a typical event at Fermilab, one proton collides head-on with one anti-proton, and tens to hundreds of particles come flying out from the collision. But this one event can be analyzed independently of the hundreds of millions of other such collisions that are happening. This means, that, in theory, each of the millions of events could be farmed out to an independent processor. It is therefore to the advantage of Fermilab to find cheap commodity computing and apply it to the task at hand.

Planning often begins for experiments ten to fifteen years ahead of time. Before an experiment is constructed, physicists do simulation of how the detector will respond. This is done to determine if they will be sensitive to new effects, to

---

\* Work supported by the US Department of Energy under contract No. DE-AC02-76CH03000.

determine how to design their detector, and to figure out how to reconstruct the output once it comes.

Data acquisition is also a crucial function. Given the huge event rates (1.5 million events/day, at 400 kilobytes each) that are coming from the detectors, it would be impossible (and horribly inefficient) to write all the output to tape. Many Linux-based PC's are used in the Level 3 trigger to select the interesting and well-measured events to write to tape.

The third task is what is referred to as reconstruction. The information that comes from the detector is a set of points in space that correspond to a particle trajectory. It is the goal of the reconstruction program to find these trajectories. Physics analysis then combines the reconstructed tracks for each other to see what these tracks may have come from. The Linux clusters that do this task are referred to as "the farms." This computing is typically quite integer-intensive and the performance on the code scales linearly with processor speed. On dual CPU units, two of these processes running simultaneously finish in the same time as one would that is running alone. This indicates that, although the code uses a huge amount of memory, it does not require a high memory bandwidth. A summary of the results is typically placed on disk and then made available from a large SMP machine (in Fermilab's case, a 176-processor SGI) with a huge amount of attached disk. A number of repetitive jobs are then run against this data to hopefully obtain physics results.

The fourth task, which requires a highly parallel Beowulf-like cluster known as the PCQCD cluster, is theoretical calculations based on the results of the experiment. This, in turn, feeds into experimental design again and the cycle repeats [1].

## **2. High-throughput clusters at Fermilab**

Fermilab has used high-throughput clustering for quite some time. Before the advent of Linux-based PC's, farm worker nodes included VAX, custom-made Motorola 68020 based boards, and more recently IBM and SGI RISC-based workstations. The typical characteristics of jobs that have been done on the farms are that they have greater than 1000 instructions executed per byte of I/O. Although the total throughput of data is staggering, the rate at which the data is read is relatively slow.

Since each node of the cluster is working on an independent piece of data, there are no high-availability requirements. The large number of worker nodes ensures that production can continue even if a few nodes break down. The nodes have only same-day service during the business day. If a worker node breaks after hours, it is fixed the next day. Likewise, there is no peer-to-peer communication necessary between the nodes, although in theory it is possible.

High-energy physics code does not lend itself well to parallelization or vectorization, so the message-passing routines such as MPI are of limited utility.

A typical example of such an experiment would be Fermilab's experiment E871. This experiment ran in 1996 and again in 1999, collecting 20,000 tapes of data at 5 gigabytes apiece, for a total of 100 Terabytes of data. The processing of this data is nearly complete on one of our older Linux farms, and has taken approximately ten months. There is an SGI-based I/O server with 20-30 tape drives which is used to read the data in from tape and copy it to the worker nodes. The worker node works on the data for several hours, and then copies a result file back to the I/O node. A number of output files are collected at once, and then spooled back out to tape.

For the current generation of collider experiments at Fermilab, the goal is to be able to reconstruct all the data that is taken on the same day. Since each of these experiments will be taking of the order of one Petabyte of data per year, the amount of computing required is much more than what has been assembled heretofore. Instead of using manual tape mounts, all tapes are processed in a tape robot. This tape robot itself is controlled by a Linux cluster of nodes known as the Enstore system, which uses a software file system called PNFS to map all of the files stored in the robot to a Unix-like name space. Then a set of mover nodes is connected to the robot, with two tape drives per Linux node, and these are what moves the data out to the worker nodes or wherever else it is needed.

Experiments being planned to run in 4-6 years from now are planning much more ambitious computing farms, hoping to complete the reconstruction in near real-time. It is estimated that the CMS experiment, which will run at the CERN laboratory in Geneva, Switzerland, will require the equivalent of one million SpecINT95. This is a factor of sixty more than the capacity of the current farms. A Fermilab experiment currently in the planning stage, BTeV, also expects to need computing at a similar scale.

### **3. Current Hardware Configuration**

There are currently 458 dual Intel-based farm worker nodes in our reconstruction farms at Fermilab. The oldest cluster, based on dual 333 MHz Pentium II, has been running for almost three years. These nodes are in a small-footprint desktop configuration with one 6-gigabyte disk that serves as system and data disk. We also have 138 dual units in mini-tower cases that were purchased in late 1999. These are dual 500 MHz Pentium III machines with a 6-gigabyte system disk and two 18-gigabyte data disks. Since summer of 2000, we have bought rack-mount worker nodes in the 2U form factor. The latest installment of 136 1 GHz dual Pentium III worker nodes has just arrived at

Fermilab. These feature a 20-gigabyte system disk and two 40-gigabyte data disks.

The nodes are divided into three groups, one for the CDF experiment, one for the D0 experiment, and one for the fixed target experiments. An SGI Origin 2200 I/O server serves each group of nodes. The I/O node is an SGI Origin 2200 with four CPU's. It serves as an NFS and NIS master for all the worker nodes. Since most new production gets its input directly from the tape robot, the main function of the I/O node is to concatenate the output files after the jobs are done and send the results back to the tape robot [3].

The farm is connected to a dedicated Ethernet switch. Each worker node has a 100 megabit/s connection, and the I/O node has a 1 gigabit/s connection to the switch. The tape robot and the machines that control it are also connected to the same switch by 1 gigabit/s Ethernet.

We can access the nodes even if there is a network problem, or if the network is not configured, by use of a console server. All console output is redirected by the OS to come out the COM1 port. Each worker node has its COM1 port connected by serial line to a Cyclades CYCLOM-Y console pod. Up to 128 such ports can be accommodated on one PC, which is called a console server. It has software that allows a virtual console to be shown for any of the machines hooked up to it. Also, it makes a log of all console messages that are generated.

Fermilab has a list of qualified vendors for Linux farm workers and for Linux desktop machines. We develop this list by asking a large list of vendors to submit evaluation units within a certain specification. The vendors are evaluated on the quality of the hardware they provide, their competence in installing Linux, the quality of their service, and the price/performance of their unit, as measured by benchmarking against the actual code run by our experiments. Vendors bid on a fully-integrated, racked solution including the worker nodes, rack, power supply, cabling, and patch panels.

## 4. Software Management Tools

We have developed a large number of management tools for installing, configuring, and maintaining the farms. We also have developed our own batch software.

### 4.1 Fermi Linux

Fermi Linux is our own distribution of Linux [4]. As a starting point we take a version of Red Hat Linux, currently version 6.1, and add a number of packages that are used here at Fermilab. We also make a number of security patches, turning off all services by default. Although we are nominally a 6.1 version, the

install images and kernels are updated from time to time to keep up to Red Hat security updates. The distribution is available to on-site users via NFS mount for network installs, and CD-ROM's are made available for off-site users and vendors. We have a list of qualified vendors who will ship the machines with Fermi Linux pre-loaded.

Updates are spread around the 1000 or so Linux nodes on site by using AutoRPM. Each node has a cron job to periodically check for updates from the server, and it downloads them and automatically installs them.

#### **4.2 ICABOD**

ICABOD is an acronym for Install and Configure, A Ballet on Disks [5]. We have a large number of nodes to install that differ only by node name and IP number. Typically nodes arrive from the vendor configured to get their network address via DHCP. The install RPM then copies the install image to the local worker node, boots off of it, and does a fresh install of Linux with the latest version at Fermilab. The configuration script, using an expect script, goes through changing the root password, copying the ssh keys to the node, partitioning the disk and copying various files to it that weren't covered in the install. These scripts allow us, in emergency, to re-install the entire farm in 20-30 minutes, and this is only limited by the speed of our NFS install server.

#### **4.2 Management Tools**

We have a number of scripts to perform various functions quickly on the farms, including changing passwords, copying files to all the worker nodes, or remotely executing a command. We have now integrated these into a single tool, available either with X windows interface or from the command line. This has the effect of having our current cluster configuration be software-configurable in one place rather than hard-coded in a number of scripts.

Even with Autorpm, it is still possible that the software of the farm workers will eventually diverge. The Fermi In-Sync Suite is a program that records the RPM content of each worker node at the beginning of the month and compares it to a known reference configuration, and then goes through daily and notes any changes that have happened.

#### **4.3 Burn-in**

All worker nodes that arrive at Fermilab are put through a thirty-day burn-in process. We use the [seti@home](#) software to fully load both CPU's. Then once an hour we use the "bonnie" software, which is a disk testing utility, to write a 1-gigabyte file to each disk. Also, simultaneously with the disk write, we run a network test to each node of 400 megabytes. It has been our experience that this is sufficient to shake out many of the initial problems of the nodes as they are

delivered. In addition to the errors that are visible from the operating system, we examine the hardware-level sensor logs to look for problems with memory and power supply.

If a node is down for any part of the day, it is considered down for the whole day. The full group of systems is required to have at least 98% uptime over the course of the 30 days. Any node with five days of down time or more is declared a lemon and must be replaced by the vendor.

#### **4.4 NGOP**

The farms at Fermilab are the beta-test sites for what will eventually become the standard monitoring program at Fermilab, the Next Generation Operations program [6]. There are several components to this software. Each node has two monitoring agents running as daemons on it. One daemon uses “swatch” to examine /var/log/syslog periodically. This detects issues such as hard disk timeouts, NFS timeouts, and daemons dying unexpectedly. It could easily be expanded to detect security intrusions. Another one measures the health of the system by checking that all the disks are mounted and accessible, that crucial daemons are running, and checks the level of memory, swap space, and disk space. It also uses either the lm\_sensors package or proprietary Intel code to read the hardware sensors and check temperatures, voltages, and fan speeds.

All changes in status observed by the daemons are forwarded to a central server, which the users access via a GUI interface. For major events such as a system going down, E-mail is automatically generated and sent. Eventually the system will be configured to page the person responsible automatically. All the configuration for the monitoring agents and the GUI is implemented in XML, which gives enormous flexibility for future expansion.

#### **4.5 FBSNG**

Initially the LSF batch system was used to allocate resources on our Linux clusters [7]. Due to prohibitive license costs, Fermilab staff developed a batch system that would be adequate for our needs. The Fermi Batch System, Next Generation, is a batch system that allows for jobs to be broken up into various sections, each requiring different resources. For example, there is often an input section that reads the tape and farms out the data to several worker nodes, a number of parallel computing sections, and an output section which concatenates all the results which will not run until all the parallel computing sections are finished. FBSNG fully supports strong authentication, requiring the job submitter to have proper Kerberos credentials and giving them to the jobs that are executing [8].

### **5. Future Plans**

Current Fermilab requirements as we understand them lead us to believe that our Linux cluster capacity will grow by at least a factor of ten over the next four years. In preparation for that, we are moving our hardware design beyond the rack level to what we term the “pod” level, a cluster of six racks with an integrated console server, networking, and display in the middle.

Given the increasing demand for cycles, and the decreasing amount of money, it is likely that Linux-based machines will also expand into the areas that heretofore have been served by other Unix vendors. Namely, using Linux machines as NFS/NIS servers and using Linux compute farms to supplement the large SMP boxes that do the repetitive analysis of disk-based data sets. Actually, a different way to share common files besides NFS will be necessary as we are already pushing the edge of NFS performance. A workable global file system is one of the main things we will have to settle on to expand further.

The Intel and Intel-compatible platform appears to be the most economical choice for commodity computing for the foreseeable future. Fermilab was one of the first high-energy physics laboratories to support Linux in a big way, and it has served us well. We have no doubt that it will continue to do so.

## References

- [1] Holmgren, D., *et al.* *Lattice QCD with Commodity Hardware and Software*, in proceedings of CHEP, 2000.
- [2] Petravick, D. *Fermilab Computing Division Systems for Scientific Data Storage and Movement*, in proceedings of CHEP, 2000.
- [3] Schellman, H. *et al.* *Large Scale Test of D0 Reconstruction Farm*, in proceedings of CHEP, 2000.
- [4] Sieh, C. *et al.* *Fermi Linux Documentation*, <http://www.fnal.gov/cd/unix/linux/>, 2001.
- [5] Dawson, T. *et al.* *ICABOD Documentation*, <http://home.fnal.gov/~dawson/tools/icabod/index.html>, 2000.
- [6] Levshina, T. *et al.* *NGOP Documentation*, <http://www-isd.fnal.gov/ngop/>, 2001.
- [7] Mandrichenko, I., *et al.* *FBSNG Documentation*, <http://www-isd.fnal.gov/fbsng>, 2001.
- [8] Mandrichenko, I., *et al.* *Farm Batch System and Fermi Inter-Process Communication and Synchronization Toolkit*. In proceedings of CHEP, 2000.