SSC PC-002
27 PAGE

Development of a RISC/UNIX Multiprocessing System for SSC Design and Simulation

Submitted By

Paul Avery, Andrew White, James K. Walker, John Yelton

University of Florida

Gainesville, FL 32611

UNIVERSITY OF FLORIDA

SEND NOTICE OF AWARD TO:
The University of Florida
Division of Sponsored Research
219 Grinter Hall
Gainesville, FL 32611
(904) 392-1582

SPONSORED PROJECTS APPROVAL FORM

AGENCY APPLICATION DEADLINE
(DO NOT LEAVE BLANK)

Date: 10/2/89

☐ **postmark** X☒ **receipt**

University Project # 89092922
(LEAVE BLANK)

Title of Proposal: Development of a RISC/UNIX Multi Processing System for SSC Design and Simulation

Submitted to Agency/Program: U.S. DOE/SSC Division
(NOTE TO THE P.I.: Please provide mailing instructions on page 2)

UNIVERSITY ENDORSEMENTS: The attached proposal has been examined by the officials whose signatures appear below. The principal academic review of the proposal is the responsibility of the **Department/Center and College.** If additional space is needed for signatures, please provide them on a separate sheet of paper.

Principal Investigator: (Project Director)

Paul Avery

NAME: Paul R. Avery — Date
TITLE: Assistant Professor
CAMPUS ADDRESS: 215 Williamson Hall
TELEPHONE: 904/392-9264
SOC. SEC. NO. 123-44-5842

Approval by Dean or Director (If more than one)

9/29/89

NAME: Willard W. Harrison — Date
TITLE: Dean, College of Liberal Arts and Sciences

Co-Principal Investigator: (If Applicable)

Andrew P. White 9.28.89

NAME: Andrew P. White — Date
TITLE: Research Scientist
TELEPHONE: 904/392-9237
SOC. SEC. NO. 593-36-2477

Other Endorsement (If Needed):

NAME: — Date
TITLE:

Department Head:

N S Sullivan 9-28-89

NAME: Neil S. Sullivan — Date
TITLE: Chairman and Professor Department of Physics

Approval by Vice-President for Agricultural Affairs (For all projects involving IFAS Personnel)

NAME: — Date
TITLE:

Department Head: (If more than one)

NAME: — Date
TITLE:

Approval by Vice-President for Health Affairs: (For all projects involving JHMHC Personnel)

NAME: — Date
TITLE:

Approval by Dean or Director:

NAME: — Date
TITLE:

Official Authorized to Sign for the University: (Leave Blank)

Dillard C. Marshall 9/29/89

NAME: Dillard C. Marshall — Date
TITLE: Assistant Director of Research
Division of Sponsored Research
University of Florida

DSR-1 (6/88)

OMB Approval No. 0348-0006

FEDERAL ASSISTANCE

1. TYPE OF SUBMISSION (Mark appropriate box)
- ☐ NOTICE OF INTENT (OPTIONAL)
- ☐ PREAPPLICATION
- ☒ APPLICATION

2. APPLICANT'S APPLICATION IDENTIFIER
a. NUMBER: 89092922
b. DATE (Year month day): 19 89 9 29

3. STATE APPLICATION IDENTIFIER (NOTE: TO BE ASSIGNED BY STATE)
a. NUMBER:
b. DATE ASSIGNED (Year month day): 19

Leave Blank

4. LEGAL APPLICANT/RECIPIENT
a. Applicant Name: University of Florida
b. Organization Unit: Division of Sponsored Research
c. Street/P.O. Box: 233 Grinter Hall
d. City: Gainesville
e. County: Alachua
f. State: Florida
g. ZIP Code: 32611
h. Contact Person (Name & Telephone No.): Paul Avery 904/392-9264

5. EMPLOYER IDENTIFICATION NUMBER (EIN): 1-596001874-C7

6. PROGRAM (From CFDA)
a. NUMBER: ___ . ___
MULTIPLE ☐
b. TITLE: Basic Energy Sciences

SECTION I—APPLICANT/RECIPIENT DATA

7. TITLE OF APPLICANT'S PROJECT (Use section IV of this form to provide a summary description of the project.)

Development of a RISC/UNIX Multi Processing System for SSC Design and Simulation

8. TYPE OF APPLICANT/RECIPIENT
A—State; B—Interstate; C—Substate Organization; D—County; E—City; F—School District; G—Special Purpose District; H—Community Action Agency; I—Higher Educational Institution; J—Indian Tribe; K—Other (Specify):
Enter appropriate letter: T

9. AREA OF PROJECT IMPACT (Names of cities, counties, states, etc.)
Nation Wide

10. ESTIMATED NUMBER OF PERSONS BENEFITING

11. TYPE OF ASSISTANCE
A—Basic Grant; B—Supplemental Grant; C—Loan; D—Insurance; E—Other
Enter appropriate letter(s): A

12. PROPOSED FUNDING

a. FEDERAL	$ 1,446,150.00
APPLICANT	.00
STATE	.00
d. LOCAL	.00
e. OTHER	.00
f. Total	$ .00

13. CONGRESSIONAL DISTRICTS OF:
a. APPLICANT: 6
b. PROJECT: 6

14. TYPE OF APPLICATION
A—New; B—Renewal; C—Revision; D—Continuation; E—Augmentation
Enter appropriate letter: A

15. PROJECT START DATE (Year month day): 19 90 01 01

16. PROJECT DURATION: 36 Months

17. TYPE OF CHANGE (For 14c or 14e)
A—Increase Dollars; B—Decrease Dollars; C—Increase Duration; D—Decrease Duration; E—Cancellation; F—Other (Specify):
Enter appropriate letter(s):

18. DATE DUE TO FEDERAL AGENCY ► (Year month day): 19

19. FEDERAL AGENCY TO RECEIVE REQUEST: U. S. Department of Energy
a. ORGANIZATIONAL UNIT (IF APPROPRIATE): SSC Divisionn
b. ADMINISTRATIVE CONTACT (IF KNOWN): Dr. Tom Dombeck
c. ADDRESS: ER-24, SSC Division, U.S. Department of Energy GTN, Washington, DC 20544

20. EXISTING FEDERAL GRANT IDENTIFICATION NUMBER

21. REMARKS ADDED ☐ Yes ☐ No

SECTION II—CERTIFICATION

22. THE APPLICANT CERTIFIES THAT► To the best of my knowledge and belief, data in this preapplication/application are true and correct, the document has been duly authorized by the governing body of the applicant and the applicant will comply with the attached assurances if the assistance is approved.
a. YES, THIS NOTICE OF INTENT/PREAPPLICATION/APPLICATION WAS MADE AVAILABLE TO THE STATE EXECUTIVE ORDER 12372 PROCESS FOR REVIEW ON: DATE _____
b. NO, PROGRAM IS NOT COVERED BY E.O. 12372 ☒
OR PROGRAM HAS NOT BEEN SELECTED BY STATE FOR REVIEW ☐

23. CERTIFYING REPRESENTATIVE
a. TYPED NAME AND TITLE: Dillard C. Marshall, Assistant Director of Research
b. SIGNATURE: Dillard C. Marshall

SECTION III—FEDERAL AGENCY ACTION

24. APPLICATION RECEIVED (Year month day): 19

25. FEDERAL APPLICATION IDENTIFICATION NUMBER

26. FEDERAL GRANT IDENTIFICATION

27. ACTION TAKEN
- ☐ a. AWARDED
- ☐ b. REJECTED
- ☐ c. RETURNED FOR AMENDMENT
- ☐ d. RETURNED FOR E.O. 12372 SUBMISSION BY APPLICANT TO STATE
- ☐ e. DEFERRED
- ☐ f. WITHDRAWN

28. FUNDING

a. FEDERAL	$ .00
b. APPLICANT	.00
c. STATE	.00
d. LOCAL	.00
e. OTHER	.00
f. TOTAL	$ .00

29. ACTION DATE► (Year month day): 19

30. STARTING DATE (Year month date): 19

31. CONTACT FOR ADDITIONAL INFORMATION (Name and telephone number)

32. ENDING DATE (Year month date): 19

33. REMARKS ADDED ☐ Yes ☐ No

Development of a RISC/UNIX Multiprocessing system for SSC Design and Simulation

Abstract

We propose a two phase program to develop a powerful multiprocessor computer system for use by the US solenoidal detector collaboration in carrying out detector simulations. Phase I will be a low cost 100 MIPS system based on DECstation 3100 computing nodes which will allow university groups to participate in SSC design and simulation work. The UFMulti control package developed at the University of Florida will allow any simulation job to use the full processing power of this facility. Our intent is that the Phase I system serve as a model for other university groups wishing to perform significant SSC work on computing facilities in their home institutions. Phase II will be an expansion of the RISC facility to the level of 500 MIPS to allow full scale SSC detector simulations, physics and trigger studies to be carried out by the collaboration.

I Introduction

Experience with the development of detectors for hadron colliders has shown that design and simulation are very demanding in terms of computer resources. The full simulation of a single event can take several hours on the processors available today. This problem will certainly be much more severe at SSC energies and, indeed, is already seen in the approximations and compromises that have had to be made in recent SSC detector and physics studies. The development and use of adequate computing facilities for SSC design and simulation work is the subject of this proposal. University groups, in particular, are restricted in their ability to contribute to this work through lack of sufficient computing power.

(a) The computing need

Three main tasks may be identified for the SSC design phase: physics event generation, detector simulation, and analysis/reconstruction of the simulated events. All of these tasks require the usual support of efficient code development and testing facilities, which do not

impose much additional burden on a system but must be provided in a way that does not produce hidden costs in time and manpower. The importance of a broad involvement of the high energy community in these main design areas has been stressed many times. It is very desirable to have a wide range of intellectual input and to provide, either locally or by network access, sufficient computing power for university groups to make effective contributions to SSC detector design.

Workshops[1,2] studying the necessary level of SSC computing resources have reported that an aggregate of several thousand MIPs (1 MIP = 1 VAX780) will be required for the design phase. We believe that a large fraction of this aggregate can be very effectively provided by the use of the multiple processor "farms" we propose to develop. The use of aggregate numbers from workshops tends to ignore the distribution of physicists and their need for access to computing facilities on an activity which may only occupy a limited amount of their time. The possibility of installing powerful computing facilities at relatively low cost in high energy groups will encourage more active participation in SSC studies and detector design, even on a part time basis. It is also clear that the earlier these facilities can be made available, and more physicists become involved in design studies, the more effective will be the final design of the detectors.

Some of the computing problems facing those designing for the SSC have already been encountered during the design of the current generation of hadron collider detectors. The solutions that have emerged are rather different than might have been expected. For instance, the use of vectorization on supercomputers has seen only limited application to high energy problems, whereas the present trend towards "farm"-like systems of multiple, small, powerful processors has seen a rapid increase. The latter trend has come about largely because of the intense commercial competition in the workstation market. There is no doubt that high energy physics has benefitted significantly from a situation in which individual processors increase in power by about a factor of two every 1-2 years, while their cost increases only slowly. The next stage of these dramatic developments for high energy physics is widely expected to be a move towards the use of reduced instruction set (RISC) processors. There is now an opportunity for the high energy community to take advantage of this situation to provide the necessary computing power for SSC detector design in a very cost effective manner. Accordingly, we propose to undertake the setting up, and development of a farm of commercial RISC processors,and its use in SSC studies in a manner similar to the successful development of our present VMS farm for our D0 and CLEO activities.

(b) **Outline of our proposal**

We propose a two phase program. Phase I would build on the work performed in setting up our present VAX/VMS farm, and provide a dedicated facility having a computing capability of approximately 100 MIPS for development of SSC simulation code and physics studies. This system would be used by a broad collaboration of US and other high energy groups developing the design of a general purpose solenoidal detector for high Pt physics at the SSC. Phase II of our program would expand the initial system into a 500 MIP facility for use in comprehensive and detailed detector simulation by the same consortium of users. It should be emphasized that Phase I will result in a highly cost-effective system that can be emulated by other HEP groups, to whom we will make the system control software freely available. Additionally, the facility planned in Phase II would be available to collaborators outside our group, via network access through DECnet and NSFnet.

II SSC Computing Needs for Design and Simulation

We first examine in detail the various tasks to be performed to support the design of SSC detectors. We estimate, using previous studies and our own experience, the quantitative requirements in each area and how this translates into the specification of a local computing system for SSC studies. In the following subsections we use 100 MIPs as a representative level of computing power. As will be seen, this level will allow our group to carry out its initial contribution to simulation development and physics studies. In addition, 100 MIPs also represents a reasonable fraction of the second phase facility to which single user might have access.

(a) **Physics simulation**

The time required to generate a typical SSC event in ISAJET is O(1) sec. on a 3 MIP processor. The most demanding studies are those where some new, and usually rare, physics is looked at in order to assess its discoverability at the SSC. The generation of sets of signal events is usually not demanding since a limited number of events can be used to establish characteristic signatures. The background calculation, however, can present a major computing task since a useful number of events must remain after multiple cuts on the data in order to have some confidence in the estimate of the background level. Current and past studies have generated $O(10^6)$ events so we shall use this number to illustrate the required level of computing. This number of events could be generated overnight on a 100 MIP facility . Each such event can have O(1000) particles yielding a say 2×10^4 bytes/event. 10^6 events would

then require 20 Gigabytes of storage space. This is about ten video cartridges. Alternatively, a summary tape can be written keeping only the essential features (jets, leptons,...) as extracted from, say, a simple detector simulation, plus a smaller sample of full event output. This is only one example, but we believe it is not unrepresentative since, for example, precision studies using frequently occuring events would require large signal samples and less background - still requiring the generation of large Monte Carlo datasets.

To illustrate the power of the multiprocessor approach, we show in Fig. * the result of a calculation of the QCD background to the missing transverse energy signal for supersymmetric particles at 2 TeV in a simple representation of the D0 detector. The sample of 5×10^5 events (equivalent to 3.6×10^6 events due to a preselection of heavy quarks before fragmentation) was generated in 10 hours using 40 MIPs. The line shape of the QCD background is clearly established out to large values of missing transverse energy, in contrast to earlier studies that ran out of statistics around 120 GeV/c.

We intend to use the power of the proposed facility to both continue our supersymmetry studies and investigate other physics processes at very high statistics.

(b) **Detector Simulation**

Many of the studies of the interaction between SSC physics processes and potential detectors have been carried out by groups at workshops using simple, idealized detector simulations. This is usually due to the limited time and computing resources available, and the lack of manpower to implement a more detailed simulation. However, the importance of a detailed simulation is well established from experience with present experiments. The realities of engineering and cost inevitably result in compromises in detector design. These compromises translate into such features as cracks, dead material and regions of reduced efficiency which, in turn, impact the ability to study physics with the detector.

The decision to use the GEANT package for SSC simulation has already been made. However, the price of using such a comprehensive program as GEANT is seen in the time required to fully simulate one event. The D0 experiment has a detailed simulation of its detector implemented using GEANT. It currently takes about 1 hour on a 3 MIP processor to simulate a 100 GeV/c transverse momentum QCD event with e.m. shower parametrization below 200 MeV. This represents about 10 sec/GeV of energy deposited in the detector. For an SSC event with, say, 10 TeV energy deposited 10^5 MIP-sec. would be required. This implies

10^3 sec./event on a 100 MIP system, or a week of running for a 1000 event sample. This sample size is at a useful level for the study of detailed detector features. Many studies for D0 have been carried out at this level of statistics for fully simulated events.

It is possible to achieve a faster simulation by the use of techniques such as "frozen showers" libraries, but care must be taken particularly in the study of rarely occuring detector phenomena and the tails of distributions. Furthermore the use of very large shower libraries presents a heavy I/O load, and the performance of such a system in a farm environment has yet to be fully investigated. Nevertheless, there certainly will be situations where the speed-up (about a factor of 50 for D0) gained from the shower library use will be very useful.

The size of simulated events is driven primarily by the channel count for calorimetry and by the event multiplicity for the tracking chambers. Both factors will increase for the SSC. Present D0 simulated event sizes are about 1 Mbyte, so SSC simulated event sizes of several Mbytes must be anticipated. A 1000 event sample will therefore occupy several Gbytes, or a few video cartridges. At this level it is reasonable to keep a few 1000 event datasets on disk for rapid and frequent analysis during an active study. This argues for a mass storage size of at least 10-20 Gbytes per 100 MIPs of processing power.

Program memory size is potentially a significant issue. For D0 we have found that, while the total size is now in the region of 16 Mbytes with all subdetectors "live", there is a kernel of code at the level of a few Mbytes that occupies the majority of time during the simulation of an event. We find that our version of GEANT can be run on an 8 Mbyte processor with a low paging rate (about 1 page/sec.). This is an area that needs to be watched closely as simulation program sizes grow during development and it is probably wise to install a minimum of 16 Mbytes per processor for SSC work.

(c) **Reconstruction and analysis of simulated events**

It has been estimated[2] that the reconstruction/analysis time of an SSC event will take about 1200 MIP-sec. Of course, a full reconstruction is not always needed for every study, but this estimate will provide a useful upper limit. A sample of 1000 events will take a few hours to fully process on a 100 MIP facility. Typically we then expect the analysis time to range from a small fraction of an hour for simple histogramming of the "data" from a simulation, to the upper limit of a few hours. Since the reconstruction software will take years to fully develop, early studies will be at the lower end of this range. It is to be expected that the analysis time of

the later, more developed studies will be reduced as more powerful processors become available.

(d) Summary

There is no doubt that SSC studies of the interaction of physics and detectors is demanding in terms of computing power. A system that will be capable of producing a useful level of output for SSC simulation and design work should have at least the following characteristics:

- a total of 100 MIPs cpu power
- at least 16 Mbytes of memory per processor
- 10-20 Gbytes of mass storage

A final facility that is to be used for a number of simultaneous studies should have a characteristic size of several times this minimum system.

III The Role of Computer Farms

(a) General comments

It is now widely agreed that the only cost effective way of handling the SSC computing load is by partitioning the events into many individual streams, each of which is digested by a powerful microprocessor. A massively parallel system of this type is sometimes called a "computer farm", although individual systems under development differ considerably in the way the processing nodes are organized (shared memory, fast data buses, hypercube topologies, integrated I/O links, etc.). Some better known examples include (1) the 168/E SLAC system used by LASS (the very first farm); (2) the ACP effort at Fermilab; (3) the Brown University VAX based system; (4) the IBM 3081/E system being developed in Europe; (5) the British Transputer project; and (6) the University of Florida VAX/VMS system. The Transputer project, as well as some other vendor based systems being developed, offers a somewhat more integrated hardware environment than the others.

(b) Vendor based farms

The cost effectiveness of large parallel systems results directly from the recent availability of powerful commercial processors. Moreover, the competition in workstations which employ these processors has forced manufacturers to adopt standard operating systems (UNIX),

language compilers (Fortran, C), communications hardware (Ethernet) and software (TCP/IP, NFS, DECnet) and peripheral interfaces (SMD, SCSI, ESDI). This trend towards standards is crucial to two of the most challenging aspects of designing SSC experiments: software development and verification. Both tasks can be considerably simplified if codes are run on vendor supplied hardware using commercially available operating systems and languages.

These considerations strongly support our argument that large multiprocessing farms should be built whenever possible using commercial hardware and software. Such systems offer the advantages of (1) Availability of outside maintenance; (2) Wider availability of standard peripherals; (3) Better debugging tools and documentation; (4) Less need for experts. These points are especially relevant for small university groups having limited economic resources and personnel.

(c) The role of UNIX

The second generation of processors now reaching the marketplace (MIPS R3000, Intel 80860, Sun SPARC) are all being used in UNIX workstations. We anticipate that the proven cost effectiveness of these systems, plus the move to UNIX by major vendors such as DEC and IBM, will cause the HEP community to shift more and more to UNIX and that by the start of SSC the primary computing load will be borne by UNIX based machines. This trend should benefit small university based groups since it will allow them to acquire commercial UNIX computers to build their own computing farms. We aim through this proposal to develop a farm based on UNIX machines as a model for such groups. This is discussed further in the next section.

In the area of application software, we shall take as much advantage as possible of implementations of HEP packages for UNIX systems. For example, our colleagues on CLEO have recently ported the CERNLIB routines to the DECstation 3100 so a large amount of simulation code is now available in UNIX. CERN also is porting many of its application packages to UNIX machines and we have access to these through Cornell.

IV Our Proposal: A UNIX Multiprocessor

(a) The proposal

We propose to build a UNIX multiprocessing system in two phases. The Phase I system would be based on the new RISC/UNIX workstations from DEC and have a total computing

capability of about 100 MIPS. The facility will be driven by the UFMulti software package so that individual jobs will have access to the entire system. It will be used by the recently formed US based collaboration for developing a solenoidal detector for high Pt physics. Even at this early stage, a 100 MIPS system is necessary for doing the requisite background calculations documented in the second section of this proposal. Part of our purpose in developing the Phase I system is to provide a model for other university based computing groups that cannot afford the large mainframe computing facilities needed for SSC work.

Phase II would expand this facility into a 500 MIPS system for performing large scale detector simulations needed for detailed work. It would consist of processors having approximately twice the computing capability of the DECstation 3100 (about 25 in all). Like the Phase I system, it would also be used by outside collaborators involved in SSC detector work and be capable of multiprocessing.

(b) **Phase I description**

The Phase I system is a 100 MIPS facility that consists of the following pieces

- DECstation 3100 server (32 Mbytes)
- 10 DECstation 3100 computing nodes (16 Mbytes)
- 2 VAXstation 3100 workstations (16 Mbytes) for code development
- 20 Gbytes of disk,
- 5 Exabyte 8mm tape drives
- Disk controllers and other I/O devices.
- 1 Physicist/programmer
- 1 Graduate assistant

The VAX3100 workstations will be connected to our VAX cluster to permit code to be developed in a VMS environment. Code development can also be done on the RISC server.

The Physicist/programmer and graduate assistant are needed to oversee the system and act as consultants for outside users. The demand for this facility will no doubt require the services of a dedicated full-time person (plus assistant) in addition to the UF HEP group. We wish to retain flexibility in the topological arrangement of the hardware elements - such as the number of nodes/Ethernet and the grouping of disks on the server vs. local disks on the nodes. We emphasize that 16 MByte memory size is necessary because of the expected large size of SSC events in simulations. Similarly, the amount of disk required reflects the size of SSC events and is commensurate with the amount of computing power requested (we have 11 Gbytes on our

cluster which has a total computing capability of 50 MIPS). The dimensions of the system are at a level that will permit us to demonstrate its effectiveness for SSC work while not overinvesting in processors whose power will double in the coming year with small increase in cost.

The SSC system will be physically connected to our existing VAX cluster (described in Section IV) by Ethernet and communicate via the DECnet and TCP/IP protocols. Access to the outside world will be provided by our leased line to Fermilab and by NSFnet. Our Fermilab line runs DECnet over X.25 packets and has a speed of 9.6Kb. We are also a node on NSFnet, which is supported by the University for the supercomputer users and is connected to the Pittsburgh Supercomputer Center (and thus to the rest of the computing community) by a 56Kb line. This link will be upgraded in 1990 to T1 status (1.3 Mb) and should support multiple remote users with excellent response. We currently use both links to communicate with the facilities at Cornell.

The Phase I system will use the UFMulti software (see a description in Section IV and in the accompanying preprint) which now works on VAX/VMS systems and DECstations. This software will permit a user to devote the entire computing capability of the RISC farm to a single job with minimal coding changes. The utility of the software has already been proven in CLEO analysis (the charmed baryon and $\Upsilon \rightarrow D^{*+}X$ projects), D0 Monte Carlo generation with GEANT (overnight runs of 100 events generated with 30 MIPS) and SSC physics studies done at UF (10^6 events generated). A user will have the option of starting the job either from the VAX cluster or from the DECstation server.

We chose to use DEC equipment for the Phase I system for several reasons. First, the cost effectiveness of DEC's current RISC products is very good and is likely to remain so for the forseeable future. DEC is a major vendor and its RISC products are based around industry standard peripheral interfaces (for disk and tape) for which it is easy to buy from third party manufacturers. Second, DEC is attempting to integrate Ultrix and VMS machines into a harmonious network. Given that the industry appears to be making a strategic turn to UNIX, it makes sense for small groups to stay in a familiar environment while the vendor provides the transition from VMS to UNIX (the connection between the two is not entirely satisfactory at present). Finally, we have a research agreement with DEC (described in Section IV) which allows us a 50% discount off all hardware and software products sold by DEC.

In addition to physics studies, we want to further develop our UFMulti multiprocessing software and understand how well a system of this kind can serve as a model for university based SSC groups. Areas of study include:

1. Determining the conditions which cause the linear speedup with the number of processors to break down (e.g., data I/O bandwidth, swapping, etc.).
2. Improving the I/O throughput to the computing nodes. We want to investigate issues such as local disk vs. disks attached to servers, new communication protocols and new types of I/O interfaces.
3. Development of efficient communication strategies between VMS and UNIX. This is important for systems with a VMS server and UNIX computing nodes which can be dominated by the time it takes to convert the data from VMS to UNIX format.
4. Use of commercial Remote Procedure Call (RPC) packages to simplify the design of the Interprocess Communication package and make the supervisory software more transparent to the user.
5. Tuning the operating system parameters to prevent large SSC simulations from swapping too rapidly.

(b) **Phase II description**

The Phase II system is a 500 MIPS facility which will be used for large scale detector simulations, especially those involving showers, for the solenoidal collaboration. It would be acquired approximately one to one and a half years after the Phase I system. We can only estimate the precise configuration of such a system because it will no doubt utilize hardware that has not been announced yet. The basic components of the system are listed below.

- 2 extra 20 MIPS servers (32 Mbytes). Also used for code development.
- 500 MIPS of computing nodes (16 - 32 Mbytes)
- 2 more VAX 3100 workstations (or equivalent) for code development
- 50 Gbytes of disk
- Many tape drives (8 mm?)
- A second Physicist/Programmer
- A second graduate assistant

We expect, projecting from current trends, that the 500 MIPS of farm based computing will provided in chunks of 20-25 MIPS or more costing about $300-$400/MIP. It is too early to tell which vendor we would choose. The machines in Phase II would also be connected to the cluster to permit outside users to easily access the system.

The Phase II system would, like its predecessor, be used by the large US based collaboration to carry out realistic detector simulations of SSC events. The large number of MIPS is driven both by the increased realism of the simulations that will be performed (realistic = slow) and by the increased number of users who will need to do simulations. We are including extra servers to accommodate users who will want to develop their code directly on a UNIX based machine. The extra staff will be needed to develop support software (as well as acquiring software from CERN, SLAC, FNAL, etc.) for the users. They will also carry out simulations directly at the request of the collaboration.

(c) **Our qualifications**

The University of Florida HEP group is particularly qualified to carry out the work described in this proposal. We have worked on the multiprocessing problem since 1986 and have built a computer farm based around VAX/VMS and DECstation equipment at UF along with a sophisticated software package called UFMulti to drive it (a description can be found in the accompanying preprint of a talk presented at the 1989 Oxford Computing Conference). The software can be applied to any system of VAXes or DECstations on a local area network (LAN), making it a useful tool for small HEP groups which have most of their computing power in the form of workstations. CLEO is using UFMulti for their ULTRIX/VMS farm and some member institutions of the D0 collaboration have indicated that they likely will adopt it as well. It is now being used at UF to generate a large number of GEANT simulated events for D0.

The UFMulti software is still being developed at UF by the HEP group with the help of 3 students and some technical assistance from DEC. We have recently succeeded in getting the system to work on DECstations and further work is under way to improve the use of CPU nodes and disk drives in a multiple user environment. We are doing this in conjunction with the Cornell computer group. We have also benefitted greatly from discussions with other HEP groups developing multiprocessor capabilities including the Fermilab ACP group, Brown University and the ALEPH group at Barcelona.

The computer system at UF is a VAX cluster used by the CLEO and D0 groups and the high energy theorists. It consists of the following pieces:

- VAX6220 computer (2 CPUs), 64 MB
- 5 VAX3200 workstations, 8MB, diskless
- 1 DECstation 3100 workstation, 8MB, 300 MB disk

- 8 VAX3200 computing nodes (multiprocessor), 8MB, diskless
- 4 VAX3100 workstations, 8MB, diskless
- 2 RA82 + 8 RA90 disks (11 GB total)
- 1 TU81 tape drive (+ 1 older drive
- 2 Exabyte 8 mm tape drives

All the above equipment was acquired from Digital at a 50% discount based on a Research Agreement signed with them in 1988. (The Research Agreement gives us special access to DEC engineers and will allow us to beta test some of their new RISC products in our multiprocessing configuration. In return, we provide DEC with performance characteristics of their machines when run in the multiprocessing mode. Cornell has a complementary agreement with DEC.) Funding is provided by a matching arrangement with the U.S. Department of Energy and the University of Florida for a total amount of $600K provided over a four year period.

The total computing power available in the current system is approximately 50 VAX 780 equivalents. This computing power is needed to handle our data analysis and simulation work in the CLEO and D0 experiments, simulation studies we are performing for the BCD proposal at Fermilab and SSC Monte Carlo work for a SUSY study.

V Budget Discussion

Phase I

We are requesting to purchase and develop a multiprocessing farm based on DECstations. The equipment includes:

- DECstation 3100 server (24 Mbytes, 3 x 332 Mbyte disks)
- 10 DECsystem 3100 computing nodes (16 MByte)
- 2 VAXstation 3100 workstations (16 Mbyte)
- 20 disk drives (5 1/4", 1.0 Gbyte apiece)
- 10 disk controllers
- 5 Exabyte 8mm tape drives
- I/O peripherals such as Ethernet controllers, fiber optic controllers

In addition, we would like to be able to purchase for evaluation purposes promising new software and I/O peripherals to enhance the system. We cannot be any more specific about the nature of these items, but we feel that $20,000 should be adequate to give us the required flexibility.

We are asking for $60,000 + overhead per year to support one Physicist/programmer and a graduate assistant during Phase I. These persons would be responsible for maintaining the SSC RISC farm and, in addition, would help outside users carry out their simulations on the system.

Phase II

We are asking for the equipment and people necessary to bring the Phase I system to Phase II level. Many of the equipment costs are unknown because they involve unannounced machines. The numbers shown in the budget reflect best estimates based on current price/performance trends. We are also requesting a second Physicist/programmer and assistant for further support for this facility.

In the second year of Phase II (Year 3) we are requesting equipment to further develop the system. This might take the form of more computing nodes, more disk and tape, better I/O facilities, etc. The precise amount can only be guessed at this time.

V Budget

Phase I (Year 1)

The prices below reflect our 50% discount on DEC products.

1 DECstation 3100 server	$ 25,000
10 DECstation 3100 computing nodes	$ 90,000
2 VAXstation 3100 workstations + local disk	$ 16,000
20 Wren VII disks (or equivalent) + controllers	$100,000
5 Exabyte 8mm tape drives	$ 20,000
4 Ethernet controllers	$ 8,000
Miscellaneous software and I/O devices	$ 20,000
Maintenance (10% of cost)	$ 28,000
1 graduate assistant	$ 10,000
1 Physicist/programmer + fringe benefits	$ 50,000
Total	$367,000
Overhead (45%) on personnel, maintenance	$ 39,600
Total Phase I (Year 1)	$406,600

Phase II (Year 2)

The following costs are only approximate since they involve unannounced equipment. The purchases shown below will raise the Phase I system to the required Phase II level.

2 Servers (20-25 MIPS apiece, 32 Mbytes)	$ 60,000
400 MIPS in Computing nodes (16-32 Mbytes) $300/MIP	$120,000
30 Gbytes disk + controllers ($4/Mbyte)	$120,000
10 8mm drives	$ 40,000
Maintenance on Year 1 equipment	$ 28,000
Maintenance on Year 2 equipment	$ 34,000
2 graduate assitants	$ 20,000
2 Physicist/programmers + benefits	$100,000
Total	$522,000
Overhead (45%) on personnel, maintenance	$ 81,900
Total Phase II (Year 2)	$603,900

Phase II (Year 3)

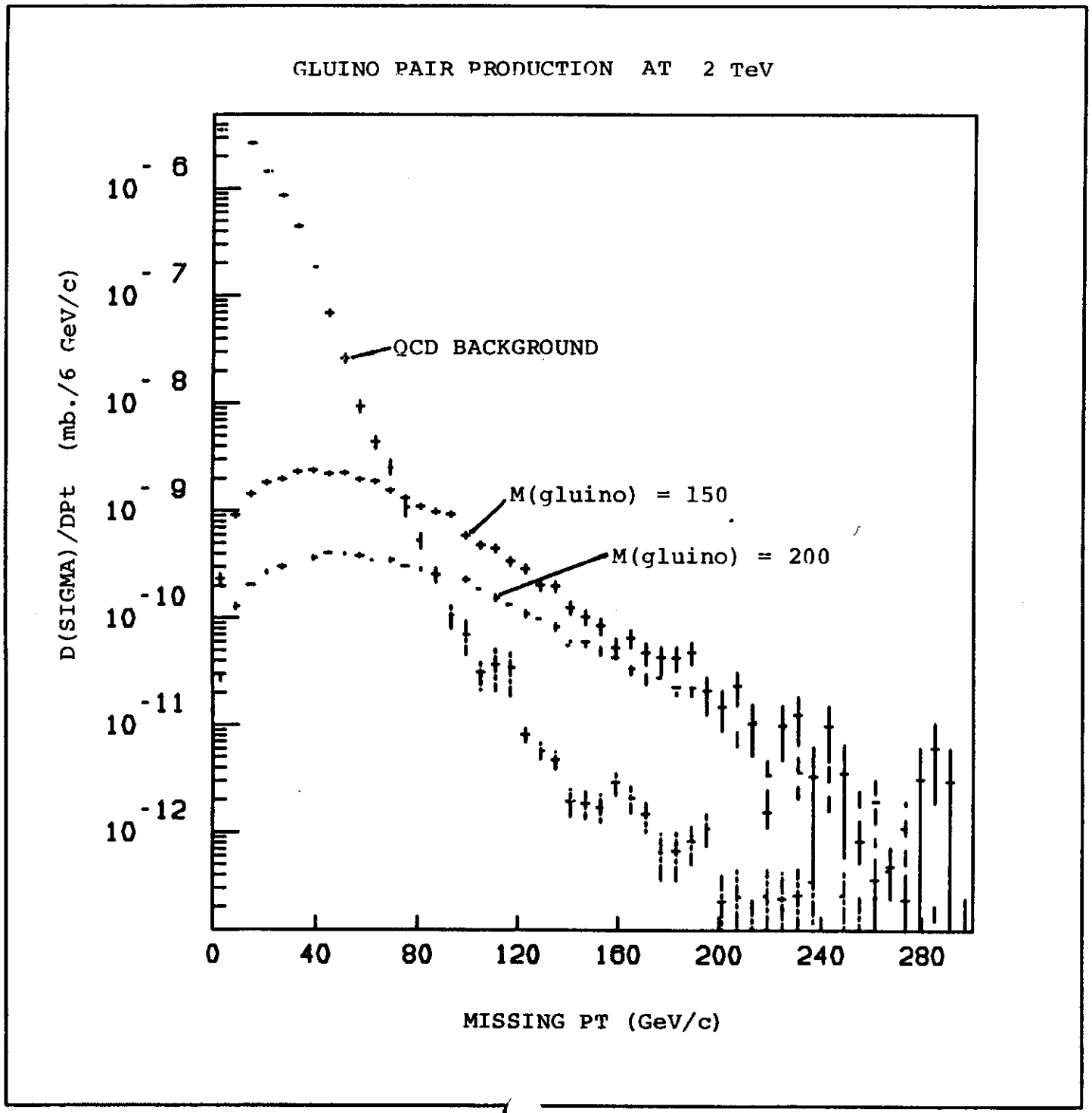
The following costs are only approximate since they involve unannounced equipment.

Unknown new hardware (for further development)	$150,000
Maintenance on Year 1 equipment	$ 28,000
Maintenance on Year 2 equipment	$ 34,000
Maintenance on Year 3 equipment	$ 15,000
2 graduate assistants	$ 20,000
Two Physicist/programmers + benefits	$100,000
Total	$347,000
Overhead (45%) on personnel, maintenance	$ 88,650
Total Phase II (Year 3)	$435,650

VI References

1. Proceedings of the 1986 Summer Study on the Physics of the Superconducting Supercollider. p479. R. Donaldson, J. Marx eds.

2. Proceedings of the Fermilab Workshop on Triggering and Data Acquisition. 1985.


GLUINO PAIR PRODUCTION AT 2 TeV
D(SIGMA)/DPt (mb./6 GeV/c)
10^-6
10^-7
10^-8
10^-9
10^-10
10^-11
10^-12
QCD BACKGROUND
M(gluino) = 150
M(gluino) = 200
0
40
80
120
160
200
240
280
MISSING PT (GeV/c)

ER F 4620.1
(7-85)

U.S. Department of Energy

Grant Application Budget Period Summary

(See Reverse for Definitions and Instructions)

OMB Approval
No. 1910-1400

Please Print or Type

Organization: University of Florida	Period Covering: From: 12/01/89 To: 12/01/90	FOR DOE USE ONLY Proposal No: Award No.:
Principal Investigator (P.I.)/Project Director (P.D.): Paul R. Avery		

A. SENIOR PERSONNEL PI/PD Co PIs, Faculty and Other Senior Associates (List each separately with title, A.6 show number in brackets. Attach separate sheet, if required.)	DOE Funded Person-Mos. Cal.	Acad.	Sumr.	Funds Requested By Applicant $
1. Programmer	12			38,532
2.				
3.				
4.				
5.				
6. (1) TOTAL SENIOR PERSONNEL				
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)				
1. ( ) POST DOCTORAL ASSOCIATES				
2. ( ) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				
3. (1) GRADUATE STUDENTS	12			10,000
4. ( ) UNDERGRADUATE STUDENTS				
5. ( ) SECRETARIAL-CLERICAL				
6. ( ) OTHER				
TOTAL SALARIES AND WAGES (A + B)				48,532
FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) 24% of A1 + 2220/yr				11,468
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				60,000
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM) (see budget explanation pages in text)				
TOTAL EQUIPMENT				279,000
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				0
2. FOREIGN				0
F. OTHER DIRECT COSTS				
1. MATERIALS AND SUPPLIES				
2. PUBLICATION COSTS/PAGE CHARGES				
3. CONSULTANT SERVICES				
4. COMPUTER (ADPE) SERVICES				
5. CONTRACTS AND SUBGRANTS				
6. OTHER (computer maintenance)				28,000
TOTAL OTHER DIRECT COSTS				28,000
G. TOTAL DIRECT COSTS (A THROUGH F)				367,000
H. INDIRECT COSTS (SPECIFY RATE AND BASE)				
TOTAL INDIRECT COSTS 45% of all except D				39,600
I. TOTAL DIRECT AND INDIRECT COSTS (G & H)				406,600
J. APPLICANT'S COST SHARING (IF ANY)				0
K. TOTAL AMOUNT OF THIS REQUEST (ITEM I LESS ITEM J)				406,600

PI/PD TYPED NAME & SIGNATURE Paul R. Avery *Paul Avery*	DATE 9/29/89
INST. REP. TYPED NAME & SIGNATURE Dillard C. Marshall *Dillard C. Marshall*	DATE 9/29/89

ER F 4620.1
(7-85)

U.S. Department of Energy

Grant Application Budget Period Summary

(See Reverse for Definitions and Instructions)

OMB Approval
No. 1910-1400

Please Print or Type

Organization: University of Florida	Period Covering: From: 12/01/90 To: 12/01/91	FOR DOE USE ONLY Proposal No: Award No.:
Principal Investigator (P.I.)/Project Director (P.D.): Paul R. Avery		

A. SENIOR PERSONNEL PI/PD Co PIs, Faculty and Other Senior Associates (List each separately with title, A.6 show number in brackets. Attach separate sheet, if required.)	DOE Funded Person-Mos. Cal.	Acad.	Sumr.	Funds Requested By Applicant $
1. Programmer	12			38,532
2. Programmer	12			38,532
3.				
4.				
5.				
6. (2) TOTAL SENIOR PERSONNEL				
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)				
1. ( ) POST DOCTORAL ASSOCIATES				
2. ( ) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				
3. 2) GRADUATE STUDENTS	12			20,000
4. ( ) UNDERGRADUATE STUDENTS				
5. ( ) SECRETARIAL-CLERICAL				
6. ( ) OTHER				
TOTAL SALARIES AND WAGES (A + B)				97,064
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) 24% of A1 + 2220/year each				22,936
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				120,000
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM) (see budge explanation pages)				
TOTAL EQUIPMENT				340,000
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				0
2. FOREIGN				0
F. OTHER DIRECT COSTS				
1. MATERIALS AND SUPPLIES				
2. PUBLICATION COSTS/PAGE CHARGES				
3. CONSULTANT SERVICES				
4. COMPUTER (ADPE) SERVICES				
5. CONTRACTS AND SUBGRANTS				
6. OTHER (computer maintenance)				62,000
TOTAL OTHER DIRECT COSTS				62,000
G. TOTAL DIRECT COSTS (A THROUGH F)				522,000
H. INDIRECT COSTS (SPECIFY RATE AND BASE) TOTAL INDIRECT COSTS 45% of all except D				81,900
I. TOTAL DIRECT AND INDIRECT COSTS (G & H)				603,900
J. APPLICANT'S COST SHARING (IF ANY)				0
K. TOTAL AMOUNT OF THIS REQUEST (ITEM I LESS ITEM J)				603,900

PI/PD TYPED NAME & SIGNATURE Paul R. Avery *Paul Avery*	DATE 9/29/89
INST. REP. TYPED NAME & SIGNATURE Dillard C. Marshall *Dillard C. Marshall*	DATE 9/29/89

ER F 4620.1
(7-85)

U.S. Department of Energy

Grant Application Budget Period Summary

(See Reverse for Definitions and Instructions)

OMB Approval
No. 1910-1400

Please Print or Type

Organization: University of Florida	Period Covering: From: 12/01/91 To: 12/01/92	FOR DOE USE ONLY Proposal No: Award No.:
Principal Investigator (P.I.)/Project Director (P.D.): Paul R. Avery		

A. SENIOR PERSONNEL PI/PD Co PIs, Faculty and Other Senior Associates (List each separately with title, A.6 show number in brackets. Attach separate sheet, if required.)	DOE Funded Person-Mos. Cal.	Acad.	Sumr.	Funds Requested By Applicant $
1. Programmer	12			38,532
2. ~~Programmer~~	12			38,532
3.				
4.				
5.				
6. (2) TOTAL SENIOR PERSONNEL				
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)				
1. ( ) POST DOCTORAL ASSOCIATES				
2. ( ) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				
3. 2) GRADUATE STUDENTS	12			20,000
4. ( ) UNDERGRADUATE STUDENTS				
5. ( ) SECRETARIAL-CLERICAL				
6. ( ) OTHER				
TOTAL SALARIES AND WAGES (A + B)				97,064
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS) 24% of A1 + 2220/yr each				22,936
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				120,000
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM) (see budget explanation pages)				
TOTAL EQUIPMENT				150,000
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				0
2. FOREIGN				0
F. OTHER DIRECT COSTS				
1. MATERIALS AND SUPPLIES				
2. PUBLICATION COSTS/PAGE CHARGES				
3. CONSULTANT SERVICES				
4. COMPUTER (ADPE) SERVICES				
5. CONTRACTS AND SUBGRANTS				
6. OTHER (computer maintenance)				77,000
TOTAL OTHER DIRECT COSTS				77,000
G. TOTAL DIRECT COSTS (A THROUGH F)				347,000
H. INDIRECT COSTS (SPECIFY RATE AND BASE)				
TOTAL INDIRECT COSTS 45% of all except D				88,650
I. TOTAL DIRECT AND INDIRECT COSTS (G & H)				435,650
J. APPLICANT'S COST SHARING (IF ANY)				0
K. TOTAL AMOUNT OF THIS REQUEST (ITEM I LESS ITEM J)				435,650

PI/PD TYPED NAME & SIGNATURE Paul R. Avery *Paul Avery*	DATE 9/29/89
INST. REP. TYPED NAME & SIGNATURE Dillard C. Marshall	DATE 9/29/89

ER F 4620.1
(7-85)

U.S. Department of Energy

Grant Application Budget Period Summary

(See Reverse for Definitions and Instructions)

OMB Approval
No. 1910-1400

Please Print or Type

Organization: University of Florida	Period Covering: 3 year total	FOR DOE USE ONLY
	From: 12/01/89	Proposal No:
Principal Investigator (P.I.)/Project Director (P.D.): Paul R. Avery	To: 12/01/91	Award No.:

A. SENIOR PERSONNEL PI/PD Co PIs, Faculty and Other Senior Associates (List each separately with title, A.6 show number in brackets. Attach separate sheet, if required.)	DOE Funded Person-Mos. Cal.	Acad.	Sumr.	Funds Requested By Applicant $
1. Programmer	12			115,596
2. Programmer	12			77,064
3.				
4.				
5.				
6. (2) TOTAL SENIOR PERSONNEL				
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)				
1. ( ) POST DOCTORAL ASSOCIATES				
2. ( ) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)				
3. (2) GRADUATE STUDENTS	12			50,000
4. ( ) UNDERGRADUATE STUDENTS				
5. ( ) SECRETARIAL-CLERICAL				
6. ( ) OTHER				
TOTAL SALARIES AND WAGES (A + B)				242,660
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				57,340
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				300,000
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM)				
TOTAL EQUIPMENT				769,000
E. TRAVEL 1. DOMESTIC (INCL. CANADA AND U.S. POSSESSIONS)				0
2. FOREIGN				0
F. OTHER DIRECT COSTS				
1. MATERIALS AND SUPPLIES				
2. PUBLICATION COSTS/PAGE CHARGES				
3. CONSULTANT SERVICES				
4. COMPUTER (ADPE) SERVICES				
5. CONTRACTS AND SUBGRANTS				
6. OTHER				167,000
TOTAL OTHER DIRECT COSTS				167,000
G. TOTAL DIRECT COSTS (A THROUGH F)				1,236,000
H. INDIRECT COSTS (SPECIFY RATE AND BASE)				
TOTAL INDIRECT COSTS				210,150
I. TOTAL DIRECT AND INDIRECT COSTS (G & H)				1,446,150
J. APPLICANT'S COST SHARING (IF ANY)				0
K. TOTAL AMOUNT OF THIS REQUEST (ITEM I LESS ITEM J)				1,446,150

PI/PD TYPED NAME & SIGNATURE Paul Avery	DATE 9/29/89
INST. REP. TYPED NAME & SIGNATURE Dillard C. Marshall Dillard C. Marshall	DATE 9/29/89

ER F 4620.1A
(7-85)

U.S. Department of Energy

GRANT APPLICATION
PROJECT PERIOD SUMMARY

(Must be completed for all new and renewal applications.)

Please Print or Type

OMB Approval
No. 1910-1400

Categories	01 Budget Period	02 Budget Period	03 Budget Period	04 Budget Period	05 Budget Period
A. Senior Personnel Totals	38,532.	77,064.	77,074.		
B. Other Personnel Totals	10,000.	20,000.	20,000.		
C. Fringe Benefit Totals	11,468.	22,936.	22,936.		
Total of A, B & C	60,000.	120,000.	120,000.		
D. Equipment	279,000.	340,000.	150,000.		
E. Travel 1. Domestic	0.	0.	0.		
2. Foreign	0.	0.	0.		
F. Other Direct Costs	28,000.	62,000.	77,000.		
G. Total Direct Costs	367,000.	522,000.	347,000.		
H. Total Indirect Costs	39,600.	81,900.	88,650.		
I. Total Direct & Indirect Costs	406,600.	603,900.	435,650.		
J. Applicant's Cost-Sharing (If any)	0.	0.	0.		
K. Total Amount of Request (Item I. Less Item J.)	(1)* 406,600.	(2) 603,900.	(3) 435,650.	(4)	(5)

*This should equal item K on Budget Period Summary (ER/F/4620.1)

ESTIMATE

TOTAL COST OF PROJECT

$ 1,446,150.00

(add K(1) thru (5))

UFMulti: A New Parallel Processing Software System for HEP

Paul Avery,[1] Andrew White
Department of Physics
University of Florida
Gainesville, FL 32611, USA

UFMulti is a multiprocessing software package designed for general purpose High Energy Physics applications, including physics and detector simulation, data reduction and DST physics analysis. The system is particularly well suited for installations where several workstations or computers are connected through a Local Area Network (LAN). The initial configuration of the software is currently running on VAX/VMS machines with a planned extension to ULTRIX, using the new RISC CPUs from Digital, in the near future.

1. Background

The concept of the computer "farm", in which independent computing elements run identical copies of a (more or less) complete analysis program but process different events, is by now familiar to most high energy experimentalists. Farms are generally believed to provide the most cost effective computing power for high energy physics experiments and their utility will increase with the next generation of experiments (LEP, SSC) as the number and complexity of events continues to rise.

The earliest computing nodes[1] (168/E) were built from small subprocessors plus memory and were designed to emulate the IBM instruction set, except for I/O instructions, thereby rendering them useful in a large IBM mainframe environment where events could be read from tape/disk by the IBM "host" and sent to the nodes for processing. The advent of cheap 32 bit microprocessors in the early 1980s made it possible for groups such as the Fermilab Advanced Computer Project [2] to design an entire computer on a board, which could then be put in a crate (along with its friends) and be driven by a VAX host to provide a large amount of processing power. Nowadays, groups are exploring other novel designs for the computing elements (transputers[3], for example), always with the aim of producing large amounts of low cost CPU power.

At the same time, advances in the computer industry have led to the appearance of single board computers in the form of workstations, each of which has a computing power previously only available from a large mainframe but at a fraction of the cost. Significantly, these vendor supplied systems come with industry standard operating systems, language compilers, communications facilities and peripheral interfaces. It is natural to consider combining a number of these systems together to obtain a large amount of computing power in a readily accessible form at low cost.

The importance of this development cannot be overstated. Several speakers at this conference[4] have emphasized that software development and verification are significant challenges to designing today's complex experiments. These tasks are made significantly easier

****

1./ Talk presented at the Conference on Computing in High Energy Physics, Oxford University, Oxford, England, April 10-14, 1989.

when codes are run on vendor supplied hardware using commercially available operating systems and languages. A farm built from standard vendor equipment has other advantages: (1) Availability of outside maintenance; (2) Wider availability of standard peripherals; (3) Better debugging tools and documentation; (4) Less need for experts. These points are especially important for small university groups having limited resources.

2. Hardware

The factors discussed in the previous section led us at the University of Florida to acquire a computer system and multiprocessing farm based around DEC equipment (DEC was chosen through a competitive bid). The system consists of the following pieces

- VAX6220 computer (2 CPUs), 64 MB
- 5 VAX3200 workstations, 8MB, diskless
- 8 VAX3200 computing nodes (in rack), 8MB, diskless, monitorless
- 4 VAX3100 workstations, 8MB, diskless
- 2 RA82 + 8 RA90 disks (11 GB total)
- 1 TU81 tape drive (+ 1 older drive

We will also soon be acquiring additional equipment:

- Several DECstation 3100s (R2000 processor, 10 MIPS apiece)
- Up to 10 GB of local disk for the individual farm processors
- 2 Exabyte 8 mm tape drives

All the above equipment was or will be acquired from Digital at a 50% discount based on a Research Agreement signed with them in 1988. Funding for the project is provided by a matching arrangement with the U.S. Department of Energy and the University of Florida for a total amount of $600K provided over a four year period.

The total computing power available in the current system is approximately 50 VAX 780 equivalents and by the end of the year we hope to have over 100 VAX equivalents. This large amount of computing power is needed to handle our data analysis and simulation work in the CLEO and D0 experiments, simulation studies we are performing for the BCD proposal at Fermilab and SSC Monte Carlo work. As we will show below, this entire computer system can be driven as a multiprocessor farm using the software system that is the subject of this paper.

3. UFMulti Software: Introduction

UFMulti is the name of the software system we have written to drive the DEC system described above as well as the new farm being acquired by Cornell University for the CLEO experiment. We want to emphasize from the start that the software developed by this project can be applied to any network of VAX/VMSes on a Local Area Network (LAN) and will soon be available for Ultrix machines as well.

The basic idea is that every multiprocessing job is controlled by a Task Supervisor. Several multi-cpu tasks, each having its own Task Supervisor, can be active simultaneously and can even share nodes with other tasks since the nodes are full fledged multitasking computers. Sharing of this sort prevents a short job (1 hour) from being locked out of the system by a long (1 day) job. Each node runs the full analysis program, including general I/O, although routines to sum histograms and statistics, and print summaries, must be added to the Task Supervisor.

3.1 Topologies supported by UFMulti

The way in which the analysis jobs read and write data defines the basic topology of a multi-cpu task executed under UFMulti. Two kinds of topologies, each software selectable, are supported:

1. Loosely coupled. In this mode each process reads data from disk/tape "directly", just as it would if it were running as a standalone job. The role of the Task Supervisor is to pass the names of the files/tapes to each process on request until no more data is left.
2. Host/node. A separate process is started by the Task Supervisor to pass events one at a time to each analysis subtask until the data is finished. This topology is probably familiar to most high energy experimentalists, being used in IBM emulators and in Fermilab ACP.

We believe that the loosely coupled mode is the simplest to use since only the names of files are passed to each of the subtasks. It also is the more efficient of the two in that I/O throughput as well as CPU resources can be parallelized, especially for farms having multiple paths to the data (local disk, multiple Ethernets, etc.). It is probably a good idea to partition the data into many small files to avoid situations where some fraction of the processors finish early but can do nothing while others finish their data. This partioning is relatively straightforward, since the files can be stored in designated directories and the names kept in an ordinary text file, one per line, to facilitate editing. The Task Supervisor reads this text file and passes each name to a subtask on demand until it exhausts its list of files. The analysis job may need to be modified slightly to be able to read one file after another, but this change should be straightforward.

The loosely coupled topology also simplifies the sorting of output events, since only the names of the relatively few output files must be rearranged to maintain the same order as the input events. In fact UFMulti has facilities for storing the names of any kind of output file (for example, event lists), where the user can specifiy whether or not he/she wants the filenames sorted in the same order as the input files.

3.2 How a Typical Task Would Execute under UFMulti

1. The user fills out a "configuration file" containing the number of nodes desired, which command file to execute, which data files to process, where output events should be stored, etc. The contents of the command file can optionally be placed here to avoid having to edit another file.
2. The Task Supervisor is started. It reads the configuration file, compares the number of nodes needed with its database, and assigns remote nodes to execute the job.
3. The Task Supervisor starts up the jobs, passing file names as requested by each analysis subtask.
4. When no more data is left, the Task Supervisor collects histograms and statistics and prints them in a file.
5. The Task Supervisor kills the remote jobs and terminates.

4. Components of UFMulti

It is obvious that to run multiple jobs under UFMulti more centralized control must be present in order to coordinate competition for resources. Several kinds of processes are active at any one time; they comprise the UFMulti software.

1. Monitor
   Monitor oversees the resources of UFMulti. It keeps track of what jobs are running and can supply statistics to outside users who want to know the status of the system. It also attempts to perform a crude sort of load balancing over the farm. The Monitor communicates with the multi-cpu jobs through the Task Supervisor.
2. Network Monitoring Task
   Each Network Monitoring Task (NMT) is a permanently running process on each member of the farm. The NMT's are responsible for gathering node statistics (to find out how busy it is) and for starting analysis subtasks on the node under the command of the Monitor. They only communicate with the Monitor.
3. Query
   The Query task is started up by a user to find out about the status of the system, submit a job or kill a job. Several Query tasks can be active simultaneously. Each Query task talks only to the Monitor.
4. Task Supervisor
   The Task Supervisor is responsible for executing a multi-cpu task. It only communicates with the Monitor and the user subtask.
5. User subtasks
   These are the tasks started up by the Task Supervisor. They only communicate with their Task Supervisor.

Figure 1 shows a typical UFMulti configuration with three multiprocessor jobs executing simultaneously.

Although we have tested the software so far in a VAX/VMS environment, the code for UFMulti has been written so that other operating systems, most notably UNIX, will be supported in the future. To facilitate this process, we have separated the code into five layers.

1. Code that is unique to each process (Monitor, NMT, etc).
2. System code needed by all processes in UFMulti.
3. Interprocess communication (IPC) standard interface
4. Device drivers for IPC (VMS DECnet, Ultrix DECnet, TCP/IP, etc).
5. Library of miscellaneous routines (array copying, time of day, etc.) which are used by the entire system.

The IPC system (standard interface plus drivers) is actually a complete package which can be tested and executed separately from the rest of the system. This separateness has proven useful for debugging and testing and it has allowed use to design special purpose executables which need to communicate with each other without bringing in the full UFMulti software.

We have also adopted certain coding practices to aid portability and debugging. For example, the top three layers listed above are written in Fortran 77 with the exception of long variable names and subroutine names. Only the IPC drivers and some library routines have been written in nonstandard Fortran since these are machine dependent anyway. We have also tried as much as possible to use an object oriented approach in our software design. Accordingly, each of the above layers has its own data and can only access the other layers via subprogram calls.

Some of the layers themselves are also composed of distinct objects of code and data (list processing, buffer management, data file database, etc.) which can be used by different program elements. Both of these practices are not foolproof, of course, but we expect them to save us a great deal of time when porting to other operating systems (e.g., UNIX) in the future.

5. Debugging, Error Recovery and Load Balancing

We felt that debugging and error recovery procedures are so important that provision should be made for them even in a preliminary version of the software. Debugging in UFMulti is straightforwardly invoked by setting a software switch in the configuration file read in by the Task Supervisor. One can then step through both the user subtask and the Task Supervisor simultaneously - and watch them talk to each other - using the language debugger native to the processors, even if the processors are of different types. This feature has proven to be unbelievably useful when finding errors in the subtask code.

Error recovery is more problematical because different courses of action need to be taken depending on the error. The following has been implemented for the loosely coupled topology. If the Task Supervisor detects a communication error to a subtask (possibly because the task aborted or the node failed), it immediately severs the connection to that process (the subtask is programmed to immediately abort if it loses its connection to the Task Supervisor). If a user specified switch is set, the data files already analyzed by that process are "unflagged" and are put back in the pool of files that need to be analyzed. Otherwise, only the data file currently being processed is unflagged. The former action would normally be taken if DST analysis was being performed, since only histograms are being filled. The latter action might make sense when output files are being written.

Several varieties of load balancing have been implemented in UFMulti. First of all, no node is forced to execute more than one user subtask until all nodes have been assigned. Secondly, within a given multi-cpu job, the Task Supervisor attempts to parcel out the names of data files to be analyzed in such a way that the simultaneous reading of files from the same physical disk is minimized. Finally, a data base containing a list of all nodes and disk drives and their relation to one another (filled in by the system manager) is used by the Task Supervisor to maximize the reading of files which are "closest" to a node, rather than allowing a node to read a file from an arbitrary location. This allows farms with complicated topologies (local disks, multiple Ethernets) to maximize their I/O throughput and prevent bottlenecks.

6. Current State of UFMulti: What Works Today

UFMulti is a somewhat ambitious project and so it is natural that not all the components described above have been implemented in the first release. First of all, the system has only been tested in a VAX/VMS environment and DECnet is the only protocol that has been used for interprocess communication. Second, we have only implemented the loosely coupled topology in order to get started. We prefer this topology since as mentioned before it is easier to distribute file names than events and parallel I/O paths can be used to achieve higher throughput. Third, only the Task Supervisor has been written; the Monitor, Network Monitoring Task and Query program are still being developed. However, this is sufficient to run jobs as we have demonstrated. Because of the lack of a Monitor, however, only local load balancing is performed.

We have successfully run CLEO DST, CLEO monte carlo and D0 monte carlo code in the UFMulti environment. The only real conversion effort needed for CLEO involved writing code to combine histograms and run statistics in the Task Supervisor. Aside from that, only about 20 lines of code had to be added to the CLEO DST and monte carlo drivers to make them work. No physics analysis code had to be changed by the user and the total conversion time was approximately 1 week per driver. The same time period of a week was needed to bring up the D0 GEANT monte carlo. Minor adjustments had to be made to the latter code to prevent multiple jobs from reading and writing the same initialization file simultaneously.

7. What's Next?

Our immediate goal is to provide support for ZEBRA and the HBOOK histogram package so that histograms are automatically summed and printed without requiring the user to write special code. Automatic summation of arbitrary arrays is also being added, primarily so that users can collect their own statistics.

We are also developing a procedure whereby jobs can be submitted across HEPnet using DECnet. We believe that with proper password control and other security restrictions this capability will prove to be very useful for small groups to gain direct access to the centralized computing facilities of their experiment using the same software tools they use at home.

Our long term goal is to bring UFMulti to the UNIX operating system since the computing industry appears to be moving in this direction. As a simplifying first step we have begun work to port UFMulti to Ultrix (using Ultrix DECnet for the interprocess communication protocol) to permit the use of the new RISC workstations from DEC as computing nodes. We hope that this procedure will expose any hidden VMS biases still lurking in our code. Later we will write the protocol driver for TCP/IP and attempt to run the system in the full UNIX environment.

Acknowledgments

We want to thank the Department of Energy, the University of Florida and Digital Equipment Corporation for their generous support which made this project possible.

References

1. P. Kunz, "The LASS Hardware Processor", Nucl. Instr. Meth, p. 135 (1976);
2. T. Nash et al.,"The ACP Multiprocessor System at Fermilab", talk presented at the XXIII International Conference on High Energy Physics, July 16-23, 1986, Berkeley, California. Also available as Fermalab preprint FERMILAB-conf-86/132;
3. I. Glendinning and A. Hey, "Transputer Arrays as Fortran Farms for Particle Physics", University of Southhampton (U.K.) Report SHEP 86/87-7 (1987); D. Wallace, talk presented at this conference;
4. See the talks of R. Mount, P. Kunz and others presented at this conference.

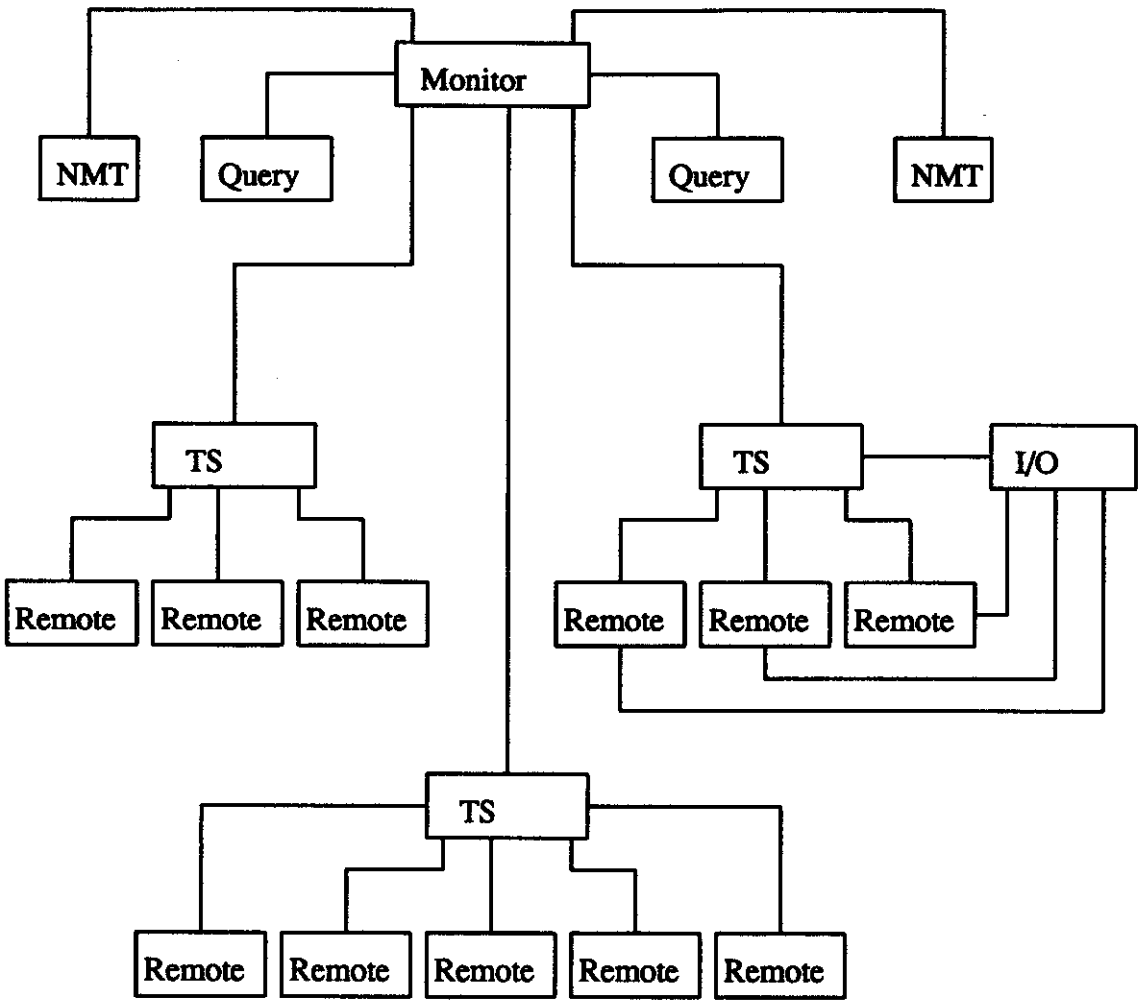


Figure 1
UFMulti software topology

Shown are three multi-cpu tasks and two users requesting statistics. One of the tasks has set up a special I/O process to send data to its remote processes. In the other two tasks the remote processes acquire data directly. The lines show the logical links over which processes communicate with each other.