Partitioned Distributed Computing

by George Yates [1]
Social Science & Public Policy Computing Center
University of Chicago

Overview

The Social Science and Public Policy Computing Center (SSPPCC) at the University of Chicago was founded in February, 1990, to provide general computing services to the University's social science research and education community. At the time, Sun Microsystems, Inc., had recently started shipping the SPARC Station 1 UNIX workstation with a RISC architecture which presaged a new class of small computer cost-performance. DEC, IBM, SGI, HP, and other UNIX workstation vendors followed suit over the next 18 months, releasing RISC workstation systems in a rapidly improving cost-performance trend. Effective industry standards for hardware and software facilities significantly unified the emerging technology in all these releases. Today, small computer systems exhibit scalar CPU performance and data storage performance and capacity that rival or exceed mainframe levels at a small fraction of the cost. Third-party software and hardware developers provide extended facilities that operate on most or all of the major vendors' systems and a network of such systems can cooperate to automatically distribute the community-wide process load. In effect, a community of users can now build a large-scale computing environment from parts supplied by an entire industry of competing vendors.

SSPPCC's technical strategy for large-scale computing in the social sciences is based on a pure distributed computing model designed as a multi-vendor network of cooperating UNIX workstations that automatically share job load submitted at any site in the local UNIX network domain. The design is being implemented using a broad selection of hardware and software technology and investigative system administration. It is important to briefly describe the current computing environment to establish the scale of contemplated services.

SSPPCC's current large-scale computing environment contains 16 high-performance UNIX workstations from HP, Sun, and IBM, equipped with 608 MB of RAM from HP, Sun, IBM, Technology Works, Kingston, and Clearpoint. Peripherals include 40 GB of disk space from HP, Seagate, IBM, and Maxtor, plus user-operated nine-track and DAT tape drives from HP and cartridge tape and CD-ROM drives from IBM and Sun. This UNIX environment is accessed from 360 networked IBM/PC-clone or Apple Macintosh desktop computers and Wyse or Qume terminals. Today, active networking technologies are Ethernet and AppleTalk driven by electronics from Cabletron, Cayman Systems, and Farallon. FDDI technology is scheduled for testing in July, 1992, using concentrators from DEC, Cabletron, Ungermann-Bass, and National Peripheral Devices.

The processing load on SSPPCC UNIX workstations originates from 1,400 login accounts, with 50–100 active UNIX logins at mid-afternoon. Research software is primarily data management and statistical analysis tools including SAS, SPSS, BMDP, Stata, MatLab, LimDep, GLIM, HLM, and Mathematica, plus custom research software in Fortran and C. Commonly used software is implemented on all machine types. An offline tape library contains over 2,500 titles, including Census, CPS, PSID, NLS, NELS, HS&B, SIPP, and IMF.

The entire assembly is growing rapidly in all dimensions, constrained mostly by budget and management issues. Importantly, growth is tightly organized by a comprehensive technical design for distributed computing. Primary distributed computing concepts addressed in the current SSPPCC design are:

1. Networking strategies,
2. Location-brokering,
3. Distributed File Systems,
4. Hierarchical Storage Schemes,
5. File Migration Service.

Implementation of these design concepts is in different stages. SSPPCC *networking strategies* are a careful design for partitioning network traffic so that the highest loads are on the fastest links. As mentioned above, high-speed FDDI is scheduled for testing soon; even faster HPPI technology is awaiting product announcement for UNIX workstations in late 1992. *Location-brokering* automatically locates job execution on a workstation that is most suited to the job load. HP's Task Broker software is specified to provide location-brokering services. Task

Broker has been extensively tested, is in limited use on HP and Sun workstations, and will be in common service by Fall, 1992. *Distributed file systems* (Sun's NFS and Transarc's AFS) have been extensively tested on all platforms. NFS is extensively used; AFS is implemented in a limited system role on all machines, to be in common user service by Summer, 1992. *Hierarchical storage schemes* that provide very large capacity storage are well-researched but untested by SSPPCC. (An R–Squared system that runs the Sybase relational DBMS server on hierarchical storage is in use today at one SSPPCC client site and represents a limited test.) Storage schemes being considered are implementations of the IEEE Mass Storage System Reference Model (MSSRM)[2] from two systems integrators—Advanced Computing Support Center (ACSC) and Epoch. An implementation of the MSSRM storage server from one of these vendors will be the primary strategy for large data base handling in the SSPPCC design. Both of these implementations include a *file migration service* which automatically removes and replaces files on local workstation disks as required by local filesystem usage. Either ACSC's UniTree on IBM's RS/6000 workstation or Epoch's Rennaisance Migration Service (RMS) on Sun's SPARC Station (to be ported to the RS/6000 in 1992) will be implemented in 1992, budget permitting.

This article describes the SSPPCC design for the distributed computing functions listed above. For ease of expression, the perspective in the following text is that all services are operating. After details of the design are developed, they will provide context for further discussion of design implementation stages at Chicago. This article concludes with consideration of design implementation on a national scale.

Networking Strategies and Location-Brokering

In distributed computing across a network, a job is not necessarily executed at its submittal site. Instead, the networked computers cooperate (unseen by the user) to locate the job's execution at a machine deemed most suitable for the job load, considering the job's characteristics and the current load at all sites. Such cooperative job placement is called *location-brokering*. Since social science research jobs are frequently I/O-bound and a relocated job might execute remotely from the disks containing required data files, efficient communication to ship data to a job's execution site is essential to realizing the gains of a distributed computing strategy.

SSPPCC's design specifies a physical network structure that allows strategic choices for permanent file location and job execution location so that any required file shipment on the network uses a communications link whose speed is appropriate for the file size. Ideally, larger files are shipped on faster links. Thus, network structure and location-brokering are coupled concepts and are discussed together in this section.

The specified network structure is a layering of four standardized communications technologies into a hierarchy of links with graduated speeds. Specified technologies and nominal link speeds are:

1. LocalTalk—0.275 Mbits/sec. Used in SSPPCC networks of Apple Macintosh computers for very low-volume traffic in printing, low-volume file-sharing, and messaging during sessions with E-mail and login servers.

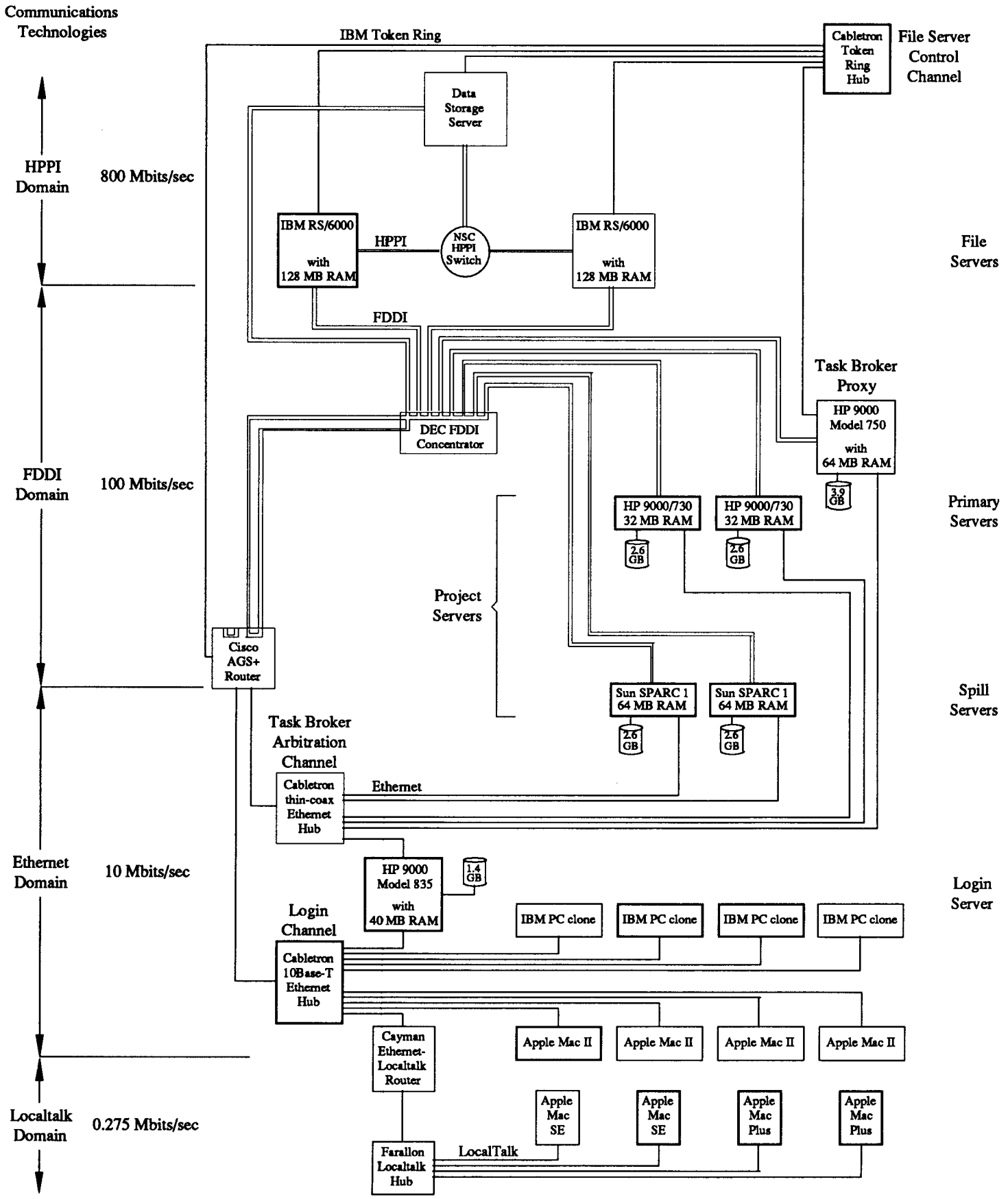
2. Ethernet—10 Mbits/sec. The most common technology for networks of desktop computers and UNIX workstations. Used for low-volume traffic including print files, login and E-mail session messages, small data files, and process control messages.

3. Fiber-Distributed Data Interface (FDDI)—100 Mbits/sec. The fastest technology commonly available for local-area networking. Being tested by SSPPCC for use as the initial primary medium for file-sharing in the distributed computing net.

4. High Performance Parallel Interface (HPPI)—800 Mbits/sec. The most recent technology in practical use for inter-computer communications. Proposed for use in the SSPPCC design for very-large file shipment. HPPI has a high-speed variant that doubles the data path width to achieve nominal 1,600 Mbits/sec. This variant is not currently announced for implementation in workstation connections.

In the parlance of the OSI networking model, the Link Layer protocol standard implemented for each of these technologies is IEEE 802.2 (LLC) at the service interface. Thus, though each Physical Layer specification is radically different, software control of these four technologies can be uniform above layer 2. In particular, the layer 3 Internet Protocol (IP) is implemented on each of these technologies.

Figure 1, Schematic Representation of Conection Domains, on the next page is a schematic diagram of SSPPCC's design for inter-computer connectivity. It shows a network connection geometry partitioned by the four networking technologies into a hierarchy of link-speed domains. (The connection topology diagrammed in each domain is a star, consistent with SSPPCC cable plant design and physical layer requirements.) Describing each of the labelled elements in Figure 1 as sites of computerized functions and services is the goal of the remaining discussion.


Communications Technologies
IBM Token Ring
Cabletron Token Ring Hub
File Server Control Channel
Data Storage Server
HPPI Domain
800 Mbits/sec
IBM RS/6000 with 128 MB RAM
HPPI
NSC HPPI Switch
IBM RS/6000 with 128 MB RAM
File Servers
FDDI
Task Broker Proxy
HP 9000 Model 750 with 64 MB RAM
3.9 GB
DEC FDDI Concentrator
FDDI Domain
100 Mbits/sec
HP 9000/730 32 MB RAM
HP 9000/730 32 MB RAM
2.6 GB
2.6 GB
Primary Servers
Project Servers
Cisco AGS+ Router
Sun SPARC 1 64 MB RAM
Sun SPARC 1 64 MB RAM
2.6 GB
2.6 GB
Spill Servers
Task Broker Arbitration Channel
Cabletron thin-coax Ethernet Hub
Ethernet
HP 9000 Model 835 with 40 MB RAM
1.4 GB
Ethernet Domain
10 Mbits/sec
Login Channel
IBM PC clone
IBM PC clone
IBM PC clone
IBM PC clone
Login Server
Cabletron 10Base-T Ethernet Hub
Cayman Ethernet-Localtalk Router
Apple Mac II
Apple Mac II
Apple Mac II
Apple Mac II
Apple Mac SE
Apple Mac SE
Apple Mac Plus
Apple Mac Plus
Localtalk Domain
0.275 Mbits/sec
Farallon Localtalk Hub
LocalTalk

Specialized hardware and software is specified in each domain as required to support specialized services provided in the domain. The HPPI domain contains IBM RS/6000 File Server and Storage Server systems specialized for large data base handling. IBM and third-party developments for RS/6000 POWER architecture and intrinsic AIX 3.2 I/O services are essential to the high-performance I/O required in the HPPI domain. The FDDI domain contains HP and Sun Project Server systems intended for general research computation. HP 700 series workstations currently provide the fastest CPU performance, the strongest upgrade path, and the best cost-performance in the workstation industry; Sun SPARC Stations are the most common and most inexpensive software development platforms. The Ethernet domain contains Login Servers and desktop computers for small-scale computation and office services required by the largest user group (plus the control channel for location-brokering). It is the domain for older UNIX systems, like the HP 8x5 series, which has reduced research computing value but supports Task Broker and is supported by a strong maintenance organization. The LocalTalk domain supports small-scale functions using Apple Macintosh computers and printers.

Common social science research activity is naturally suited to the diagrammed structure. For example, in the most common social science paradigm for research data file construction, a large file is read to extract a smaller research file for subsequent processing by a variety of statistical software. In the SSPPCC environment, source file sizes are typically 100–1,000 MB and extracted file sizes are typically 10–100 MB. This paradigm suggests a distributed computing scheme in which extraction jobs are located in a hardware and software domain specialized for large data base handling and statistical analyses are located in a domain specialized for research computation. Specifically, extraction jobs should be located in a domain with fast network links and large, fast disks, like the HPPI domain, while statistical analyses should be located in a domain with lower emphasis on large file handling and greater emphasis on multi-tasked CPU performance and broad software support, like the FDDI domain.

As stated above, the primary principle for configuring the layered network specifies that a file should be located in a domain where services are specialized for efficient handling of the file size. The assignment of file sizes to domains is roughly calibrated by requiring that job relocation by location-brokering minimally impact job throughput in the domain of Project Servers where the bulk of research computing is done. As already explained, relocating a job to a machine remote from the disks holding its data introduces network communications overhead into job I/O. By specifying a data communications technology for the Project Server domain whose *effective* data transfer rate dominates that from disk-to-host, communications overhead will not slow job processing because the slowest I/O transfer will then occur in the disk-to-host step. The calibration emerges from this rationale.

Project Servers perform statistical analyses on data files typically sized from 10 to 100 MB. These workstations are equipped with a variety of SCSI I/O interfaces connected to a variety of data disks. SCSI-2 performance is typical of current disk equipment. The effective transfer rate of SCSI-2 implementation is about 0.7 MB/sec. Effective Ethernet rates are about 0.2 MB/sec while effective FDDI rates are about 2 MB/sec. Hence, FDDI rates should dominate Project Server disk rates even when the FDDI net is under load. Thus, FDDI technology is specified for file-sharing between Project Servers so Figure 1 shows the Project Servers in the FDDI domain. Since the typical research data file size is 10–100 MB, we conclude that files larger than 100 MB should be located in the HPPI domain, the FDDI domain should hold files in the 10–100 MB range, and the Ethernet domain should hold files smaller than 10 MB.

It should be clear now that SSPPCC's distributed computing design does not view the workstation systems and network links as an assembly of homogeneous hardware and software capabilities and functions. Perhaps a more descriptive term for the design specifications is *partitioned distributed computing*. To understand how location-brokering distributes job execution into the partitions defined by file-size constraints, some details of the job relocation strategy are needed.

Referring to Figure 1, all Project Servers and Login Servers run location-brokering software called *Task Broker*, Hewlett-Packard software implemented for HP and Sun workstations. Local systems staff configures a Task Broker daemon running on each machine to broker the execution site for selected UNIX commands. When a user submits a brokered command, the submittal site, called the *client* (a Login Server, for example), broadcasts the command, its parameters, and the user ID to the community of Task Broker *servers* connected to the Task Broker Arbitration Channel in the Ethernet Domain (see Figure 1). Each server independently evaluates its current load, the characteristics of the command, and the user ID, summarizing its willingness to accept execution of the command by reporting an *affinity* integer in the range 0–999 back to the client. Affinity calculations are programmed by systems staff specific to the server and the command. The client (which might also be a server) awards job execution to a server reporting the highest non-zero affinity. Remote file systems needed during execution and unavailable to the server are then NFS

auto-mounted and the job is dispatched by the daemon's invocation of the Task Broker *service script* on the server. When the server completes the job, the service script automatically returns products to the client, subject to the file-size constraints of the client's domain. The client notifies the user of job completion by a screen display if the user's submittal login is still active, and by electronic mail if it is not active. The user might never know which workstation actually executed the job. The location-brokering process is thus a client auction of job execution to the servers.

Reporting zero affinity constitutes a server's refusal to accept assignment. Affinity programs generate zero if the brokered command software is not installed on the server, if the server is not configured to run the command as parameterized, if the server is too busy to accept the job, or if the user is restricted from using the server. If all responding servers bid zero, the job is queued on the client for a fixed period (or until a server notifies the client of another brokered job's termination), whence the client opens another auction.

Given this job-relocation strategy, significant characteristics of the distributed computing operation are determined by affinity calculations. The SSPPCC design specifies that affinity calculations must follow certain rules. The important rules are discussed in the following paragraphs.

When a project registers with SSPPCC for computing services, it is assigned a Project Server based on information gathered in an interview of project staff. In SSPPCC parlance, if a research project's permanent data files are located on disks connected to machine M, the project is *local to machine M*. The primary rule of affinity calculation specifies that a server must calculate an affinity biased upward for a command submitted by a local project; i.e., if the command parameters indicate that required data reside on local disks, a server adjusts its affinity upward, subject to its ambient load conditions. Because of this rule, there is a systematic bias to locate job execution where the project is located, and thus a systematic bias against saturating the network.

There are two classes of Project Servers, called *Primary Servers* and *Spill Servers*. Superior performance of HP 700 series workstations compared to Sun SPARC Stations make them the preferred sites for project locations, so HP 700 series machines are Primary Servers. Primary Servers adjust their affinity for a non-local project's command downward; hence they are biased to prefer local project commands and to disdain requests for non-local project service. Spill Servers calculate affinities between the preference and disdain ranges of the Primary Servers. Thus, as a very active project crowds its local Primary Server with excess load, the load migrates first onto the Spill Servers before crowding other project activity on Primary Servers. Sun SPARC Stations are specified as the Spill Servers because Spill Servers must ideally accept job load originating from any site in the domain. Since Sun workstations are the most active and economical software development platform, all software used in the SSPPCC domain is implemented on them at the lowest cost.

To preserve the file-size domains described above, a server will bid zero for any command requiring a remote file larger than the range assigned to its domain. Hence, a Login Server in the Ethernet domain will bid zero for any command requiring a remote file larger than 10 MB. A Project Server will bid zero for any command requiring a remote file larger than 100 MB. Thus, projects with permanent files larger than 100 MB must be located on the File Servers in the HPPI domain.

The restriction of large file extractions to the HPPI domain emerges naturally in this structure. A user submitting a job from anywhere in the network to extract data from a source file larger than 100 MB must reference the source file in his command syntax. All servers below the HPPI domain will bid zero for the job. Only the File Servers will participate in the auction and the extraction will be awarded to a File Server currently suited for the load, with preference for the Server where the source file resides. File Servers are equipped with very high-performance disks for the huge I/O tasks assigned to the HPPI domain. Since HPPI channel transfer rates dominate even these disk transfer rates, job location on a File Server remote from the data will not impact job throughput. Thus, HPPI domain job re-location operates as in the other domains and job execution suited for the HPPI domain is automatically located there. This example is canonical for partitioned distributed computing.

Importantly, Task Broker is not implemented for IBM RS/6000's specified as File Servers in the HPPI domain. This problem is circumvented by designating an HP 750 as Task Broker Proxy for the File Servers (see Figure 1). In an auction run by a client outside the HPPI domain, this proxy machine calculates its own bid *and* directly invokes affinity calculations on each File Server via the token ring control channel using standard UNIX RPC facilities. The File Servers report their affinities back to their proxy. The proxy then reports the highest bid to the client. If the client awards job execution to the proxy and the high bidder was a File Server, the job is dispatched on the File Server by the proxy, again using RPC over the token ring channel. Conversely, commands directly submitted to a File Server, so that the File Server is the Task Broker client, are brokered simply by the File Server's RPC execution of the same command on the proxy machine. The HP 750 functions as proxy client in

this case, auctioning the job as if the user had typed the command directly to it.

It should be clear that using Task Broker to make sophisticated judgments about job placement on a per-job basis requires significant program development by local system staff. Task Broker as shipped is primarily the skeleton for implementing distributed computing strategies.

By the above strategies, jobs submitted to any UNIX machine in any domain are automatically located on workstations suited to the loads. Each machine loses its stand-alone identity and contributes its CPU and I/O resources as subsystems to a common pool of computing services. Such a distributed computing structure is called a *multicomputer*. In SSPPCC's Task Broker implementation, the multicomputer network is guarded against saturation, local projects are shielded from remote extravagant projects, and job load is located in domains specialized for job characteristics.

The File Server Control Channel shown in Figure 1 is not just a Task Broker arbitration channel for communication with the Proxy. Another primary function of this channel involves the HPPI domain's Storage Server which maintains a large data library accessed via the network by all workstations acting as Storage Server clients. In particular, File Servers access very large data files located in the library via HPPI links. HPPI is a point-to-point technology, so for each HPPI packet transferred from the Storage Server to a File Server, the single HPPI interface of the Storage Server must be committed to a client connection established through the Network Systems Corp. HPPI switch. In order to reserve the Storage Server's single HPPI data channel for block transfers, each File Server client must communicate its request for the next data block to the Storage Server. The Server queues these requests and ships the blocks as they arrive from its peripheral devices. The HPPI channel is reserved by requests on the File Server Control Channel. It also serves as the file migration channel for the Ethernet domain via the Cisco router. (The Storage Server and file migration services are described in detail below.)

The description of Figure 1 is completed by remarking that networked desktop PC and Macintosh computers typically access the multicomputer via the Login Server. Since a user might never know which workstation actually executed his job, a Login Server appears to the user as a machine capable of running every installed program on files of arbitrary size with robust throughput under load. In other words, the Login Server appears as a powerful mainframe. But a multicomputer enjoys distinct and profound advantages over a mainframe:

1. It is dramatically cheaper, costing less than 10% of mainframes with similar capacity.

2. It is out-of-service only in the event of central power failure. The redundancy of the independent systems is a virtual guarantee against total system downage.

3. It is indefinitely extendible to accomodate increased load at relatively low cost. Adding new machines (or new sub-nets in any of the technologies) is a simple network expansion. New machines are purchased at workstation pricing.

4. It does not obsolesce in the usual sense. New nodes at the current level of technology can be continuously added. Older nodes are not necessarily replaced; they participate in the client auction with diminished bids. Furthermore, new nodes are dramatically lower-priced than large platform upgrades. Upgrades can be small specialized increments tightly correlated with demand. Institutional budget planning for multi-million dollar upgrades every 5 to 10 years is never necessary.

5. There is no hardware vendor lock-in. Hardware upgrade is not constrained by a single vendor's upgrade calendar. New nodes can be acquired from any vendor able to satisfy a specification of formal system requirements. To connect to the multicomputer, a workstation must run required networking services (ARPA-Berkeley, NFS, and AFS). All major vendors satisfy these requirements. Lack of location-brokering software is circumvented by use of the proxy strategy described above.

Indeed, the popularization of workstations and the resulting potential for propagation of multicomputers has profound implications for the future of mainframes.

Because the multicomputer is a shared centrally-supported instrument like a mainframe, it also has advantages over decentralized solutions. Arguments for decentralization are constrained by lower-powered hardware, inequities in the availability of services to a large group, and the redundancy of relegating to local sites all the burdens of staff expertise, system maintenance and upgrade, planning, and software acquisition and support. As administered by SSPPCC, the multicomputer avoids all these evils.

Distributed File Systems

In networked file-sharing, a directory structure residing on a disk connected to a remote machine (the file-sharing server) is made available to a local machine (the file-sharing client) as if it resided on a disk directly connected to the client. NFS and AFS are the file-sharing

services specified by the SSPPCC design. The two facilities are not equivalent. They differ markedly in their strategies for record-level I/O, their strategies for file security and data integrity, and their system management facilities. Additionally, today NFS is a facility bundled with every major vendor's UNIX software so any machine can be an NFS server or client. In contrast, the AFS server facility must be separately purchased from Transarc for each server, hence only designated machines can be AFS servers today. Because AFS 4.0 is the Distributed File System of the OSF DCE which enjoys wide vendor acceptance, the AFS server facility might one day be as ubiquitous as NFS without additional cost.

In the SSPPCC design, at least one AFS database server is specified in each of the Ethernet, FDDI, and HPPI domains. All workstations in the multicomputer are AFS clients. All AFS database servers are referenced by all AFS clients, hence files in every AFS volume are available to all machines in the multicomputer, subject to ordinary data security protections. Except as noted in the next paragraph, AFS database servers ordinarily hold only relatively fixed files such as software libraries and social science data base archives. Users can request private AFS volumes with specified quota to hold user files accessible by any Task Broker server.

Location-brokering invokes NFS services whenever an existing file is otherwise unavailable to a Task Broker server. New files created by a re-located job are never allocated in an NFS-mounted filesystem during job execution because *writes* by an NFS-client process addressed to a remote filesystem are slow—they must complete on the NFS-server disk before I/O completion is signalled to the NFS-client process. Instead, new files are created in scratch disk space directly attached to the Task Broker server and returned to the Task Broker client at job termination, subject to file-size constraints. If the new file is too large for the domain of the Task Broker client, the file is copied to the AFS server in the proper multicomputer domain with full user access privileges. (Unlike NFS, *writes* by an AFS-client to a remote AFS filesystem are cached on a local AFS-client disk so that I/O wait-time is significantly reduced.) The user is notified of the file's AFS path name during the job-completion notification sequence described in the previous section. The AFS file will be automatically scratched 120 hours after creation, so the user must execute an administrative action to save the file.

Importantly, scratch disk space on Task Broker servers is supported by the central file migration service described in the next section. Essentially, file migration service provided by the Storage Server in the HPPI domain ensures that scratch disk space cannot fill, subject to configuration limits on the Storage Server. Hence, the user is protected against early job termination when new file creation at the Task Broker server exceeds scratch disk capacity.

Hierarchical Storage Schemes and File Migration Service

The Storage Server shown at the top of Figure 1 is a formal MSSRM service that separates high-level UNIX and network filesystem facilities from the physical structure of storage devices. The SSPPCC design specification for MSSRM storage services is a client-server model for data management in which the Storage Server maintains a very large library online, responding over network links to requests for sequential data access by workstation clients. Importantly, client access to a library file is managed by communications software on both the server and the client so that details of the client-server interaction are hidden from a program running on the client. In effect, the entire Storage Server system and communication link appear to each client program as a very large disk connected to the client workstation. This structural transparency means that commonly used commercial software, like SAS and SPSS, can access files in the library with no change in existing software or user procedures.

Client file-level and record-level access to the library must be fast because there is competing demand for server attention by several clients and the library files can be very large. For example, large surveys are often read several times for data extraction until subsequent preliminary analyses satisfy the research group that sufficient data for the research inquiry is obtained. But the library is too large to maintain on a collection of fast disks connected to the server. The capacity for maintaining the entire library online for sequential access is available only in slow-speed tape robots or optical juke boxes. This conflict between speed and size is reconciled in operation by an internal Storage Server structure called *hierarchical storage*. In a hierarchical storage scheme, the Storage Server is equipped with a *slow-speed hierarchy*, say, a tape robot large enough to hold the entire library, plus a *high-speed hierarchy*, a collection of very fast disks with capacity about 10% of the tape system (depending on summed sizes of files in common-use). The disks act as a cache for files requested from the tape library, as follows. When a client request is received, the server checks its disk cache to determine if the requested file is already resident there. If so, the file is shipped to the client immediately. If not, the server reads the file from tape and simultaneously writes each record to its disk cache and to the client. Subsequent requests for that file will then be found already in the cache. In effect, the cache is a small "window" to the large library through which, at any one time, clients can view the currently active files at high speed. To maintain

available space in the relatively small cache, in idle moments the storage server executes an aging algorithm on cache contents that is parameterized by system staff for sensitivity to the identity of cached files and the time elapsed since last reference. If a file is designated for removal from the cache, it is simply erased if it has not been changed, or rewritten to tape in place of the original copy if it has changed. Efficient tape usage is automatically administered by the server.

A storage server is typically programmed to provide other essential support functions, such as network-wide disk backup to the slow-speed hierarchy and online library administration and documentation.

As mentioned in the Overview, Storage Server software is available today from ACSC (UniTree running on an RS/6000 under AIX 3.2) and Epoch (RMS running on a SPARC Station under SunOS 4.1.2). The products provide essentially the same hierarchical storage management, but UniTree's *Central File Manager* (CFM) only operates as a single NFS file-server while RMS operates as a networked disk analogue that can support multiple independent NFS file-servers (by file migration as described below).

To configure the Storage Server, one must specify the workstation base system, the disk cache, and the large-capacity library unit. As an expensive look-ahead example, a UniTree Storage Server realized in early 1993 by an IBM RS/6000 Model 560 with 128 MB RAM and equipped with an IGM 270 GB Exabyte tape carousel, a dual-ported 27 GB Seagate Elite-3 enhanced IPI-2 disk cache (eight 3.38 GB IPI disks), plus IPI and HPPI interfaces for the RS/6000 microchannel bus will deliver cached data from the 270 GB library to clients at *effective* rates in excess of 3 MB/sec (1 GB in 5 minutes), about five times faster than current SCSI-2 effective rates. (The Elite-3 disk and HPPI interface will not be announced until Fall, 1992. Such an assembly with 2 HPPI-equipped clients would cost an estimated $350,000 at academic pricing.) More conservatively, in July, 1992, UniTree driving the same workstation and tape carousel but with a 15 GB Elite-2 IPI cache and FDDI interfaces instead of HPPI could deliver cached data to clients at about half the speed for half the price. If data is not in the cache when requested, the tape unit delivers data at roughly nine-track tape rates.

The speed of the disk cache is an important determinant of Storage Server data delivery rate. RAID, RAM-disk, and SCSI fast-and-wide technologies are all faster alternatives to the IPI disk cache specified above. RAID and RAM-disk are significantly more expensive per megabyte and SCSI fast-and-wide availability is indeterminate at this time, hence SSPPCC prefers IPI technology today.

Importantly, both ACSC and Epoch software offer file migration service. Essentially, file migration extends the Storage Server's disk cache management capabilities to every disk in the multicomputer. For each filesystem on a disk attached to any workstation, a logical file space local to the workstation is defined larger than the physical filesystem space. The total file space is actually available in the slow-speed hierarchy at the Storage Server. The local filesystem is managed as a window of active files on the larger logical file space, exactly analogous to the Storage Server's primary disk cache management. File migration is available for UniTree as a separate facility called the *Distributed File System Manager* (DFSM). In direct contrast, file migration is intrinsic to Epoch's RMS.

Current Implementation

The current status of multicomputer implementation is diagrammed in Figure 2. SSPPCC has connected public and private workstations below the FDDI domain with extensive LocalTalk and Ethernet links and is in the process of evaluating FDDI concentrators from four vendors by inter-operability and performance criteria. When the evaluation is complete, six workstations will be connected by FDDI links in the summer of 1992 as the initial realization of the FDDI domain. Figure 3 shows the planned extension of the current connectivity, including construction of the HPPI domain loosely scheduled for Summer, 1993. Figure 4 is a schematic diagram of the logical structure in Figure 3 constructed in four campus buildings in the SSPPCC domain.

The remainder of this section presents ongoing considerations in the refinement of the SSPPCC design.

The current SSPPCC sensibility about the choice between UniTree and RMS favors RMS for two ill-defined reasons: (1) RMS originated in the workstation industry whereas UniTree originated in mainframe distributed computing environments. It is possible that RMS is more mature as UNIX software—more efficient and more stable. (2) In MSSRM terms, UniTree is fundamentally constructed as a high-level service (NFS file service) integrated with the Storage Server primitive; hence it is "too big". RMS provides the primitive MSSRM Storage Server function and hence can naturally support a multitude of MSSRM-compatible high-level services without recoding. In this sense, the design of RMS is "cleaner". Consistent with this sensibility, Epoch's literature states commitments to Open Systems strategies, describing plans for port of the MSSRM Storage Server from SPARC to IBM, DEC, HP, and SGI architectures.

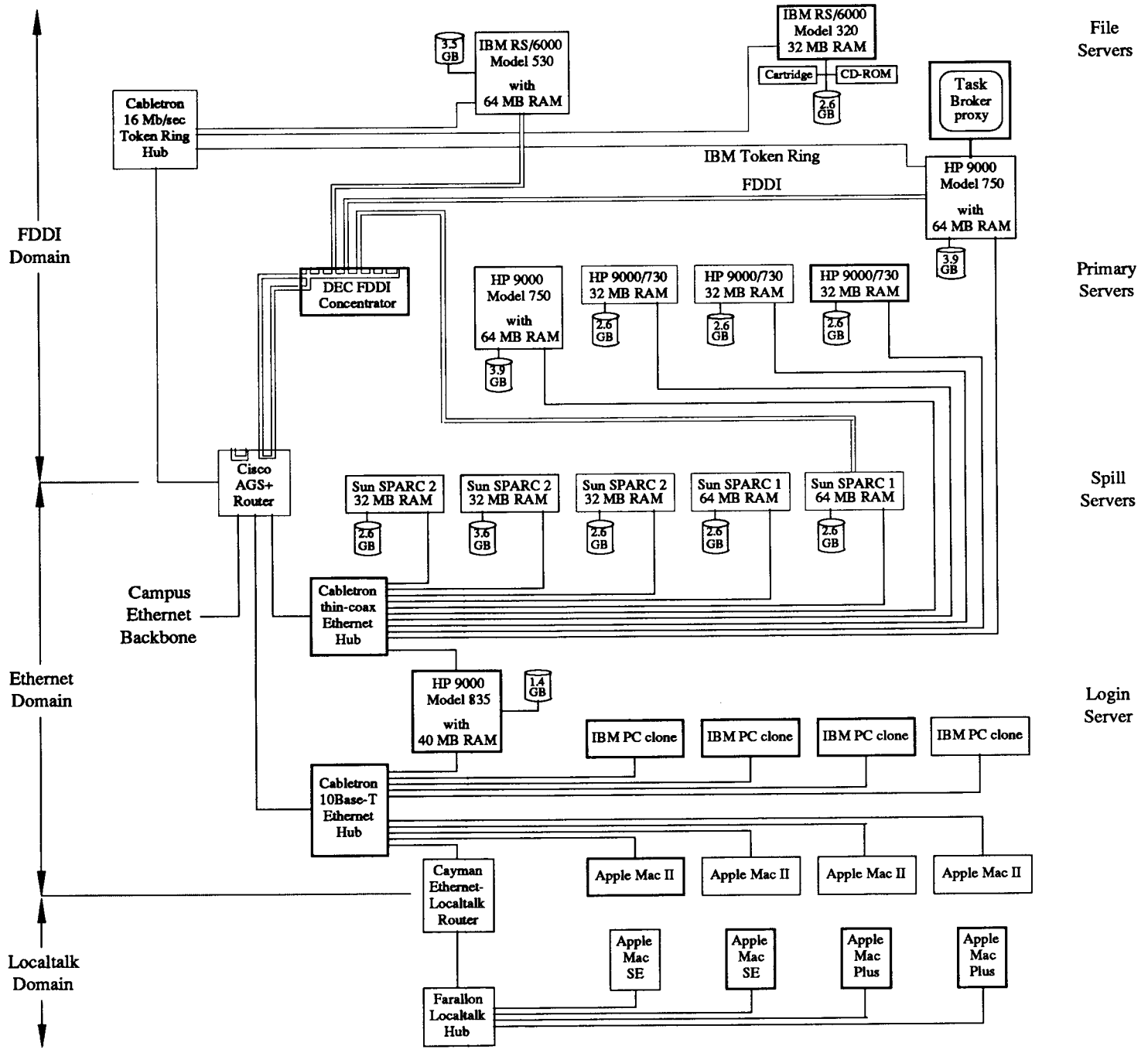


Figure 2: Schematic of Current Social science network with FDDI evaluation

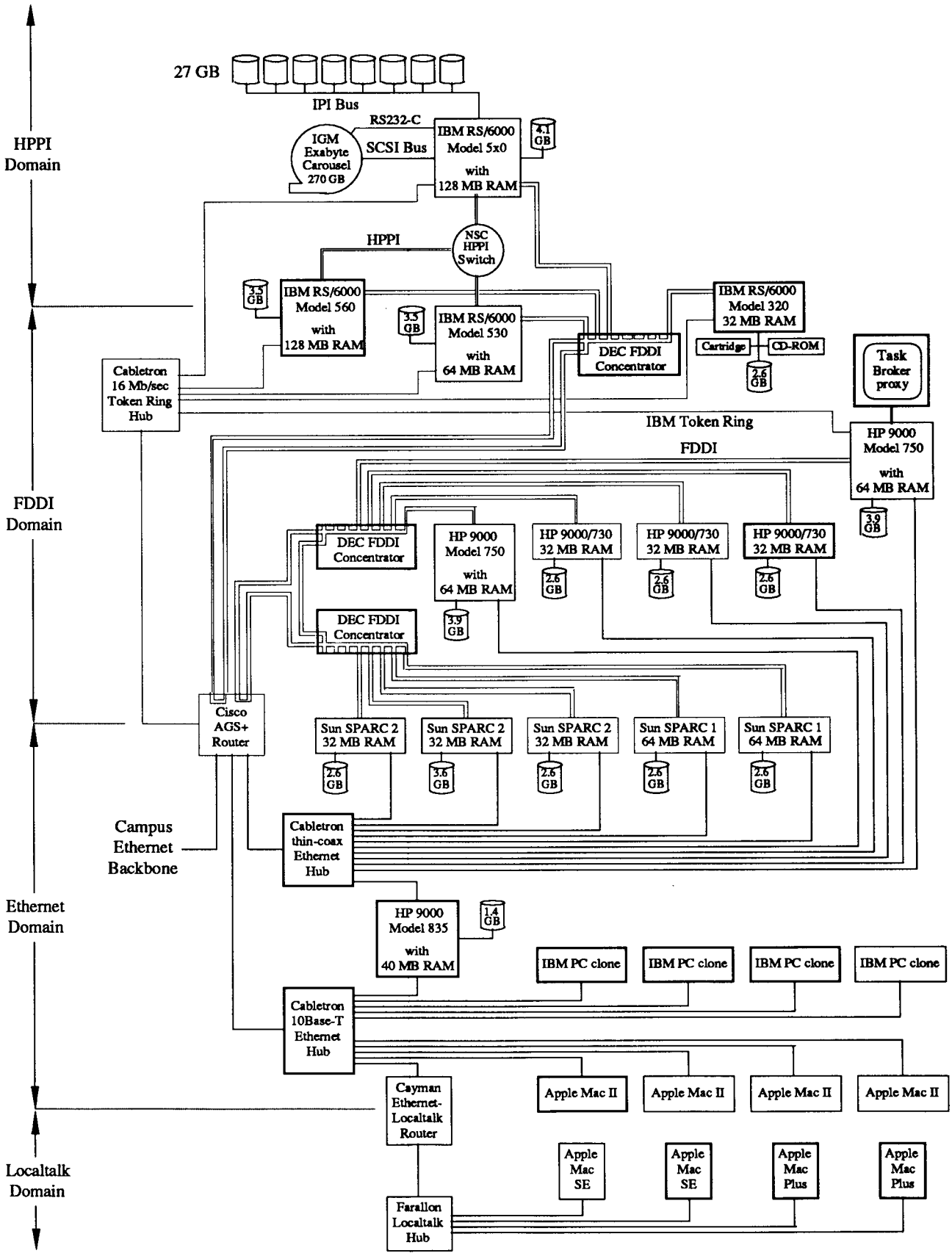


Figure 3: Social Science Network with HPPI and FDDI extensions

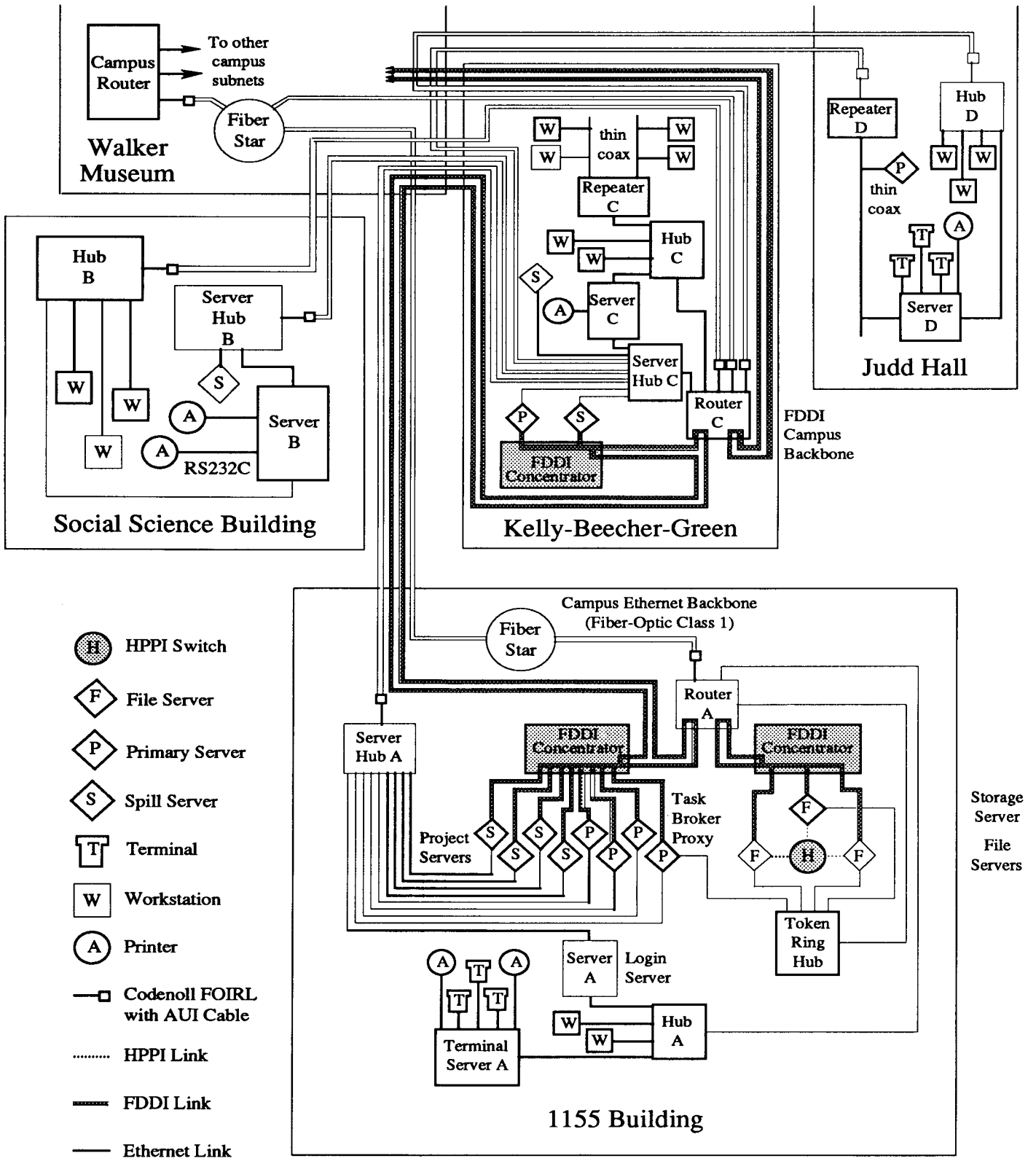


Figure 4. Social Science and Public Policy Network
with Planned Extensions for Fiscal Year 1992-93

The RMS port to IBM POWER architecture is very important because the primary platform identified for the HPPI domain is the IBM RS/6000 running AIX 3.2. The IBM platform is specified for the HPPI domain today because it is the only system which projects the specialized hardware and software requirements of the HPPI domain in the foreseeable calendar. (However, the new DEC Alpha series departmental server might be an alternative, pending further information from DEC.) For required hardware, IBM and third-party suppliers are preparing implementations of HPPI for high-speed inter-computer data communications and IPI-2 interfaces for high-speed disk I/O. Furthermore, the CPU speed of the high-end POWER architecture currently rivals the HP 700 series and is thus suited to intense data-handling tasks, particularly buffer-to-buffer moves and compression. AIX is the only version of UNIX that allows file definitions to span physical volume boundaries so file sizes are not constrained by disk sizes. AIX also automatically maps open data files into *permanent segments* of the virtual memory sub-system, so an intrinsic dynamic RAM-caching for data is always in operation. AIX 3.2 also includes intrinsic disk-mirroring for critical file integrity and deferred I/O for true overlap of I/O and CPU services to a single process. For these reasons, if operational pressures to acquire file migration service force acquisition of RMS on SPARC architecture in the near-term, SSPPCC will pursue a subsequent technical evaluation to determine the wisdom of converting the RMS license to another architecture such as POWER.

There is a disadvantage of the choice for IBM—an apparent lack of near-term plans for multi-processor POWER architecture. Multi-processing is potentially significant in the Storage Server for parallelism of data shipment, data compression, and data uncompression. To explain, one of the primary limitations on effective data transfer rate is the so-called "internal rate" of the disk—the speed at which data bytes are transferred from the disk surface to the disk controller's buffers. A quick strategy for significantly increasing that rate is data compression prior to writing the data, so that internal byte transfers are minimized. Of course, data must then be uncompressed upon reading. I/O interface designers claim that adding available hardware logic for data compression and uncompression onto the controller slows the transfer rate well-below data bus speeds. That explains why hardware compression in the device controller is only found on slow devices like tape drives. Hence, data compression and uncompression are today ideally done by a main processor. Since data compression logic is very CPU-intensive, the parallelism provided by high-performance symmetric multi-processor architecture would probably significantly enhance Storage Server performance. Such UNIX workstation architectures are reported for release by Sun in the SPARC 3 series, HP in the 700 series, and DEC in the Alpha series. Perhaps one of these other vendors will offer the requirements of the HPPI domain in a suitable calendar.

Solutions for relational data base management are weakly researched and unspecified in the SSPPCC design at this time. Major social science databases are not available as relational structures and common-use software mostly requires rectangular flat files, hence most database access in the SSPPCC environment has not used relational queries. (The R–Squared system mentioned in the Overview is one exception.) Nonetheless, relational DBMS strategies are an investigative direction for SSPPCC to be guided by the newly-formed Faculty Database Committee. The current proposed design specifies networked relational database servers supported by file migration service. Initial conversations with Sybase are promising for successful integration of database server and file migration facilities, but essential operational testing will await implementation of the file migration server. In general, SSPPCC believes that wise planning must specify a DBMS server that is supported by an implementation of the MSSRM, but these concepts appear to be weakly considered by the major DBMS vendors today.

Integration of the AFS server facility with MSSRM-based systems is not available today because the current AFS 3.2 database server uses non-standard UNIX filesystem structures on server volumes. However, integration with file migration service appears well-considered by ACSC and Epoch and will probably await shipment of AFS 4.0 as the Distributed File System standard of the OSF 1.0 DCE in 1993.

It should be clear that the SSPPCC design is a comprehensive specification for the technical steps in implementing distributed computing services. Perhaps its most important feature for the social science computing environment at the University of Chicago is the coherence that it brings to technical planning. A large collection of competing vendors' current and future hardware and software offerings is organized for selection by the criteria that emerge from the design specifications.

Wide-Area Extension of the Design

So far, this article has described a strategy for campus-wide large-scale computing services provided by a local multicomputer. Extension of these services to a national (or even international) user community are quickly realizable because existing wide-area networking facilities already provide the essential data communication links and the SSPPCC design intrinsically manages services so that the links can be used efficiently.

The Ethernet domain of the Chicago campus is connected to the Ethernet domains of every campus on the Internet via wide-area links operating at, say, nominal 1 Mbit/sec. Suppose a relationship with a project on a remote campus for use of the SSPPCC multicomputer is established. In the simplest case, project staff could *telnet* to an SSPPCC Login Server and submit jobs. Ideally, if the remote project's computer system is a member of an AFS cell, file structures on SSPPCC AFS servers could be made directly available to the staff through routine AFS file-sharing.

More interestingly, suppose a remote user's workstation is running a Task Broker daemon or is recognized by either a local or SSPPCC Task Broker Proxy Machine. Then SSPPCC multicomputer services could be transparently provided to the user. For example, a remote user might invoke an extraction via SAS from a large data base maintained in the Chicago Storage Server library. Affinity calculations would automatically locate the job in the SSPPCC HPPI domain and, after SAS execution on an SSPPCC File Server, the Task Broker service script could ship the resultant extracted file back to the user's workstation (subject to the file-size constraints and associated procedures described above). The user could prepare the job as if it were to execute locally and might never know that the job was brokered to a remote site.

Now suppose that several such multicomputers are operating on the national Internet and the project has established a service relationship with each of them. Then the user's file extraction would be transparently brokered across the community of multicomputers, each site calculating its affinity based on current load and the availability of the requested source file. (Source files would have to be referenced by generic names that are resolved at each site into local pathnames. The MSSRM Nameserver function is one strategy for such name translation.)

This last example is a scheme for national computerized database libraries accessed transparently by widely distributed projects. Though there are probably political and academic issues, such libraries can be established with relatively low-cost technology as described above. (Most of the long-term costs would be required for support of library staffs.) In general, once a multicomputer is providing services to a local network, extension of those services to a wide-area network is not a difficult technological issue. Such extension accomplishes the epitome of distributed computing services.

[1] Paper presented at the IASSIST 92 Conference held in Madison, Wisconsin, U.S.A. May 26 - 29, 1992. George Yates, Director, Social Science & Public Policy Computing Center (SSPPCC), Uuniversity of Chicago, 1155 East 60th Street, Chicago, Illinois 60637. (312) 702-0793

[2] For a detailed description of the IEEE Reference Model, refer to the document entitled *Mass Storage System Reference Model, Version 4*, edited by Messrs. Sam Coleman and Steve Miller and published by the IEEE Technical Committee on Mass Storage Systems and Technology.