Mainframe Services from Gigabit-Networked Workstations

J-P. Baud, C. Boissat, F. Cane, F. Hemmer, E. Jagel, A. Kumar, G. Lee, B. Panzer-Steindel L. Robertson, B. Segal, A. Trannoy, I. Zacharov – CERN – Computing & Networking Division

ABSTRACT

Until recently, large mainframes and super-computers were considered essential for powerful scientific batch computing services requiring intensive tape usage, large well-managed disk storage systems, high throughput and maximum reliability. However, this situation has changed dramatically over recent years with the appearance of RISC-based workstations with performance characteristics, at least for scalar computations, comparable with the fastest mainframes but with an order of magnitude better price/performance. At the same time, competitively priced workstation-class disk and tape systems with adequate performance and reliability have become available. Combined with newly-developed LANs and Gigabit networking solutions, it is now possible to provide scalable and integrated *mainframe-class* services on workstation platforms with the UNIX operating system.

Previous papers have summarized CERN's work over the past two years in developing and introducing such services on a large scale. The latest system is called *SHIFT*, or *Scalable Heterogeneous Integrated FaciliTy*. The SHIFT facility performs a wide range of scientific data processing tasks including many with high I/O requirements and is comparable in CPU capacity to the CERN computer center. Similar systems are now being built within the budgets of smaller institutes which previously had to depend on remote university or national computing centers.

The present paper gives a short review of the SHIFT project's goals and architectural principles, and a detailed account of the networking and software design and implementation problems that were encountered and solved.

Background

The work described in this paper was initially motivated by the appearance on the market of inexpensive processors and storage systems, using technology developed for personal workstations, but which had performance characteristics comparable with those of traditional mainframes.

CERN Central Computing Environment

CERN is the European Laboratory for Particle Physics and is host to many physics collaborations using the laboratory's accelerator facilities. Physics data from the experimental particle detectors are recorded by on-line data acquisition systems and written to IBM 3480 cartridges. The data are organized in *events* where the size of an event varies from 10 KBytes to 200 KBytes. Analysis of the raw data is carried out both at the CERN computer center and at collaborating institutes throughout Europe.

At CERN, computer systems used for data analysis are benchmarked in units called *CERN CPU Units* using a representative suite of High Energy Physics codes, written in FORTRAN. For comparison, a VAX 11/780 is rated at about 0.25 CERN Unit. Note that only scalar CPU power is compared in this paper as CERN's workload is not generally vectorizable.

Currently the CERN computer center provides three mainframe services, as shown in Table 1.

Approximately 80% of the mainframe CPU goes to batch work. Most batch jobs require access to tapes. The CERN tape vault houses 150,000 tapes and cartridges with an equal number of tapes stored

Mainframe	CPU (CU)	Disk (GB)	3480 Tapes	8mm Tapes
Cray X/MP-48	32	50	6 manual 4 robotic	
IBM 9000/900	120	400	38 manual 10 robotic	8 manual
Vax 9000-410	9	50	8 manual	

Table 1: CERN – Central Mainframes

outside the vault but in active use. A robot with a capacity for 18,000 3480 cartridges handles approximately 20% of the mount requests. Round-the-clock manual mounts are the responsibility of operations staff.

Into this environment, a batch project based on RISC workstations was initiated two years ago. Beginning with a single APOLLO DN10040, the project has grown substantially and now forms an operational service which exceeds the total deliverable CPU capacity of the central mainframes. The service is collectively known as the *Centrally Operated RISC Environment* or *CORE*, and has three components: *SHIFT, CSF,* and *HOPE.*

SHIFT The SHIFT system forms the subject of the present paper. It is a general purpose facility for jobs with a broad range of I/O requirements and which require access to many Gigabytes of on-line data. SHIFT workstations are networked via both Ethernet and UltraNet. The SHIFT CPU and disk servers are currently SGI Power Series 340 workstations and the tape servers are SUN 4/330s.

CSF The *Central Simulation Facility* or *CSF* is a platform for CPU-intensive work with low I/O requirements. The service runs on 16 HP9000/720 machines which are networked via Ethernet and which have full access to the SHIFT tape service. To the end user, CSF systems are seen as a single batch facility.

HOPE The HOPE service is an earlier system based on 3 APOLLO DN10040 machines. It is for CPU-intensive, low I/O work and it will be phased out during the course of 1992 as HOPE workload is taken over by *CSF.* HOPE is a joint project between *Hewlett-*

Service	CPU (CU)	Disk (GB)	3480 Tapes	8mm Tapes
SHIFT	100	150	6 manual	2 manual 2 robotic
HOPE	50	10		
CSF	150	10		

Table 2: CERN – Central RISC Services

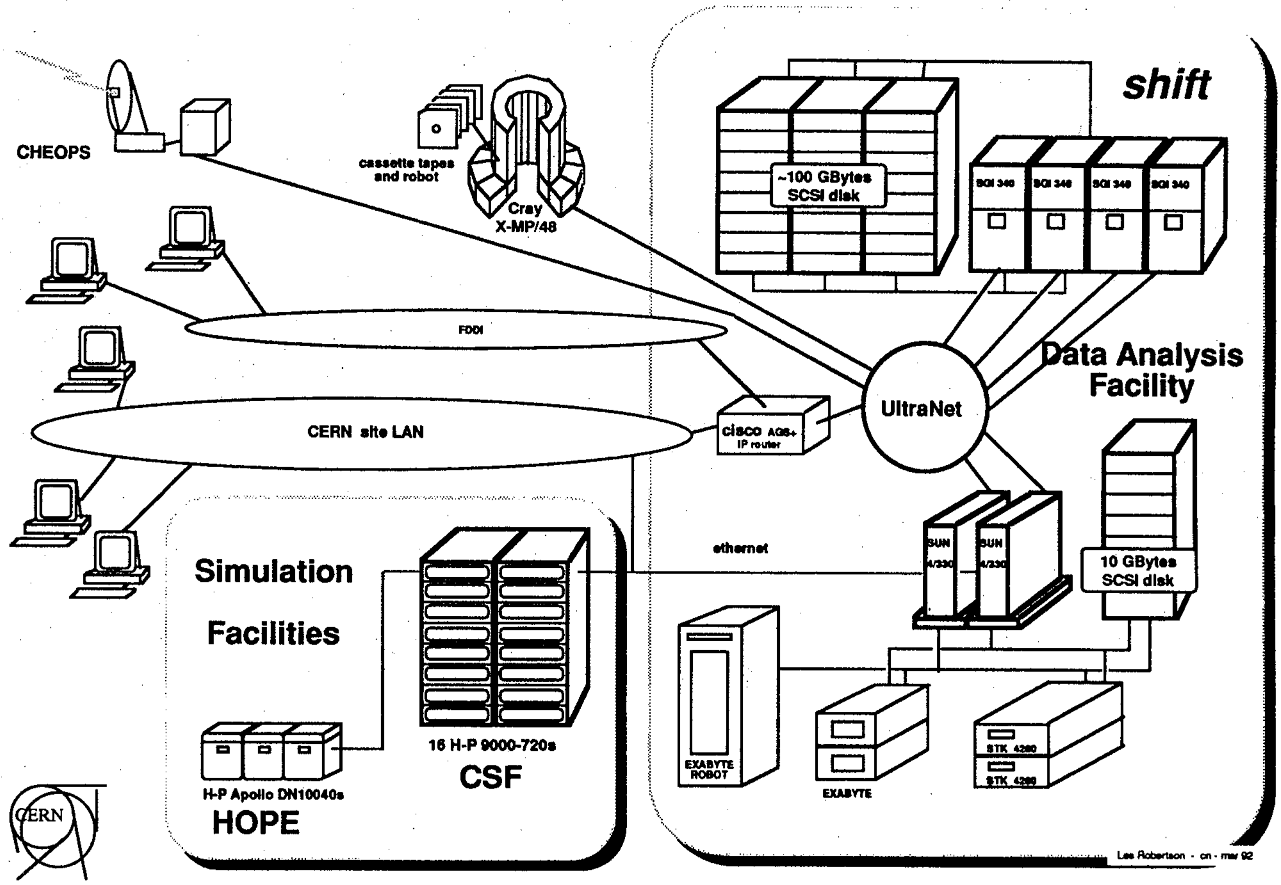

Figure 1: CERN – Centrally Operated RISC Environment

Packard and *OPAL*, a large physics collaboration based at CERN.

The current configuration for the centrally operated RISC-based workstation batch services is given in Table 2, and also indicated in Figure 1.

Project Goals

The goal was to develop an architecture which could be used for general purpose scientific computing, could be implemented to provide systems with excellent price/performance when compared with mainframe solutions, and could be scaled up to provide very large integrated facilities, or down to provide a system suitable for small university departments. The resulting systems should present a *familiar and unified system image* to their users, including access to many Gigabytes of disk data and to Terabytes of tape data: this is what we imply by the word *integrated.*

The goals of the SHIFT development were as follows.

- Provide an INTEGRATED system of CPU, disk and tape servers capable of supporting a large-scale general-purpose batch service
- Construct the system from heterogeneous components conforming to OPEN standards to retain flexibility towards new technology and products
- The system must be SCALABLE, both to small sizes for individual collaborations/small institutes, and upwards to at least twice the current size of the CERN computer center
- The batch service quality should be at least as good as mainframe batch quality, operate in a distributed environment, and have a unified priority scheduling scheme
- Provide automatic control of disk file space, integrated with a tape staging service
- Provide support for IBM 3480-compatible cartridge tapes, Exabyte 8mm tapes, and other developing tape technologies, with access to CERN's automatic cartridge-mounting robots
- System operation and accounting to be integrated into the CERN central computer services
- The architecture should also be capable of supporting interactive scientific applications

SHIFT Architecture and Development

The SHIFT system has been outlined in earlier papers [1,2,3]. A prime goal of the SHIFT project was to build facilities which could scale in capacity from relatively small systems up to several times that of the combined power of the CERN central mainframes. To achieve this, an architecture was chosen which encouraged separation of functionality. This allowed modular extensibility, flexibility, and optimization of each component for its specific function. Figure 2 shows this schematic architecture.

The principal elements of SHIFT are logically divided into CPU servers, disk servers and tape servers, with distributed software which is

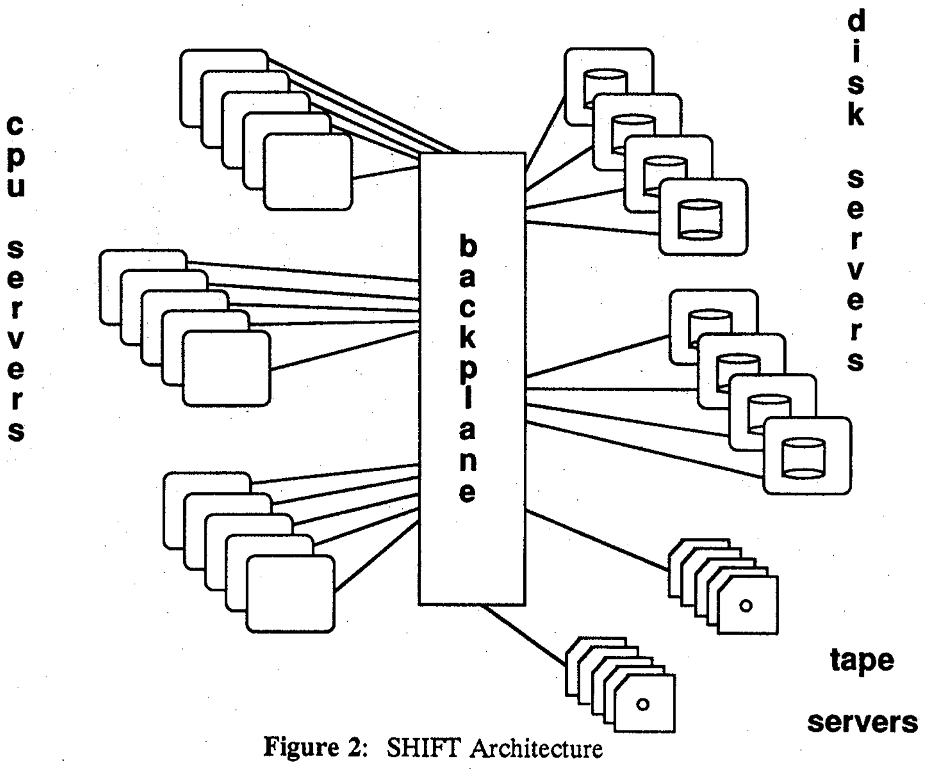

Figure 2: SHIFT Architecture

responsible for managing disk space, staging data between tape and disk, locating staged files, batch scheduling and accounting. These servers are interconnected by the *backplane*, a very fast network medium used for optimized special purpose data transfer. A detailed discussion of the backplane's requirements and properties is given later in the paper. The backplane is connected to the site's general purpose network infrastructure by means of an IP router, providing access to workstations distributed throughout CERN and at remote institutes.

An emphasis throughout the project has been on software portability, to allow flexible choices to be made for hardware platforms for each system component. Such choices can then be made using the most up to date evaluations of currently available products. Addition of further system types to the existing configuration is regularly reviewed. No major difficulties are foreseen in incorporating any UNIX based systems to SHIFT. As an example, a change from DEC to Sun workstations was made very quickly during the development of the system tape servers.

We believe that the modular approach we adopted was also the key to the very short development timescale we achieved. The design studies for the SHIFT project began in mid-1990. In parallel, the technical evaluations of various workstation and networking products were undertaken. By September 1990, code development had begun and orders for hardware had been sent out. The first local tests with SGI Power Series workstations connected via UltraNet took place at the end of December 1990. A full production environment was in place by March 1991. Software and performance improvements were made throughout 1991 and a decision to double the SHIFT CPU, disk and tape capacity was taken in November 1991 and carried out in early 1992.

The Backplane

A critical issue in the SHIFT design turns out to be the need for a high performance network: the *backplane*. Its aim is to provide to CPU servers remote disk and tape I/O facilities with as good performance and as low an overhead as I/O to locally-connected disks and tapes.

Our simulations of SHIFT configurations and workloads were used to compare various modern LAN technologies as backplane candidates. They showed not only that Ethernet was entirely inadequate, but that even FDDI would prevent scaling up to large SHIFT configurations due to its limited total bandwidth, as well as by its low delivered per-interface bandwidths which result from (today's) high CPU and system overheads when running TCP/IP over FDDI. Thus only small SHIFT systems can use an FDDI backplane.

Approximate backplane requirements can be illustrated by the following simple calculation: a medium I/O bound physics analysis job, running on a nominal 1 CERN Unit power CPU, is estimated to read about 20 KBytes/sec of data from disk, and to write up to half of this amount back to disk before completing. This translates to an aggregate 3 MBytes/sec backplane rate for a 100 CERN Unit system doing remote disk I/O. In a worst case scenario, all of this disk data must be staged from tape beforehand, and the results written back to tape afterwards, doubling the backplane aggregate to 6 MBytes/sec. If now heavily I/O bound jobs are included, this will multiply the aggregate still further: to avoid network congestion a backplane peak capacity of 15 MBytes/sec is considered necessary to support a general mix of I/O intensive jobs running on this medium-size 100 CERN Unit configuration. This already exceeds FDDI's possibilities.

Equally significant is the CPU consumption incurred by such data rates using presently implemented FDDI interfaces. Taking the nominal aggregate of 6 MBytes/sec from above, and noting that *two network interfaces participate in each backplane transfer*, a total of 12 MBytes/sec of interface activity is present under medium conditions on a 100 CERN Unit system. Measurements of various FDDI implementations at CERN have shown that between 2 and 6 CERN Units of CPU are needed to drive 1 MByte/sec through todays' FDDI interfaces, translating into a figure between 24 and 72 CERN Units for 12 MBytes/sec, or an average of **50% of installed CPU capacity**.

Finally, the *peak per-interface* data rate available will affect the number of networked units required to assemble a SHIFT configuration. Using the figure of 15 MBytes/sec peak backplane traffic (i.e., 30 MBytes/sec of peak interface traffic) on a 100 CERN Unit system with full tape↔disk staging, we find that the total interface traffic breaks down into 7.5 MBytes/sec each of CPU server and tape server traffic, plus 15 MBytes/sec of disk server traffic. In order to satisfy these rates with a reasonably small number of server modules (say about 3 of each type), we require sustained interface rates of between 3 and 5 MBytes/sec. Today FDDI can achieve only about half of these sustained data rates under normal production conditions, thus forcing the use of many server modules (particularly disk servers).

The current solution for the large SHIFT configuration at CERN is to use UltraNet [4] equipment for the backplane, as this product includes special purpose protocol-processing hardware on each interface, plus several times the FDDI total bandwidth. Equally importantly, it supports the most widespread standard TCP/IP application interface (BSD sockets).

However, the optimal use of UltraNet requires some special understanding. First of all, UltraNet is designed to assist *stream-type* applications and is not very effective for datagram or small packet-size transfers. Thus file access via stream sockets, with large record-lengths, is well supported whereas access via NFS is not. This was well understood from the start, and fitted our model of remote disk and tape I/O on condition that such accesses are sequential and use large record length; this is the case for High Energy Physics analysis programs, which typically use record lengths of 32 KBytes.

Table 3 summarizes the performance characteristics of FDDI and UltraNet, for simple memory↔memory transfers. It shows UltraNet's dependence on blocksize and (in the final column) the relative FDDI and UltraNet CPU costs of data transfer, expressed as the number of MBytes/sec achievable per single fully-loaded CPU. It can be seen that an UltraNet blocksize of 128 KBytes, even under such test conditions, is much more effective than one of 32 Kbytes. During our detailed performance analysis, 128 KBytes was found to be an optimal choice under actual operational conditions.

Software Architecture

Four areas of software development were identified in order that the SHIFT systems could offer a scientific computing environment comparable to that of a conventional mainframe.

Distributed Batch Job Scheduling The *Network Queuing System* (NQS) is used on SHIFT for batch job submission, control and job status enquiry. As the physics workload is located on several machines, the batch jobs must be scheduled evenly across all CPU servers to maximize job throughput and CPU utilization. In addition, users expect to see consistent job turnaround times. To achieve these goals, a load balancing scheme was incorporated into NQS.

Distribution of Files The SHIFT filebase is composed of many distinct UNIX filesystems located on different hosts across the network. Current technologies in distributed file systems are unable to satisfy the demand for data throughput. Moreover, these distributed file systems usually do not allow file systems to spread over more than one machine. A *Disk Pool Manager* (DPM) was developed to manage the SHIFT files and filesystems across the network.

Remote File Access Current distributed file systems have performance limitations when used for demanding applications over high speed networks such as UltraNet. A specialized Remote File Input Output (RFIO) subsystem was developed taking into account the underlying network characteristics.

Sink	Source	Blsize(KB)	MB/sec	CPU (Sink)	MB/1CPU (Sink)
UltraNet:					
SGI 340S	SGI 320S	10	2.8	35%	8
		32	4.7	21%	23
		128	8.8	14%	62
		256	10.5	12%	84
		512	11.3	9%	122
		1000	11.8	8%	140
Cray/LSC	SGI 340S	20	3.1	4%	80
		200	5.5	1%	550
		2000	6.0	1%	600
Sun4/330	SGI 340S	20	2.5	20%	13
		200	3.4	14%	24
		2000	3.5	13%	27
FDDI:					
SGI 320S	DEC 5200	32	2.2	40%	5.5
SGI 320S	Sun4/670	32	3.0	68%	4.5
Sun4/670	SGI 320S	32	3.2	60%	5.3
Sun4/670	DEC 5200	32	2.4	45%	5.2
DEC 5200	SGI 320S	32	2.1	80%	2.6
DEC 5200	Sun4/670	32	1.7	65%	2.6

Table 3: UltraNet vs. FDDI Performance (memory↔memory)

Tape Access High Energy Physics computing at CERN makes extensive use of IBM 3480 tape cartridges for the storage of data from experiments. A 3480 cartridge has a capacity of 200 MBytes. Exabyte cartridges are also used to a smaller extent. These have up to 5 Gigabyte capacity but with a data transfer rate substantially lower than the 3480.
UNIX systems have been traditionally weak in the area of tape support, one exception being the Cray UNICOS system. A portable tape subsystem was developed to manage a range of tape devices attached to different hosts. In addition, a remote tape copy utility, RTCOPY was developed which could be used in the distributed environment.

The subsequent sections describe each of the four areas in more detail.

Batch System Enhancements

The *Network Queuing System NQS* is a facility for job submission and scheduling across a network of UNIX batch workers. At CERN, it has been ported to numerous workstation platforms and useful enhancements have been added such as limits on the number of jobs run for any user at one time, an interactive global run limit, the ability to move requests from one queue to another and the ability to hold and release requests dynamically. Moreover, CERN has implemented in NQS the ability to have the destination server chosen automatically, based on relative work loads across the set of destination machines. Users submit jobs to a central pipe queue which in turn chooses a destination batch queue or *initiator* on the least loaded machine that meets the jobs' resource requirements. If all initiators are busy, jobs are held in the central pipe queue and only released when one becomes free. In addition, a script running above NQS holds or releases waiting jobs with a priority based on their owner's past and current usage of the SHIFT service.

Disk Pool Manager

The SHIFT data filebase comprises many UNIX filesystems which are located on any of the SHIFT hosts across the network. In order that users see a unified data file space, the notion of a *pool* was created. A *pool* is a group of one or several UNIX filesystems and it is at the *pool* level that file allocation is made by the user. Pools can be much larger than conventional UNIX filesystems even where logical volumes are available. Pools may also be assigned attributes. For example, a pool used for staging space can be subject to a defined garbage collection algorithm. The pools in SHIFT are all managed by the *Disk Pool Manager.* The *Pool Manager* balances disk space when creating new files and directories and it may be used to locate and delete existing files.

The interface to the *Disk Pool Manager* is via UNIX user commands. The **sfget** command allocates a file of a given size within a specified pool. The command returns a full path name for the file based on the convention that all SHIFT file systems are mounted globally with NFS on the mount point */shift/<host name>*. If the file requested already exists within the pool, **sfget** simply returns the path name without allocating any space. Other commands are provided to list, remove and manage files. In addition, a user-callable garbage collector has been implemented which maintains defined levels of free space in a pool. This is useful for physics data staging where data are copied from tape to disk before being accessed by user programs.

The design of a central *Pool Manager* has proved limited and, in particular, did not scale well with the rapid growth of the filebase. The first implementation was for a filebase of 40 Gigabytes whereas there are currently over 100 Gigabytes connected. In addition, problems arose from the fact that not all users were using **sfget** to allocate files and file system usage grew outside the control of the Pool Manager. To counter this problem, file system scans were incorporated to reflect the actual status and the results were stored in a centrally managed table. Overall performance is still a problem with the centralized *Disk Pool Manager* and a project is now under way to rewrite the software using a more distributed approach.

Remote File I/O System

The Remote File I/O system (RFIO) provides an efficient way of accessing remote files on SHIFT. Remote file access is also possible using NFS but RFIO takes account of the network characteristics and the mode of use of the files to minimize overheads and maximize throughput. RFIO maintains portability by using only the BSD socket interface to TCP, and thus operates over UltraNet, Ethernet, FDDI or other media. RFIO transmits I/O calls from client processes to remote RFIO daemons running on all SHIFT hosts.

RFIO is implemented with both C and FORTRAN interfaces. In C, the system presents the same interface as local UNIX I/O calls: **rfio_open** opens a file like **open(2)**, **rfio_read** reads data from a file like **read(2)** etc. Most High Energy Physics programs are written in FORTRAN, and usually interface their I/O via one or two intermediate library packages. RFIO has been incorporated into these, so its usage becomes completely transparent to the users of these programs.

RFIO was treated as one of the key performance factors of SHIFT. When a detailed investigation of system performance was undertaken, a major effort was made to reduce the operating system overheads incurred by RFIO. This is described in the section below on System Performance.

Magnetic Tape Support

High Energy Physics computing makes extensive use of IBM 3480 and Exabyte 8mm tape cartridges. The initial approach for tape access on SHIFT was to access tape units connected to the Cray UNICOS system. Subsequently, a portable UNIX Tape Subsystem was designed to satisfy all SHIFT's requirements in the area of cartridge tape access. The subsystem runs on all SHIFT hosts to which tape devices are connected.

Portable UNIX Tape Subsystem

UNIX systems usually offer a primitive tape interface which is not well adapted to a multiuser environment. Four basic functions are typically provided:

- **open**(2)
- **read**(2)
- **write**(2)
- **close**(2)

Several **ioctl**(2) commands are also provided but there is no operator interface, label processing, or any interface to a tape management system. The SHIFT Tape Subsystem offers dynamic configuration of tape units, reservation and allocation of the units, automatic label checking, an operator interface, a status display and an interface to the CERN/Rutherford Tape Management System. It is written entirely as user code and does not require any modification of manufacturers' driver code. It currently supports StorageTek's 4280 SCSI tape drive (an IBM 3480 compatible) as well as Exabyte 8200/8500 drives.

Automatic tape file labelling is not provided as this can only be done by modifying the tape I/O driver. Instead, a set of user callable routines were written to perform the same task. In practice, most tape I/O is done by using a tape staging utility, RTCOPY which hides details of these routines from the user.

Tape Copy Utility, RTCOPY

To provide tape access for every SHIFT CPU and disk server, a tape copy utility RTCOPY was developed which allows tape access across the network. Internally RTCOPY uses RFIO software to maximize the data transfer speed and thus minimize the tape unit allocation time. RTCOPY intelligently selects an appropriate tape server, by polling all known tape servers to query the status of their tape unit(s). RTCOPY supplies any missing tape identification parameters by querying the Tape Management System as needed. RTCOPY then initiates the tape copy, informs the user when the operation is complete, and deals with error recovery.

Software Maintenance and Distribution

The SHIFT software is distributed to many sites outside CERN and the distribution and maintenance across different platforms presents a new challange. We currently support SGI Irix 3.3.3, Sun3/SunOS 4.1.1, Sun4/SunOS 4.1.1, Sun 4/SunOS 4.1.2, UniCOS 6.1, DomainOS, VAX/Ultrix 3.x, DecStation/Ultrix 4.x, RS 6000/AIX 3.2, HP 9000/HP-UX 8.05.

Our experience has shown that UNIX tools like **make**(1) and **sccs**(1) are only a partial solution to the problem of software maintenance in a heterogeneous environment. For example, **make** does not support conditional rules, and subtle differences in operating system versions are hard to deal with. We are currently investigating the **imake**(1) tool used by the X11 consortium for maintaining our software suite.

System Performance

During the design phase of the project, performance estimates were made using a straight-forward simulation program using data obtained from benchmark programs running on limited test configurations (made available by potential suppliers and other organizations). When our own hardware was installed, these tests were repeated, placing a great deal of emphasis on what we believed would be the major performance issue, the UltraNet performance. Our initial configuration had only two disk channels and we were therefore unable to perform full-scale disk performance tests and were content to extrapolate the disk performance from simple tests.

These tests led us to assume that we would be able to support an aggregate (multi-stream) data rate between a two processor SGI 4D/320S disk server and a four processor SGI 4D/340S CPU server in excess of 6 MBytes/second with less than 60% utilisation of either the disk server's network interface or its CPUs (and therefore incurring no serious queueing problems). In practice we installed a four processor disk server, and so assumed that this aggregate performance target would be easily achievable; we thought that our second performance target, a single stream remote disk data rate approaching that of a local disk, would be much more difficult.

When SHIFT was initially commissioned, the job mix submitted was more or less as had been expected and the performance was adequate, satisfying the aggregate demand of about 2 MBytes/second. However, due to a change in physics emphasis after some months, many more I/O intensive jobs began to arrive and, in spite of the presence of the UltraNet backplane, the networking performance was now found to be quite disappointing. Instead of rising according to our estimates, we were seeing network data rates saturating at just over 2 MBytes/sec on the running systems (which were of course heavily loaded with batch computation and disk and Ethernet I/O), and at about 4 MBytes/sec when doing multistream RFIO under test conditions between unloaded SGI 4D/340's.

Investigation showed that the UNIX system load had become the factor limiting performance. This was much higher than we had predicted, and we assumed that it was due to the RFIO protocol and the architecture of the RFIO server. We therefore began a series of improvements in these areas.

The system overhead due to UltraNet is to a good approximation the same when transferring a few bytes or hundreds of kilobytes. We also noticed that there is a strong dependence of UltraNet performance on network block size, and we showed that in a general operating environment there was optimal performance when using a 128 KByte block. This led to the implementation of a buffered mode of operation for RFIO reads, which now transfer 128 KBytes across the network when using UltraNet.

At the same time we embarked on major refinements to the RFIO protocol, with the aim of minimizing the number of system calls required. The initial version of the RFIO protocol had been very straightforward and robust, simply mapping local FORTRAN and C file I/O calls to remote calls on RFIO daemons. This resulted in many small-size network transactions (e.g those mapping file seeks, I/O completions, etc.) being interspersed with transfers of user data.

The computing workload which we expect to support has a number of special features:

- the application record size is normally 32 KBytes;
- more than 90% of I/O operations are reads;
- most programs either read records from a file sequentially, or in a *skip-sequential* mode. In the latter case, the program uses a directory which contains some physics characteristics of each of the records in the file and pre-selects a list of the records which are of interest. The program then processes this list sequentially.

The RFIO protocol was thus re-designed with the following improvements:

- As far as possible control messages were eliminated by piggybacking them along with the data transfers.
- Three read access modes which could be specified by the user were defined: *sequential*, *pseudo-sequential* and *random*.
- Sequential file read access was optimized by using buffered read-ahead with 128 KByte data blocks: the server simply loops, reading from the disk and writing 128 KBytes to the network.
- Pseudo-sequential reading of files uses a *pre-seek* procedure call which enables the user to provide a vector containing a list of {record address, record length} pairs. This list is forwarded to the server, which pre-reads and blocks up the required data. The RFIO client returns it to the user program in response to appropriate seek and read requests.
- In random mode, the client requests each record in turn from the server as directed by the application program.
- The mode selected by the user is merely *advisory* in the sense that the result is functionally correct even if the application changes mode without informing RFIO. The mode selection is required only for performance.

Tests showed that a factor of about two had been gained in RFIO performance as a result of this work. Most of this improvement was due to the use of record buffering and large 128 KByte blocks (which of course itself implies a certain level of *read-ahead*).

Table 4 shows the aggregate data rates (Mbytes/sec) achieved for two modes of transfer for the initial and improved versions of RFIO:

Under test conditions, we were now able to achieve our target data rate (aggregate 6-7 MBytes/second), but this required so much of the disk server CPU that we could not achieve it under realistic production conditions. The RFIO protocol was now as simple as the test programs used in our initial benchmarks, and so we realized that our original method of estimating the performance must be

Number of streams	1	2	4	8
RFIO Version1 (32 KByte buffers)				
sequential	0.8	1.1	1.4	1.5
random	0.6	0.9	1.2	1.3
RFIO Version2 (128 KByte buffers)				
sequential	1.1	2.3	4.4	5.7
random	0.9	1.9	2.6	3.7
RFIO Version3 (128 KByte buffers + fewer system calls)				
sequential	1.2	2.4	5.1	6.7
random	1.2	2.2	3.5	6.0

Table 4: Aggregate RFIO Data Rates (Mbytes/sec)

flawed. We had been blinded by the assumption that the difficult task was the network performance, and we had neglected to study the disk performance. Only at this stage did we carry out full scale disk performance tests, and this immediately gave us the answer to the problem. On the main SHIFT disk servers (SGI multi-processor Power Series 4D/340 systems running IRIX 3.3.3) it was found that the CPU cost to perform a given unit of disk I/O (e.g., reading 1 MByte) increases with the system load. After detailed studies by the manufacturer, this is believed to be due to contention for the internal bus linking CPUs and memory. The load on the bus can be reduced by using *direct* I/O. This method circumvents normal file system operations and avoids the copy from kernel buffer to application buffer. Table 5 shows the improvement in both aggregate data rate and, more significantly, in CPU cost which is now constant.

Table 6 shows the results of a series of tests which read from disk and write their data to UltraNet using filesystem I/O and direct I/O. We are investigating how to exploit direct I/O within RFIO in a production environment.

Current Service at CERN

The SHIFT service at CERN currently has about 400 registered users from two large physics collaborations. Scripts have been implemented to handle most of the repetitive tasks such as account creation, automatic code updates, architecture-independent compilation and linking, tape staging, remote job submission, job query and file transfer, system and user accounting and so on. Usage of the service is expanding. Table 7 summarizes some of the current service characteristics:

As the SHIFT configuration expands, the number of physics groups given access will also grow. It is expected that the SHIFT capacity will again double this year, with extra CPU, disk and tape servers being added in a modular way.

Conclusions

We recognized from the start of the project that networking performance was a major challenge if SHIFT were to be able to handle I/O intensive problems. But we had not realized that *free protocol processing* was not the only answer to that problem; in fact operating system overheads remain the major challenge. Our initially simple remote file access

Number of streams	1	2	4	6	8
Using File System I/O					
Aggregate MBytes/sec	1.8	4.1	7.3	7.8	7.5
CPU cost sec/MByte	.15	.18	.29	.37	.41
Using Direct I/O					
Aggregate MBytes/sec	1.8	3.8	7.1	9.2	10.8
CPU cost sec/MByte	.04	.04	.05	.06	.06

Table 5: CPU Cost of Disk I/O

Number of streams	1	2	4	6	8
Using File System I/O					
Aggregate MBytes/sec	1.8	3.5	4.3	4.7	5.1
CPU cost sec/MByte	.20	.24	.27	.32	.40
Using Direct I/O					
Aggregate MBytes/sec	1.5	2.8	4.9	6.5	7.4
CPU cost sec/MByte	.08	.10	.10	.11	.13

Table 6: CPU Cost of Disk and Network I/O combined#

Users per day	50
Batch jobs per day	150
Tapes staged per day	150
Data staged per day (Gigabytes)	25
MTBI (hours) 1 Apr 91 - 1 Apr 92	150
Percentage CPU Currently Utilized	50

Table 7: Current SHIFT Service Profile

protocol, implemented on high-performance UltraNet sockets, required fundamental modification and tuning, taking advantage of some characteristics of our user applications, before it reached acceptable performance. Moreover, to maximize total aggregate throughput, it is necessary to bypass the traditional UNIX file system handling.

Another problem area we have encountered is that of disk unreliability and repair. A fifty thousand hour MTBF figure sounds good for a SCSI disk unit, but with over a hundred such disks installed this translates to a failure every few weeks. We have learned that we need RAID technology and/or disk mirroring techniques to deal with such issues.

Overall, the system's users consider SHIFT to be successful and are increasing their investments in such equipment. The CERN system is running a wide variety of physics production jobs, and has confirmed our belief that such an approach is entirely practical and economic for many physics computing applications. Even though far from being fully loaded, the current SHIFT is processing about 8,000 CERN CPU Unit-hours of work per week, for which it is mounting over 1000 tapes (and transferring about 150 Gigabytes) of physics data per week. The associated systems CSF and HOPE are processing an additional 20,000 CERN CPU Unit-hours of low I/O work per week. For comparison, the CERN central mainframes deliver a total of about 20,000 CERN CPU Unit-hours per week.

The CERN centrally operated RISC facilities are already delivering one and a half times as much physics computing as the conventional mainframe systems. We consider that the SHIFT goals listed earlier in this paper have been met, and that inexpensive RISC based workstations, suitably deployed, can now be used to provide reliable large scale scientific computing services.

Acknowledgements

Important contributions to this work were made by Alfred Lee, Thierry Mouthuy, and Steve O'Neale of the OPAL physics collaboration, and by Julian Bunn of CERN-CN in simulation studies. We also thank Gail Hanson of Indiana University and David Williams, CERN-CN Division Leader, for their support, and HP-Apollo for their generous contribution of equipment.

References

1. "SHIFT, the Scalable Heterogeneous Integrated Facility for HEP Computing", J-P. Baud et al., Proc. Conference on Computing in High Energy Physics, (CHEP 91), March 1991, Tsukuba, Japan. Universal Academic Press.
2. "Scalable Mainframe Power at Workstation Cost", J-P. Baud et al., Proc. Spring 1991 EurOpen Conference, May 1991, pp. 113-122.
3. SHIFT User Guide and Reference Manual", J-P. Baud et al., CERN-CN, 1211 Geneva 23, Switzerland.
4. UltraNet – An Architecture for Gigabit Networking", R. Beach, Proc. IEEE Conference on Local Computer Networks, September 1990.

Author Information

Jean-Philippe Baud has a background in databases and systems programming for supercomputers. He is responsible for the design and implementation of the portable tape software. Contact him via e-mail at baud@cernvm.cern.ch.

Christian Boissat is a systems programmer in the software group at CERN. He is responsible for development of the Network Queueing System, NQS. His e-mail address is boissat@vxcern.cern.ch.

Fabrizio Cane is a CERN fellow and he implemented the Disk Pool Management software. His e-mail address is cane@vxcern.cern.ch.

Frederic Hemmer is a database and distributed computing specialist who joined CERN in 1984. He is responsible for the overall software architecture, and its continuing development and maintenance. His e-mail address is hemmer@sun1.cern.ch.

Erik Jagel joined CERN as a physicist in 1988. He shared responsibility for SHIFT design and simulation and he now works in the computer center where he is in charge of CSF and HOPE operations. His e-mail address is jagel@cernapo.cern.ch.

Ashok Kumar is a computer scientist specializing in real time systems and parallel computing. He is responsible for performance measurement and analysis and for porting the RFIO subsystem to Ultrix and AIX platforms. His e-mail address is ashok@sun1.cern.ch.

Gordon Lee is a UNIX systems specialist in the User Support Group at CERN. He is responsible for all system administration of the SHIFT configuration. His e-mail address is gordon@cernvax.cern.ch.

Bernd Panzer-Steindel is a physicist working in the OPAL collaboration. He has been actively involved in using and developing the SHIFT service.

Les Robertson has spent 18 years at CERN, in various technical and managerial positions. He is currently the leader of a series of projects, including SHIFT, using RISC computers to provide scientific computing services. His e-mail address is les@cernvm.cern.ch.

Ben Segal is a networking and distributed computing specialist, at CERN since 1971. He was responsible for the SHIFT network development. His e-mail address is ben@cernvax.cern.ch.

Antoine Trannoy spent 15 months as a visitor to CERN. He wrote the tape copy software and enhancements to the RFIO package. He is currently employed at *Centro Ricerche Sviluppi et Studi Superiori* in Sardinia, Italy. His e-mail address is trannoy@cernvm.cern.ch.

Igor Zacharov is a physicist who was a CERN fellow from 1989 to 1991. He now works for Silicon Graphics and is responsible for the CERN account. His e-mail address is zacharov@cernvax. CERN is located at 1211 Geneva 23, Switzerland.