PC Farms For Offline Event Reconstruction at Fermilab

A. Beretvas [a], M.T. Cheng [b] P.T. Chang [b], F. Donno [c], J. Fromm [a], D. Holmgren [a], C.H. Huang [a], Y.C. Liu [b], D. Petravick [a], C. Sieh [a], K. Stox [a], P.K. Teng [b], K. Thayalan [a], S. Wolbers [a], G.P. Yeh [a], P. Yeh [b]

[a] *Fermi National Accelerator Laboratory, Batavia, Illinois 60510*

[b] *Academia Sinica, Taipei, Taiwan 11529, Republic of China*

[c] *Laboratori Nazionali di Frascati, Istituto Nazionale di Fisca Nucleare, I-00044 Frascati, Italy*

Fermilab is investigating the use of PC's for HEP computing. As a first step we have built a full offline environment under Linux on a set of Pentium (P5) and Pentium Pro (P6) machines (the "PC Farm"). The Pythia simulation has been ported to run serially and in parallel (using CPS) on the PC Farm. Fermilab software products and CDF offline packages have also been ported to Linux. Run I CDF data has been analyzed on both Linux and SGI (Irix) with essentially identical results. The performance of the system is compared to results with commercial UNIX systems.

Key words: PC; cluster; parallel computing; farm

1 Introduction/Hardware

A cluster of nine PCI-bus PC's has been purchased and installed by Fermilab for the purpose of determining the utility of such machines for various offline tasks in high energy physics. The hardware configurations of these PC's are purposely varied in order to evaluate the various options available. Six of the systems are Pentium based (P5 166 MHz), and three are Pentium Pro based (P6 200 MHz, of these one is a dual); we also have two Cyrix 6X86 processors (166 MHz) which may be swapped for Pentium processors. Six of the machines use SCSI (ultra, wide) disk interfaces, and two use EIDE interfaces. 32 MB of system memory are available on seven systems, 64 MB on the remaining two. The machines are interconnected via a fast ethernet network. Seven of the machines were assembled from components, and two assembled systems were purchased from major vendors (Dell, Micron).

The PC Farm allows us to explore the use of low cost computing hardware for HEP computing problems. Issues that are under investigation include operating systems, utilities and software products, and the possibility of building and testing larger systems for full offline event reconstruction, including parallel processing.

2 Software Platforms

To date, our work has used the Linux operating system[1]. Most of the Linux system infrastructure outside of the kernel (libraries, compilers, utilities) comes from the Free Software Foundation (GNU). Codes written in the C language are compiled using GNU CC (gcc), and Fortran codes are compiled using the g77, Microway, or Absoft compilers, or the f2c converter. We will also investigate one or more of the FreeBSD, Solaris, and Microsoft Windows NT operating systems.

Many of the software packages that physicists use as part of everyday work are available for Linux. These include TeX, editors (NEDIT, EMACS), CERNLIB, and so forth. Various pieces of the Fermilab UNIX environment have been ported as well, including UPS (a configuration and product management system) and UPD (a product distribution system). We have ported CPS (Cooperative Processes Software [2]) and PVM (Parallel Virtual Machine Package [3]), both toolkits for distributing computational tasks across multiple processors and nodes. Porting of software to Linux has in general been straightforward, and no major problems have been encountered. Useful porting advice is readily available [4],[5].

3 Physics Code

The first physics code that we have ported to Linux is the Pythia Monte Carlo, used to simulate collisions at high energy. In this simple example no detector simulation is performed. The results quoted in Table 1 refer to Higgs production at 200 TeV center of mass energy. We use CPS to run Pythia on multi-processors in parallel on the PC Farm. For comparison the same Pythia FORTRAN code has also been run on two standard systems (an IBM RS/6000-560, and an SGI Challenge with 200 MHz R4400 processors).

For Monte Carlo studies, large amounts of computing resources may be required to run event generator programs such as Pythia to generate large samples of events for precision measurements. More frequently, large computing resources are necessary for complete event simulation with programs such as GEANT, which require much more computation per event. GEANT and CDF simulation programs can also be run with Linux. PC Farms can be an excellent platform for Monte Carlo studies.

For each of the 50 million collision events recorded by CDF in the last two years, CDF Offline Data Production reconstructs from tracking chamber hits and calorimeter pulses to more readily useable information including particle energy, charged particle tracks, electron and muon candidates, and characteristics of jets of particles. The entire Offline Production code including YBOS (the CDF data management system) has been ported to Linux. Reading and writing data, as well as other simple tests have been successful. Porting of the CDF software packages to Linux will allow individual collaborators to do analysis from their own PC's. It may also be possible to use a PC Farm during data taking online to select events to record.

4 Results

Table 1 shows the results of our Pythia tests, for samples with over 500,000 events. We find that a typical event requires 28 msec on a P6 chip and 42 msec on a P5 chip. This is in agreement with the ratio of speeds given by the SPECint95 benchmark. Running Pythia with CPS on a single node, the CPU efficiency is 99%. When eight nodes (2 P6 + 6 P5) are used, a factor of 5.8 (8.6) in computing is obtained relative to a single P6 (P5) chip. We achieved a utilization of 96% for the PC farm.

Table 1. Comparison of UNIX machines and the PC Farm

Pythia Higgs production at E_{CM} = 200 TeV		
	CPU$\left(\frac{\text{msec}}{\text{events}}\right)$	MIP
Single Node (no CPS)		
P5	42.	82
P6	28.	113
IBM R6000/560	75.	39
SGI R4400/200	38.	83

The P6 chip has a processing power of 113 MIP's (VAX-11/780 equivalent) as indicated by timing studies using both our benchmark program ("Tiny") and

Pythia. For comparison, the corresponding power of the R4400 prosessor on an SGI Challenge was 83 MIP's.

Recently, we have performed tests using the CDF offline reconstruction code with CDF run I data using both Linux and SGI (Irix). Essentially identical results are obtained. The CPU time to process a single event on the R4400, P5, P6 and dual P6 are 3.02, 3.57, 2.16, and 1.11 sec. The dual P6 is found to be 1.95 times as fast as a single P6. The dual P6 has the same speed as a single SGI/R10000 and is currently about 3 better in price/performance.

5 Conclusions/Future

The CDF offline codes and Fermilab and CERN software packages have been ported to the PC. We have achieved our goal to demonstrate a full HEP analysis package running under Linux, as well as using the parallel processing software CPS. This demonstrates the capabilities of the system and gives a feel for the potential of PC farms for HEP event processing.

At a systems level, we hope to study the problem of making a useable large ensemble of these machines. Examples of usability are: finding a mechanism for omitting the VGA monitors and keyboards, getting optimal transfer rates on 100baseT networks, and integrating tape drives and large amounts of high-performance disk. We will investigate the performance and suitability of cluster nodes which are dual processor (SMP) systems.

References

[1] We installed Linux using the Slackware 96 distribution (Version 2.0 of the Kernal).

[2] M. Fausey *et al.*, "CPS User's Guide, CPS Version 2.9", Fermilab Computing Division Library GA009, June 24, 1993.

[3] Geist, Beguelin, Dongarra, Jiang, Manchek, Sunderam, "PVM: Parallel Virtual Machine", The MIT Press, 1994.

[4] M. Welsh, "Porting Applications to Linux", 1st International Dutch Linux Symposium, December 8, 1994 (http://www.cl.cam.ac.uk/~mdw24/porting-linux/paper.tex).

[5] S. Goldt *et al.*,"The Linux Programmer's Guide", Linux Documentation Project (ftp://sunsite.unc.edu/pub/Linux/docs/linux-doc-project/programmers-guide /llpg-0.4.tar.gz).