

Fermi National Accelerator Laboratory

FERMILAB-Pub-92/119

Using Multiple RISC CPUs in Parallel to Study Charm Quarks

C. Stoughton

*Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510*

D. Summers

*Department of Physics and Astronomy
University of Mississippi, Oxford, MS 38677*

April 1992

Submitted to Computers in Physics.



Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CHO3000 with the United States Department of Energy

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Using Multiple RISC CPUs in Parallel to Study Charm Quarks

by

C. Stoughton

Fermilab, Batavia, Illinois 60510 USA

and

D. J. Summers

Department of Physics and Astronomy

University of Mississippi, Oxford, MS 38677 USA

ABSTRACT

We have integrated a system of 16 RISC CPUs to help reconstruct and analyze a 1.3 Terabyte data set of 400 million high energy physics interactions. These new CPUs provided an affordable means of processing a very large data set. The data was generated using a hadron beam and a fixed target at Fermilab Experiment 769. Signals were recorded on tape from particles created in or decaying near the target and passing through a magnetic spectrometer. Because all the interactions were independent, each CPU could completely reconstruct any interaction without reference to other CPUs. Problems of this sort are ideal for multiple processors. In the offline reconstruction system, we used Exabyte 8mm video tape drives with an I/O capacity of 7 Terabytes per year and a storage capacity of 2.3 Gigabytes per tape. This reduced tape mounts to one or two per day rather than one or two per hour as would be the case with 9-track tapes. The ETHERNETTM network used to link the CPUs and has an I/O capacity of 15 Terabytes per year. The RISC CPUs came in the form of commercially supported workstations with little memory and no graphics to minimize cost. Each 25 MHz MIPS R3000 RISC CPU processed data 20 times faster than 16MHz Motorola 68020 CPUs that were also used. About 8000 hours of processing was needed to reconstruct the data set. A sample of thousands of fully reconstructed particles containing a charm quark has been produced.

I. INTRODUCTION

The computing needs of many experiments in high energy particle physics can be met using multiple CPUs working in parallel. Typical experiments record 10^6 to 10^{11} independent events. The results of the computation performed on one event do not affect other events. Therefore, it is straightforward to adapt these problems to a parallel processing environment [1].

Experiment 769 [2] at the Fermi National Accelerator Laboratory (Fermilab) studies the production of particles containing the charm quark. During this experiment, 400 million interactions were recorded at the Tagged Particle Spectrometer [3] on 9000 nine track tapes. The system of UNIX workstations that we describe here performs two compute intensive tasks on this data set. First, the event reconstruction algorithm, which requires three-fourths of a CPU second per event, reconstructs particle trajectories, momenta, and type. Second, a filtering algorithm inspects each event and retains candidates that could contain charm particles. We could have performed both tasks at once but chose instead to write all of the reconstructed events on serial media during the first pass of the data. Then, we read the reconstructed events and filtered during the second pass.

High energy particle physics is often similar to gold mining. A miner sifts through an enormous amount of rock to find specks of gold. A physicist often has to examine an enormous number of particle interactions to find rare events. In E769, a beam of 250 GeV/c pions, kaons, and protons interacted with 26 metal foils. A 9 month run produced a yield of over 2 billion interactions, of which 400 million were selected and recorded. Out of these interactions, we have been able to reconstruct thousands of particles containing the charm quark [4]. Typical decays include the two modes shown in Figure 1,

$$D^0(c\bar{u}) \rightarrow K^-(s\bar{u}) \pi^+(u\bar{d}) \text{ and} \\ D^+(c\bar{d}) \rightarrow K^-(s\bar{u}) \pi^+(u\bar{d}) \pi^+(u\bar{d}).$$

Charm particles are relatively massive and long lived. We calculated the mass from the measured four-momentum of decay particles. The long charm lifetime leads to a decay length of a few millimeters. The decay length in each event is the difference between two positions:

the location of the interaction of the beam particle with the target foil, and the location of the decay of the charm particle. If the mass and decay length were calculated as each event happened, it would be easy to select only the few thousand containing charm quarks. However, it is very expensive to calculate these quantities in real time. We found it easier to selectively record 400 million events out of 2 billion interactions with quantities constructed directly from the analog signals of the detector. This selection was based on the amount of energy that is produced transverse to the beam direction, since events containing charm quarks will have more transverse energy than the more frequent events with the lighter up, down, and strange quarks. This selection yielded 1.3 terabytes of event data, which were written on 9,000 nine track tapes with a high bandwidth data acquisition system [5]. We later copied these data to the 8mm format. The rapidly improving price/performance of some types of offline computing allowed us to play back these tapes and completely reconstruct the events.

II. SYSTEM REQUIREMENTS

We wanted to speed the extraction of charm events from this data. To do this a system of Reduced Instruction Set Computer (RISC) processors was devised. This supplemented a large number of Fermilab Advanced Computer Program Motorola 68020 microprocessors (ACP-I) [6] already shared by many Fermilab experiments. The idea was to build a dedicated system for one project which would avoid the delays and expense of a general purpose computer which could support scores of software packages and users.

An in-depth market survey was performed and then vendors were asked to bid on a fully competitive basis. This led to the purchase of one Silicon Graphics (SGI) [7] 4D/240S compute server by the Fermilab Physics Section which was used by E769 and other experiments. Three additional SGI 4D/240S compute servers dedicated to E769 were subsequently purchased. The network architecture of this system of compute servers is shown in Figure 2.

Each SGI 4D/240S utilizes four 25 MHz processors developed by MIPS Computer Systems, the MIPS R3000 RISC CPU and MIPS R3010 Floating Point Unit. The four processors

in one of the 4D/240S servers share 16MB of memory. The other three servers each have 8MB of shared local memory. A dedicated bus is used to connect processors and memory within the compute server. We tested the performance of the shared memory by measuring the performance of a task when it is running alone, and then with three other tasks on the compute server. A performance degradation of a few per cent was noticed in the first job run on the 4D/240S when three more jobs ran concurrently. Characteristics of the processor and memory systems are in Table 1. Table 2 compares its performance to Fermilab's ACP-I Motorola 68020 processors. We had nine key goals. The SGI systems solved our nine goals as follows:

1. **GOAL.** We needed enough CPU power to complete the event reconstruction in a year. Additional delays would dilute the scientific interest in the results of this data set. We needed the processing power of an additional 300 Motorola 68020 CPUs. Our goal was get this processing power and to simultaneously minimize the cost of reconstructing an event.

SOLUTION. The SGI system provided the raw CPU power that we needed to find charm quarks in a timely fashion at a price about 50 times lower than a traditional mainframe, and three times lower than the cost of additional ACP-I systems. Anything that needlessly added cost (e.g. graphics displays or extra memory) was rejected.

2. **GOAL.** Commercial availability: We wanted to minimize designing, building and maintaining the computer hardware and operating system.

SOLUTION. The SGI system was commercially supported and incurred little in the way of engineering and self-maintenance costs. A three year software and hardware maintenance contract was bundled with each compute server.

3. **GOAL.** We needed robust, optimizing FORTRAN and C compilers. It would have been difficult to translate the 58,000 lines of the E769 FORTRAN reconstruction program into assembly code by hand. The assembly code would have

been nearly impossible to maintain even if it could be produced. We also wished to avoid debugging a FORTRAN compiler.

SOLUTION. The MIPS corporation took the unusual approach of developing their high level language compilers and hardware in parallel. RISC compilers in general must be able to overlap instructions in order to take advantage of the hardware. For example, during a MIPS floating point divide 11 other instructions can also be executed [8]. MIPS brought out excellent FORTRAN and C compilers at the same time they brought out their RISC chips. SGI provided and supported versions of these compilers. FORTRAN licenses were purchased for the two compute servers that were used for software development.

4. **GOAL.** Data Throughput: Moving 1.3 terabytes of input data and an equal amount of output data to and from processors over a period of a year requires an average I/O rate of 85 kilobytes per second. The data had to be read from the input media, distributed to the processors, collected, and written to the output media.

SOLUTION. The data throughput was provided by Exabyte 8mm tape drives [9] for the input and output data streams. ETHERNETTM provided a network data path between servers. Small Computer System Interfaces (SCSI) were used to move data to and from the tape drives. The I/O rate of an Exabyte 8mm drive is 210 kByte/sec on the SGI 4D/240S, about a factor of five faster than necessary for an E769 reconstruction input tape. For a comparison of various media characteristics see Table 3. A single ETHERNETTM can move 15 terabytes per year given a rate of 0.5 MB/s. This exceeded our minimal I/O requirement by a factor of six. Typically a CPU processed a 4kb data event for $\frac{3}{4}$ of a second. During this time no I/O was required.

5. **GOAL.** Convenience: For operational simplicity, we wanted to limit tape mounting to a few short periods per day. This avoids having to run shifts 24 hours per day to load tapes.

SOLUTION. If we had used nine track tapes directly, the system of compute servers would have required over 30 tape mounts per day to process 6,000 input and output tapes over a year. To avoid this bottleneck, we used the Fermilab tape copy facility to transfer data to the 8mm media. Each 8mm tape can hold 2.3 Gigabytes of data, or up to 13 nine track tapes. This decoupled the large number of nine track tape mounts from the continuous data flow required by the compute servers and reduced the media cost for the reconstructed data by a factor of 25. We worked with SGI to adapt the Exabyte 8mm tape drives to the 4D/240S. Our data format has variable length blocks up to 65 kilobytes. Before purchasing the system we copied an E769 nine track data tape to an 8mm cartridge. By reading this tape on an SGI system we verified that the software drivers included in the operating system handle large variable length blocks. We purchased the first tape drive from SGI to ensure system integrity. The remaining drives were supplied by third party vendors.

6. **GOAL.** Software to distribute the events to CPUs and collect results: The compute tasks can be performed in parallel by having a complete copy of the reconstruction or filtering algorithm on each of many parallel CPUs. The data flow, process scheduling, and bookkeeping tasks require careful software design.

SOLUTION. The Fermilab Advanced Computer Program's Cooperative Process Software (CPS) [10] was used to distribute events and collect results. This was the first use of CPS in a physics experiment and it was well supported.

7. **GOAL.** Software to communicate with VAX/VMS computers: The Fermilab Physics Section's Local Area VAX Cluster (LAVC) is an interactive system of VAXstations which supports the general computing needs of physicists at Fermilab. The LAVC is shown in Figure 2. The FORTRAN code for our reconstruction and filtering algorithms was developed and maintained on this cluster.

SOLUTION. The SGI computers support TCP/IP communications. Multinet [11] software was in use at Fermilab, and this was installed on the VAX/VMS systems that we used, to provide them with TCP/IP communications.

8. **GOAL.** Sufficient memory for efficient CPU use: 1.3 MB of memory was needed per CPU during execution to contain the E769 reconstruction program and required data without paging to disk.

SOLUTION. A total of 8 MB of memory was put on three of the SGI computers and 16 MB on the fourth. The 8MB computers thus had 2 MB per CPU which allowed the 1.3 MB E769 reconstruction program to run without paging, even after the operating system's memory usage. Excessive memory can add substantially to the cost of a system. To minimize our memory requirements the reconstruction and filtering algorithms were run as separate passes through the data.

9. **GOAL.** We needed about 5 gigabytes of disk to store user programs and intermediate data.

SOLUTION. We wished to try to reserve the bandwidth of the SCSI buses for Exabyte 8mm tape drives. To do this we had SGI put one four-channel Enhanced Small Disk Interface (ESDI) in three of the compute servers and two in the fourth. As shown in Figure 2 all system disks are EDSI as well as seven user disks on the first compute server. Five SCSI disks are also used. All disks are 5 1/4". Many of the disks included a 5 year warranty. This greatly lowers the life cycle cost of a disk. Network File System (NFSTM) software is used to share disks between the different compute servers.

III. SOFTWARE

The compute servers ran the IRIX operating system, which is the Silicon Graphics implementation of UNIX, based on System V.3 with BSD 4.3 enhancements. IRIX is fully

symmetric, allowing each process to run on any of the processors within one of the compute servers. The IRIX operating system managed CPU scheduling automatically.

A good FORTRAN-77 compiler was an essential software tool for our application, since the algorithms contain 58,000 lines of FORTRAN. The MIPS FORTRAN compiler is reliable, and we had little trouble porting our code from the VAX/VMS environment. The three optimization levels are -O0 (no optimization), -O1 (default, basic optimizations), and -O2 (global optimization). In Table 4 we compare the performance of one R3000 processor at these three optimization levels. The source line debugger (dbx) and profiler (pixie) utilities were useful in porting and optimizing this code.

We kept the sixteen processors in this system busy by using the CPS package mentioned above to distribute events among the processors. The structure of one job is shown in Figure 3. Each of the bubbles represents one process on a compute server. There were three different kinds of tasks: input, output, and compute. The input and output tasks were each carried out by a single process. The compute task was either the event reconstruction algorithm or the event filtering algorithm. This compute-intensive part of the application was carried out by sixteen processes distributed among the four compute servers. One additional process, the job manager, synchronized the work of these processes.

The input process read the data stream from tape and sent blocks of events to each compute process. It then waited until one of the compute processes became available. The compute process received the event block, performed the event reconstruction or event filtering algorithm to generate an output event block, sent this block to the output process, and then signaled the input process that it was available. The output process polled the compute processes, received data from each process as it became available, and wrote it to tape or disk.

To keep the system busy full time while we were running the reconstruction algorithm, we used two pairs of tape drives. One pair had the input and output tapes for the active job, and the second pair of drives contained the tapes for the next job. The physicist on shift who determined which data to process and monitored the progress of the event reconstruction needed to attend to the system only two or three times daily. This avoided

the costs associated with 24-hour operator coverage, and kept the system utilization above 95%, comparable to batch mainframe utilization. System failures were infrequent and caused no appreciable loss of time.

After the event reconstruction was completed we used the same system to reduce the data sample from 400 million reconstructed events to the few thousand events containing charm quarks. Filtering algorithms select events with charm quarks and reject events that do not contain charm quarks. We performed this filtering in two passes. In the first pass we reduced the data set with a general filter algorithm. It selected events containing a pair of particle tracks intersecting at a point downstream of the primary interaction point. This first stage reduced the number of events by a factor of 15, leaving just under 30 million events for the second filtering pass.

The most significant difference between the event reconstruction task and the filtering task was the number of compute cycles required for each event. The characteristics of the two jobs are summarized in Table 5. For the filtering task, we were not able to keep all sixteen of the processors busy with a single input data stream, since the I/O rate is limited to 210 kByte/sec for a single Exabyte 8mm tape drive. We split the filtering task into two concurrent data streams, using each stream to feed eight processors. The ETHERNETTM connection between servers was able to handle the aggregate bandwidth of 292 kByte/sec during filtering.

A second, and related, difference was with tape mounts. Since each 8mm cartridge was filtered in three hours (instead of the ten hours required for event reconstruction) we used additional tape drives to hold the tapes waiting for execution. With all the tape drives loaded, the system ran the filtering task unattended for over 12 hours. A final difference was with the output data stream. Since the amount of data was reduced by a factor of 15 in the filtering task, we were able to write the output to separate disk files. Once a day, these files were copied to an output tape to free up disk space.

After the first stage of filtering the reduced data set fit on 23 8mm cartridges. The next stage of data filtering used criteria specific to the different charm particles and decay modes that we are studying. We ran different filtering algorithms on the reduced data set

to extract the final event samples. For these final filtering stages we have been using an automatic tape loader [12] which has two Exabyte 8mm tape drives and slots for 54 8mm cartridges in a carousel. These stages of the data filtering algorithms also ran in the CPS environment. The automatic tape mounts allowed us to scan and filter the complete data set without intervention. Different teams of physicists have developed, tested, and run new filtering algorithms on the complete filtered data set in a matter of days.

IV. CONCLUSIONS

One year after the arrival of the 16 RISC CPUs in July 1989, the reconstruction of the 400 million E769 events was completed, and the physics analysis [4] of thousands of particles with charm quarks was well under way. This was the first time UNIX/RISC computers were used for such a large data set in High Energy Physics. The affordable compute power in workstations could be exploited because our processing can be broken down into tasks which process independent events. The I/O bandwidth of ETHERNETTM was sufficient to distribute events. The operational expense of running shifts 24 hours a day to load tapes was avoided by using Exabytes.

The 58,000 line program used to reconstruct E769 data is now serving as a benchmark to track the rapidly improving cost performance of workstations (see Table 6). Given this rapid progress, it is often prudent to buy a system *just in time* and bring it on-line quickly.

V. ACKNOWLEDGEMENTS

We thank D. Green, L. Lederman, C. Kastner, S. Bracker, C. Johnstone, M. Isely, J. Pfister, R. Sidwell, L. Lueking, M. Streetman, G. Luste, B. Yeltsin, and P. Karchin for their contributions to this effort. The majority of this work was performed at the Fermi National Accelerator Laboratory, which is operated by Universities Research Association, Inc., under contract DE-AC02-76CHO3000 with the U.S. Department of Energy. This work was also supported by the U. S. Department of Energy (DE-FG05-91ER40622) and the National Research Council of Canada.

REFERENCES

- TM ETHERNET is a trademark of the Xerox Corporation. NFS is a trademark of SUN Microsystems.
- [1] Pioneering work in the use of multiple processors has been done by the Fermilab ACP group [6] and also at SLAC (see Paul F. Kunz et al., Experience using the 168/E Microprocessor for Off-line Data Analysis, IEEE Trans. Nucl. Sci. 27 (1980) 582).
- [2] E769 Collaborators. CBPF-Rio de Janeiro, Fermilab, Mississippi, Northeastern, Toronto, Tufts, Wisconsin, and Yale.
- [3] B. H. Denby et al. (E516), Inelastic and Elastic Photoproduction of $J/\psi(3097)$, Phys. Rev. Lett. 52 (1984) 795;
J. R. Raab et al. (E691), Measurement of the D^+ , D^0 , and D_s^+ Lifetimes, Phys. Rev. D37 (1988) 2391;
J. C. Anjos et al. (E691), Measurement of D_s^+ Decays and Cabibbo-Suppressed D^+ Decays, Phys. Rev. Lett. 60 (1988) 897;
J. C. Anjos et al. (E691), A Study of $D^0-\bar{D}^0$ Mixing from Fermilab E-691, Phys. Rev. Lett. 60 (1988) 1239;
J. C. Anjos et al. (E691), Measurement of the Λ_c^+ Lifetime, Phys. Rev. Lett. 60 (1988) 1379;
J. C. Anjos et al. (E691), Measurement of the Form-Factors in the Decay, $D^+ \rightarrow \bar{K}^{*0} e^+ \nu_e$, Phys. Rev. Lett. 65 (1990) 2630.
- [4] D. Errede et al., Hadroproduction of Charm at Fermilab E769, 25th International Conf. on High Energy Physics, Singapore, 2-8 Aug. 1990.

- [5] C. Gay and S. Bracker, IEEE Trans. Nucl. Sci. 34 (1987) 870;
 R. Vignani et al., IEEE Trans. Nucl. Sci. 34 (1987) 756;
 S. Hansen et al., IEEE Trans. Nucl. Sci. 34 (1987) 1003;
 Mark Bennett et al., IEEE Trans. Nucl. Sci. 34 (1987) 1047;
 R. Hance et al., IEEE Trans. Nucl. Sci. 34 (1987) 878.
- [6] I. Gaines et al., The ACP Multiprocessor System at Fermilab, Proc. of the International Conf. on Computing in High Energy Physics, Asilomar, USA (2-6 Feb. 1987) 323; FERMILAB-Conf-87/21.
 J. Biel et al., Software for the ACP Multiprocessor System, Proc. of the International Conf. on Computing in High Energy Physics, Asilomar, USA (2-6 Feb. 1987) 331; FERMILAB-Conf-87/22.
- [7] Silicon Graphics (SGI), 2011 N Shoreline, Mountain View, CA 94039
 Digital Equipment (DEC), 146 Main St., Maynard, MA 01754
 Sony Microsystems, 645 River Oaks Parkway, San Jose, CA 95134
 MIPS Computer Systems, 950 DeGuigne Dr., Sunnyvale, CA 94086
 Hewlett-Packard/Apollo, 3000 Hanover St., Palo Alto, CA 94304
 IBM, Research Park, Suite 106, Starkville, MS 39759
 SUN Microsystems, 2550 Garcia Ave., Mountain View, CA 94043
 Cray Research, 655 Lone Oak Dr., Eagan, MN 55121
- [8] Gerry Kane, MIPS RISC Architecture, Prentice Hall (1989) 7-12.
- [9] Exabyte Corp., 1745 38th Street, Boulder, CO 80301.
- [10] Joe Biel, Mike Isely, Mark Fishler, Chip Kaliher and Matt Fausey, ACP Cooperative Processes User's Manual, Fermilab GA0006, 19 Nov. 1990.
- [11] TGV Inc., 603 Mission St., Santa Cruz, CA 95060
- [12] IGM Data Autoloader Division, 4041 Home Rd., Bellingham, WA 98226
- [13] System Performance Evaluation Cooperative (SPEC) c/o Waterside Associates, 39510 Paseo Padre Parkway, Suite 350, Fremont, CA 94538.

TABLES

Processor:	Central Processor	R3000
	FP Processor	R3010
	FP Data Format	IEEE 754, 32- and 64-bit formats
	Registers	32 CPU, 32 FP (single precision)
	Word Length	32 bits
	Clock Speed	25 MHz
Cache Memory:	Cache Type	Write-back
	Cache Size	64 KB instruction
		64 KB Data (1st level)
		256 KB Data (2nd level)
	Write Buffer	4 words deep
	Read Buffer	16 words deep
Processor Bus BW	64 MB/sec sustained	
CPU Memory:	Size	8MB (16 MB on one server)
	Maximum allowed	128 MB
	Bandwidth	64 MB/sec sustained
Virtual Memory:		2 GB per process
MPLink Bus:	Width	32 bit address; 64 bit data
	Bandwidth	64 MB/sec sustained

Table 1. Characteristics of Silicon Graphics 4D/240S processors. Each SGI 4D/240S contains four R3000 central processing units and four R3010 floating point units with local caches and shared main memory.

	CPU Maker	CPU/FPU	Clock MHz	Instruction Cache	Data Cache	E769 sec/event
ACP-I	Motorola	68020/68881	16	256 bytes	0	18.01
ACP-I	Motorola	68020/68882	16	256 bytes	0	15.32
SGI 4D/240S	MIPS	R3000/R3010	25	64kb	320kb	0.76

Table 2. A comparison of SGI 4D/240S and Fermilab ACP I computers. The time shown is that required to run the E769 particle reconstruction benchmark on one CPU. The Motorola 68882 is a faster version of the Motorola 68881 floating point chip. Concurrent execution of some floating point instructions are allowed in the 68882 and not allowed in the 68881.

Tape Type	Length	Capacity	\$/tape	\$/Terabyte	Tapes/ Terabyte
8mm Video	106m	2.3 GB	\$4.25	\$ 1848	435
4mm DAT	60m	1.2 GB	\$7.79	\$ 6492	833
IBM 3480	165m	0.22 GB	\$4.60	\$20909	4545
9-track	732m	0.16 GB	\$9.31	\$58188	6250

Table 3. A Comparison of Storage Media. The 8mm, 9-track, and 3480 tape prices are from the Fermilab stockroom catalog. The 4mm DAT price is from the New York Times, 20 Jan. 1991, page 31.

Optimization Level:	Time:
-O0 (no optimization)	1.45 seconds/E769 event
-O1 (default optimization)	1.09 seconds/E769 event
-O2 (global optimization)	0.76 seconds/E769 event

Table 4. MIPS R3000 at different optimization levels. The time shown is that required to run the E769 particle reconstruction benchmark on a single CPU of an SGI 4D/240 computer.

Processing Task	Reconstruction	Filtering
CPU Time/Event	0.76 seconds	0.07 seconds
# Input Events	400×10^6	400×10^6
Average Input Event Size	3.2 kByte	1.2 kByte
# Processors per Data Stream	16	8
Input Bandwidth	67 kByte/sec	137 kByte/sec
# Output Events	400×10^6	27×10^6
Average Output Event Size	1.2 kByte	1.2 kByte
Output Bandwidth	25 kByte/sec	9 kByte/sec
# Concurrent Data Streams	1	2
Aggregate Bandwidth	92 kByte/sec	292 kByte/sec

Table 5. Comparison of event reconstruction and filtering tasks. The throughput requirements are more demanding for the filtering task, so we used two independent data streams to keep the entire system efficiently utilized.

COMPUTER [7]	CPU Type	CPU Clock MHz	Cache Memory I+D+2nd kByte	Write Through Buffer Depth	Main Memory Speed. MB/sec	SPEC-marks [13]	E769 sec/event	E769 Normalized to 25MHz
SGI Crimson S	R4000	50	8+8+1024	N/A	400	70.		
SGI 4D/35S	R3000	36	64+64	11	97	31.	0.53	0.76
SGI INDIGO	R3000	33	32+32	11	91	26.	0.60	0.79
SGI 4D/25S	R3000	20	64+32	1	7	14.	1.07	0.86
SGI 4D/20	R3000	12	16+ 8	1	7	8.	1.94	0.93
DEC 3100	R2000	16	64+64	4	13	11.7	1.10	0.73
DEC 5000-200	R3000	25	64+64	6	100	23.5	0.66	0.66
DEC 5000-120	R3000	20	64+64	1	50	16.4	0.91	0.73
DEC 5000-25	R3000	25	64+64	1	50	19.1		
Sony 3710	R3000	20	64+64	1		12.6	1.00	0.80
MIPS Magnum	R3000	33	32+32	8	133	25.1		
HP/Apollo 705	PA-RISC	35	32+64	N/A		34.		
HP/Apollo 720	PA-RISC	50	128+256	N/A	400	59.5	0.39	0.78
HP/Apollo 730	PA-RISC	66	128+256	N/A	528	76.8		
IBM 6000-320	IBM	20	8+32	N/A	160	32.8	0.82	0.66
IBM 6000-320H	IBM	25	8+32	N/A	200	41.2		
SUN 2	Sparc	40	64	N/A	49	24.7	0.72	1.15
SUN ELC	Sparc	33	64	N/A	41	20.1		
Cray Y-MP	Cray	166	0	N/A	4200	142.9	0.32	2.13

Table 6. Comparison of Current Workstations. The Cray figures are for a single CPU. Only the MIPS R2000 and R3000 use a write through cache architecture. The fifth column does not apply to the other CPU types. The last column shows how long it would take to reconstruct an E769 benchmark event if the computer clock speed were *scaled* to 25 MHz.

FIGURES

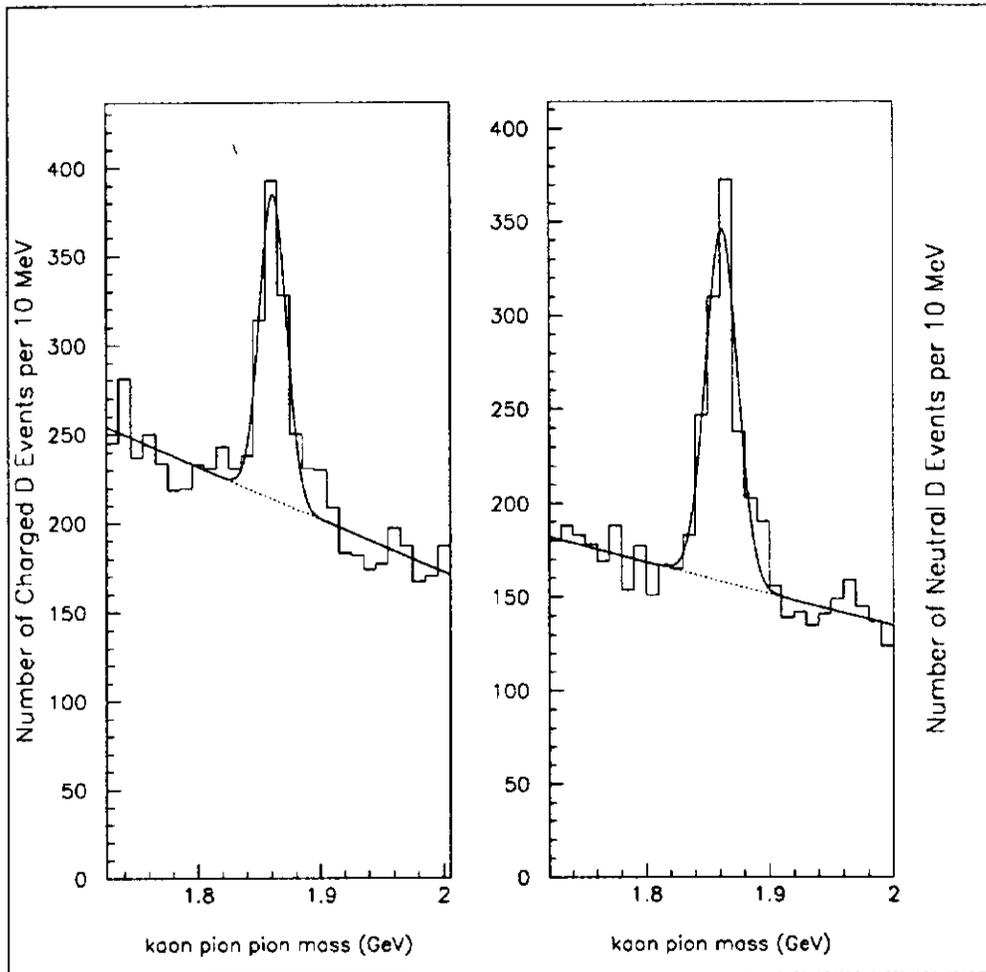


Figure 1. Charm Meson Signals. $D^+(1.869) \rightarrow K^-\pi^+\pi^+$ and $D^0(1.865) \rightarrow K^-\pi^+$.

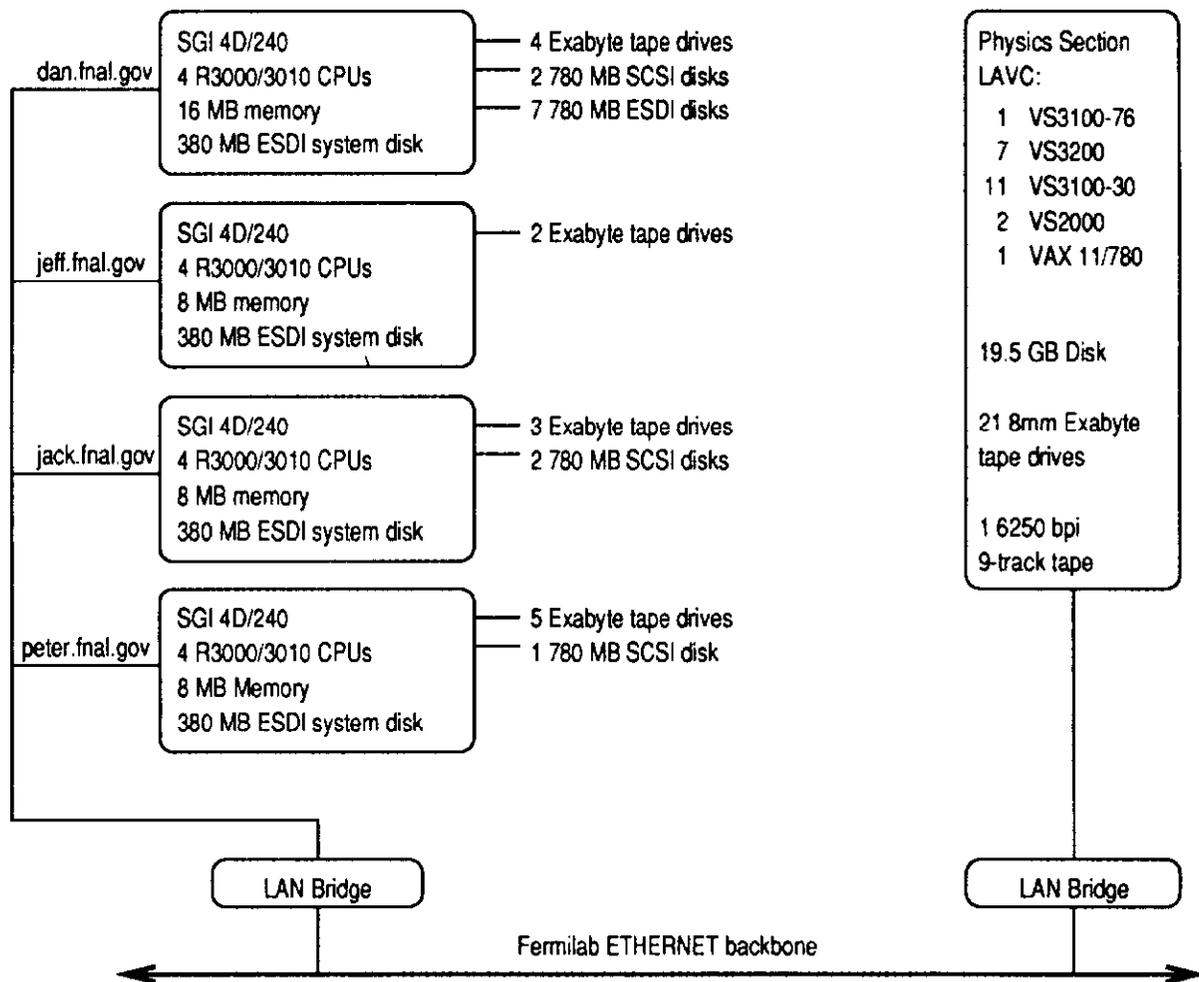


Figure 2. Network Configuration for Compute Servers and LAVC. ETHERNETTM was used to connect four SGI 4D/240S compute servers and a Local Area VAX cluster of VAXstations. The VAXstations have 8-user VAX/VMS licenses.

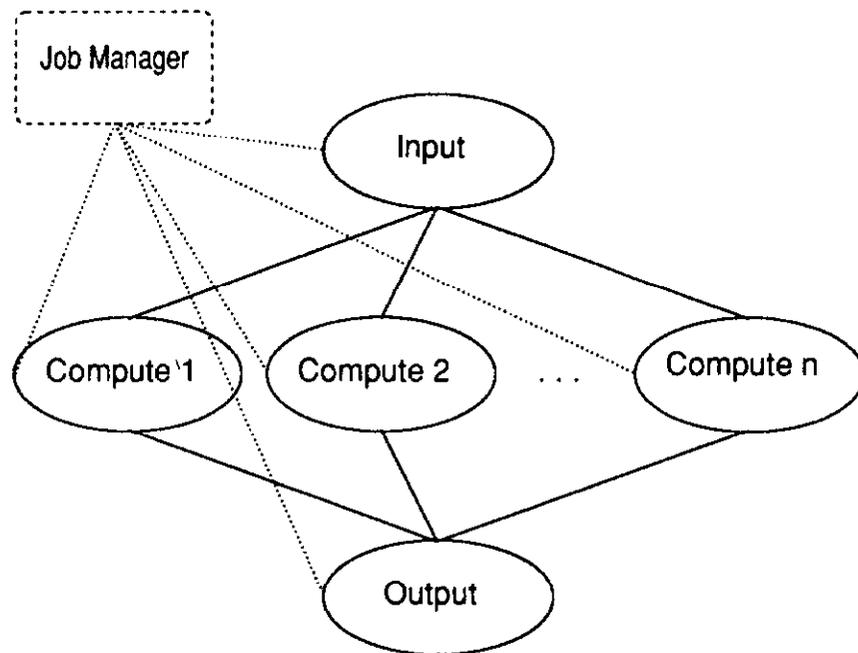


Figure 3. Structure of a CPS job. Fermilab's Cooperative Process Software [10] was used to distribute independent events from tape to CPUs which processed the data. CPS was then used to gather results together.