Conf-930537--99

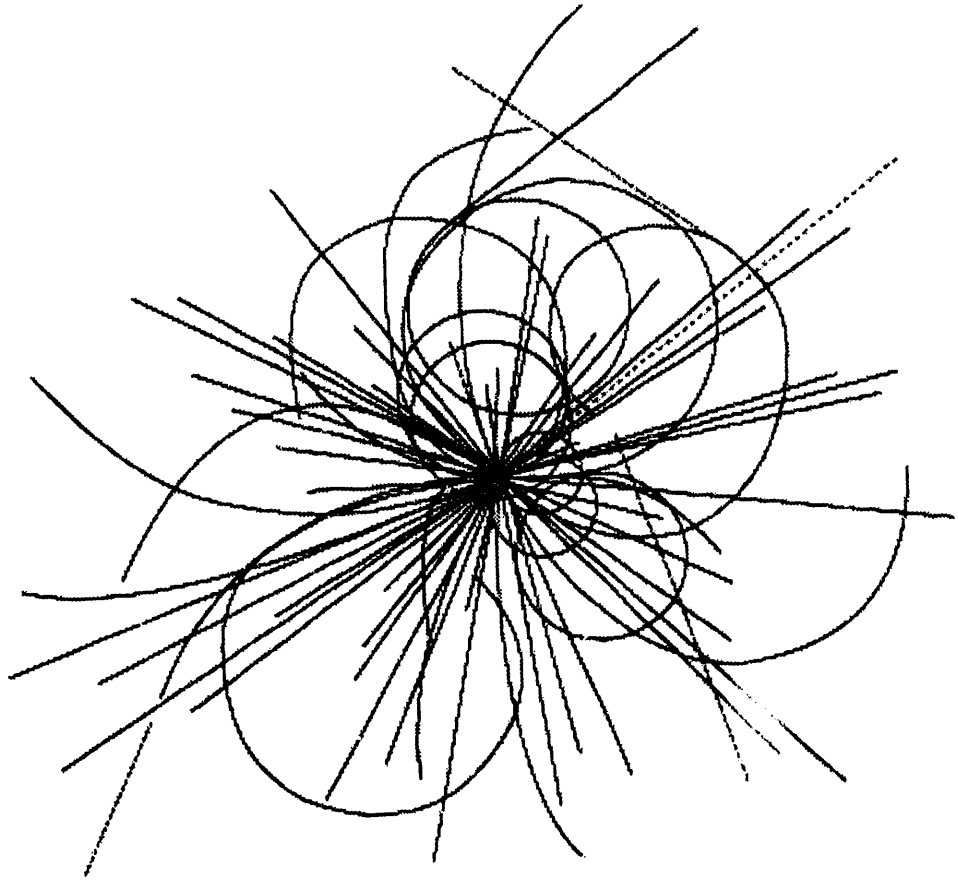
2

SSCL-Preprint-322

SSCL-Preprint-322
May 1993
Distribution Category: 400

B. Scipioni
J. Allen
C. Chang
J. Huang
J. Liu
S. Mestad
J. Pan
M. Marquez
P. Estep

Physics Detector Simulation Facility Phase II System Software Description



Superconducting Super Collider Laboratory

To be published in *Supercollider 5*

SSCL-Preprint-322

Physics Detector Simulation Facility Phase II System Software Description*

B. Scipioni, J. Allen, C. Chang, J. Huang, J. Liu,
S. Mestad, J. Pan, M. Marquez, and P. Estep

Superconducting Super Collider Laboratory†
2550 Beckleymeade Ave.
Dallas, TX 75237

May 1993

*Presented at the Fifth Annual International Symposium on the Super Collider, May 6–8, 1993 San Francisco, CA.

†Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

PHYSICS DETECTOR SIMULATION FACILITY PHASE II SYSTEM SOFTWARE DESCRIPTION

B. Scipioni, J. Allen, C. Chang, J. Huang, J. Liu, S. Mestad, J. Pan
M. Marquez, and P. Estep

Physics Computing Department
Superconducting Super Collider Laboratory*
2550 Beckleymeade Avenue
Dallas, TX 75237

ABSTRACT

This paper presents the Physics Detector Simulation Facility (PDSF) Phase II system software. A key element in the design of a distributed computing environment for the PDSF has been the separation and distribution of the major functions. The facility has been designed to support batch and interactive processing, and to incorporate the file and tape storage systems. By distributing these functions, it is often possible to provide higher throughput and resource availability. Similarly, the design is intended to exploit event-level parallelism in an open distributed environment.

INTRODUCTION

The Superconducting Super Collider Laboratory (SSCL) has adopted a computing strategy that is intended to provide the greatest amount of low cost computing power for as many users as possible.[1-5] By acquiring open systems and conforming to industry standards, the SSCL has been successful in acquiring and integrating heterogeneous networks of commercially available computers. As a result, we are able to integrate multi-vendor solutions by requiring industry standard interfaces, communication, formats, protocols, and the commonality of UNIX.

MOTIVATION

Detector simulations are characterized by loosely coupled, event parallel data runs. In order to accommodate this, the PDSF has been reconfigured. High speed FDDI networks and high disk I/O have been integrated into a highly heterogenous network of RISC based workstations. The system is composed of four groups of computers call corrals. Each corral contains an SGI 4D/360 data server, 15-16 SUN Sparc2 or HP 9000/720 compute servers with a total of approximately 140 GB of disk space, 40 GB on the corrals and 100 GB on the data servers. All machines within the PDSF are connected via FDDI rings. In addition, two database machines and two 8-mm tape robots are installed to perform database and data management functions. The architecture is represented by Figure 1.

---

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

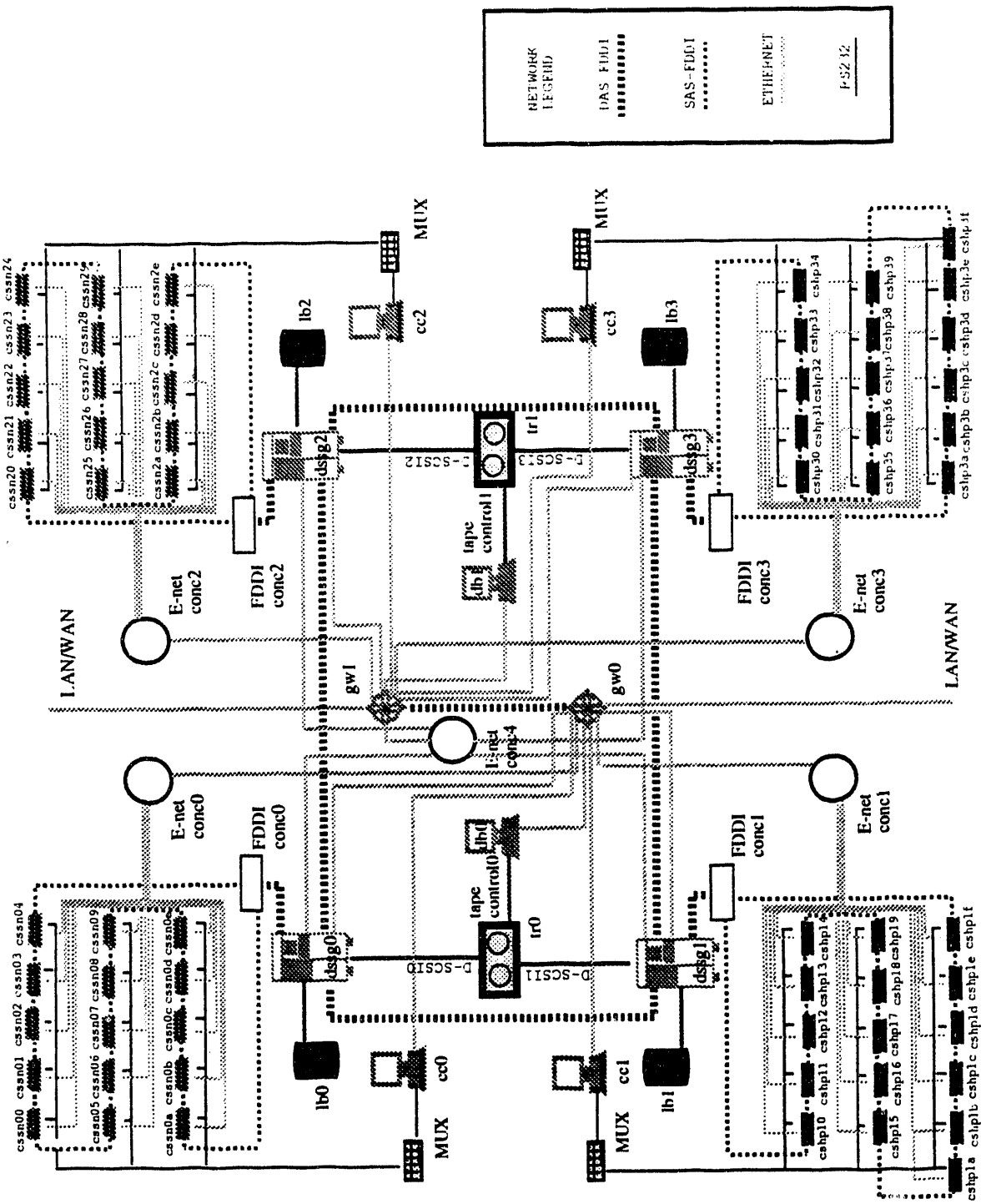


Figure 1. Physics Detector Simulation Facility (PDSF) Architecture.

With multiple gateways available, the PDSF user is allowed to choose between a SUN Sparc2 or HP 9000/720 computing corral. File systems are cross-mounted on each data server thus allowing the sharing of data. A simulation program can be executed on a single computer or fanned out to multiple compute servers.

REQUIREMENTS

The facility operational requirements were broken down into two major functional subsystems:

- Compute servers for interactive and batch usages
- Data servers containing user filesystems and supporting parallel processing

DESIGN

The system software provides the support for the various functions of the PDSF. It is the foundation upon which the interactive and batch processing is built. In addition, it has some responsibility for managing the resources of the PDSF. The system software consists of several subsystems which include the Workstation Allocation System, Console Concentrator, System Database and Polling System, System Mapping Utility, Data Management System, Robotape System, Network Queueing System, Operator Message System and Cooperative Processes Software.

Workstation Allocation System (WASH)

The function of the compute servers is to provide the user with a dedicated resource in order to give instantaneous (or near instantaneous) response to facilitate interactive use. Unfortunately, there are only a finite number of compute servers. Furthermore, there are more users than workstations. The goal of WASH is to intelligently choose a workstation for each login request in order to provide the best possible interactive environment to each user.

Users gain access to the PDSF via ***rlogin***, ***telnet***, or ***dlogin*** to one of the four Corrals. The WASH process, in turn, queries the database to determine the best machine for assignment. The database contains information concerning number of users on each workstation as well as system load per box obtained from the polling daemons. After the best machine is chosen, WASH then performs an ***rlogin*** to that machine on behalf of the user.

The Console Concentrator (CONCH)

In a distributed environment such as PDSF, there are many workstations, most of them headless (without monitor and keyboard). To manage such a configuration of headless workstations all their consoles must be directed to a single workstation with a reasonably large bitmapped display. CONCH was originally written by Neal Ziring of Washington University to run on VAX computers running 4.2 BSD flavor of UNIX. It was extensively modified at SSC. CONCH has a client-server architecture.

The server implements the hardware specific sections of the system. It listens for a connection request over RS232 lines. It also maintains a log file for each managed workstation. Any message that arrives is logged into the log-file. The server also provides a call-up service. Clients use this service to connect to a specific managed workstation. A client is a user's agent. It provides a terminal session with a managed workstation. A client is a separate process; several of them can run simultaneously.

We have combined this program with X Windows to form a more attractive operator interface. Each managed workstation is represented as an icon on a canvas. An alert feature is also provided. When any managed workstation is isolated or down, the icon turns into a different color indicating an alarmed condition.

System Database and the Polling System

The PDSF system database and the polling subsystem are responsible for providing information that is crucial for PDSF subsystems to run on the network. Systems information is gathered by the polling subsystem (SYSPOLL) and sent to the database via a SQL executor service. Information collected by the polling daemons includes running processes, user logins, disk file system usage, and system load for each workstation on the network.

PDSF subsystems, those that take care of network batch job queuing, workstation allocation, and data management, are integrated together through the system database. Each subsystem obtains a certain kind of system information from the system database: WASH monitors the system load to make login assignment decision; SYSMAP makes queries for information regarding running processes, user logins, workstation inet address, and file system usage; NQS uses workstation ID to keep track of job queues for each workstation; DMS queries the database for the status of processes.

The SYBASE database was chosen for its client/server architecture, query performance and availability on major hardware platforms. In addition to the database server software provided by

Sybase Inc., a layer of in-house developed SQL executor services is added to the PDSF database system. This structure is devised to extend database service to all workstations on the network.

The System Mapping Utility (SYSMAP)

The SYSMAP utility takes advantage of the system database to assist system administrative personnel in maintaining the system and monitoring its activities. The availability of system information in the database makes the monitoring task easier.

To facilitate administrative functionality, SYSMAP executes procedures to add/delete/ change users and groups on the PDSF system. SYSMAP modifies the UNIX system files, updating system database, broadcasting network information, and executing script to setup user directory along with initial user files.

SYSMAP offers user the option of checking on processes, user logins, system loads, and file system usage of all workstations on the network. Users may also look at the picture of a particular object in a selected range of workstations, such as which workstations a particular user has logged into. Illegitimate background jobs running on a workstation designated for interactive use can be easily spotted by using SYSMAP. SYSMAP has a X window based GUI interface designed to provide graphical display of system usage.

Data Management System (DMS)

Physics processing can be characterized by large amounts of data consumption and data generation. With a number of applications of this type running concurrently, it is clear that on-line disk storage must be supplemented by off-line tape storage. The DMS software is designed to de-emphasize the importance of tape access in physics processing. Therefore the user may concentrate on the data and not the particular storage medium on which it resides. DMS serves a two-fold purpose: to manage the means of data transfer and to provide a catalog service for the data sets transferred between disk and tape.

Robotape

The Robotape subsystem has a series of library routines, a central robotape daemon, and a daemon for each system which has juke boxes connected for use by robotape. The library routines make calls on the robotape daemon for service. The robotape daemon then passes that request to the appropriate system for execution on the proper juke box. Requests are handled concurrently and executed in parallel to the greatest extent possible without collisions.

The Network Queuing System (NQS)

NQS is a package written by Sterling Software for NASA and available from COSMIC. It is a package designed to handle batch processing on a wide variety of UNIX systems. The SSCL has made some enhancements to the software in addition to some minor bug fixes. One modification was to make use of SYBASE to store workstation information instead of NQS's NMAP facility. The other major modification made was to support load balanced queues across multiple execution systems. Some routines were also added to generate reports on the NQS accounting information.

The PDSF consists of four corrals, each of which also has a server. Each corral is divided into systems used for interactive work and systems used to run batch jobs. On each corral, there is a set of short, medium, and long queues, along with a queue to the server. These queues move a job from the interactive system to a batch system or the server for execution. When the job finishes executing, its results are moved back to the system from which it was submitted and the user can be notified of the job's completion if desired.

The Operator Message System

Because of the distributed nature of the PDSF, a simple method of sending messages and soliciting responses from the operators was needed. Programs need only make a call to the message system and indicate if a response is desired and the message system will handle the details of sending the message and returning the response string.

The message system is configured with a list of destinations to route messages. Valid destinations include printing the message, displaying the message, or E-Mailing the message. When a message enters the system, it is broadcast to all the listed destinations. If a message requires a response, the message is assigned a number. A list of unanswered messages can be requested by the operator. Responses to any outstanding message can be sent from any system in the PDSF.

Cooperative Processes Software (CPS)

The Cooperative Processing Software(CPS) is a package of tools that makes it easy to split a computational task among a set of processes distributed over one or more computers. Apart from considerations of speed, the set of processes will operate identically whether on a single computer or spread across multiple computers. Each process runs a program written by the user.

CPS is developed by Fermilab and currently supported by SSCL. The PDSF has contributed many enhancements and modifications to this package through close cooperation with Fermilab.

CONCLUSION

The system software has been fully functional on the PDSF since March, 1991 and is an effective way of integrating distributed computers. It is planned to be at full computing capacity (4000 SSCUPs) during the year of 1993. The facility will continue its primary role to support physics and detector simulations to test and verify SDC and GEM detector designs. The upgrades for PDSF in the near future will consist of adding additional data servers to provide access to tertiary storage devices as well as providing increased on-line storage and memory capacity as required by user demand.

REFERENCES

1. "Report of the task force on computing for the superconducting super collider," SSC-N-579, M. G. D. Gilchriese (editor), Dec. (1988).
2. "Report of the SSC computer planning committee," SSC-N-691. L. Price (editor), Dec. (1989).
3. L. R. Cormell, "High energy physics computing at the SSCL," presented at the *9th International Conference on Computing in High Energy Physics (CHEP91)*, Tsukuba. Japan, Mar. (1991).
4. B. A. Kinsbury, "The network queuing system," Sterling Software, 1121 San Antonio Road, Palo Alto, CA, (1986).
5. Manlio Marquez, "Physics detector simulation facility system software description," SSCL-SR-1182. Dec. (1991).

END

DATE FILMED

11/02/93