

Design of a Very High-Speed Remote File System

Joseph P. Thomas
John David Cavanaugh
Timothy J. Salo
Minnesota Supercomputer Center, Inc.*

December 14, 1992

Abstract

A new application for gigabit networks, the network supercomputer, is described. Requirements for a very high-speed remote file system to support the network supercomputer are given, and a high-level design is described which meets those requirements.

1 Introduction

The advent of gigabit-per-second wide-area networks (which we refer to as *gigabit networks*) will enable a variety of applications that have previously been infeasible. While it will be some time before all the implications of this technology become apparent, some applications can be envisioned.

One such application is the *network supercomputer*. We use this term to describe a high-performance computing system which is comprised of geographically distributed components communicating with each other via a gigabit network.

The components of high-performance computing systems have, until now, been collocated in the interest of overall system performance. The speed of the channels connecting components of a supercomputer system (for example, the channels connecting mass storage to the processor) has a

*The research described herein was sponsored by DARPA through the U. S. Army Research Office, Department of the Army, contract number DAAL03-91-C-0049. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by DARPA or the U. S. Army.

great effect on the overall performance of the system. Collocation was mandated by the order-of-magnitude differences in speed between computer channels and wide-area networks (e.g., compare an 800 Mb/s HIPPI channel to a 1.5 Mb/s T1 telephone line, or even a 45 Mb/s T3 line). Geographic distribution has been impractical because, prior to gigabit networks, high-performance channels were generally limited to a distance of a few tens of meters.

Gigabit networks, which operate at speeds comparable to those of supercomputer channels, promise to alter the way that networks are used with high-performance computing systems; gigabit networks will enable network supercomputers.

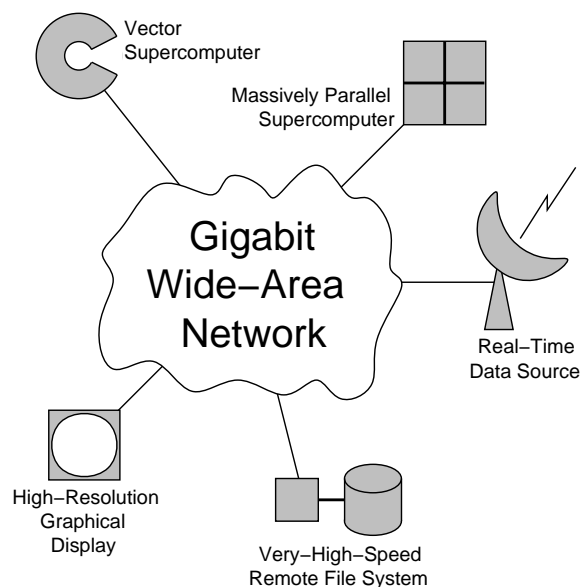


Figure 1: Network Supercomputer

A network supercomputer might be composed of

a number of high-performance components communicating via a gigabit network, as shown in figure 1. Some or all of the following components might be configured into a network supercomputer:

- A vector supercomputer
- A massively parallel supercomputer
- A shared, very high-speed remote file system
- A high-resolution graphical display
- A source of real-time data, perhaps at very high speed

The network supercomputer will enable new methods of high-performance computing. A few of the new computational methods which might be enabled include:

- **A rapidly configured high-performance system:** A researcher could “construct” a supercomputer tailored to his or her specific computational needs by interconnecting network-attached high-performance components. For example, a researcher might dynamically “construct” a network supercomputer by using a gigabit network to connect a vector supercomputer at a university campus with a prototype massively parallel supercomputer at a vendor’s laboratory. These processors might process data (e.g., satellite imagery data only recently received by the ground station) from a shared, very high-speed online archive located at a third site. The researcher could be working at a fourth location which has a high-resolution graphical display.

The network supercomputer would “exist” only for the duration of the computations. When the researcher’s work was complete, the network connections which “created” the network supercomputer would be dropped.

- **Shared, very large, and possibly time-critical data archives:** Some databases are currently difficult (or impossible) to share either because of their size or the time-critical nature of their data. Making copies of these databases is not practical: the size of some very large data

bases makes copying impractical, and the data in time-critical databases could be out of date by the time a copy was made. A shared very high-speed remote file system, however, would enable researchers to use data from remote locations as if they were local. The gigabit network would minimize the degradation of performance caused by accessing the data from a remote location.

- **Rapidly prototyped mixed-architecture supercomputers:** Gigabit networks will enable researchers to quickly prototype high-performance computing systems composed of computational elements with different architectures. For example, a mixed-architecture network supercomputer could be constructed by connecting a vector processor at one location with a massively parallel processor at another. More important, the mixed-architecture network supercomputer might include unique machines which could not be easily collocated with other supercomputers (e.g., one-of-a-kind or prototype machines).
- **Integration of multiple archives:** A network supercomputer could be “constructed” with connections to multiple large remote databases. The supercomputer could integrate data from these archives, perhaps displaying the result in graphical format. For example, a network supercomputer could combine air temperature data from one location with ocean surface temperature data from a second location to assist a researcher in formulating a model of the effects of air-water interaction on weather systems.

One fundamental limitation on the performance of a network supercomputer is imposed by propagation delay. Network supercomputer components could be separated by wide distances, so there would be a delay in communication while signals propagate between them (at the speed of light, a signal takes approximately 15 ms to cross the USA from coast to coast). Some activities would be affected more than others. For example, an algorithm that required a lot of short, synchronous communication between machines would be affected more than one which sent large messages without requiring synchronization.

Most of the components of the network supercomputer are now available, or are under construction. Both vector and massively parallel supercomputers are available, and they already have network interfaces operating at high speed (e.g., HIPPI interfaces at 800 Mb/s). Experimental gigabit networks are being constructed, and are the focus of much research.

The component that remains to be studied and constructed is the very high-speed remote file system. Requirements and design criteria for such a file system are discussed in the following sections.

2 Requirements

This section defines the requirements that a very high-speed remote file system (VHSRFS) must meet in order to be useful in a gigabit network (e.g., as part of a network supercomputer). A very high-speed remote file system must have these attributes:

- Very high speed
- Shared
- Expandable
- Standard interfaces
- Cost-effective

Very high speed: There are two speeds which are important to the VHSRFS. The first is the I/O speed of supercomputers. Supercomputers typically have very high-speed I/O channels, for example, HIPPI channels at 800 or 1600 Mb/s. Since supercomputers are envisioned to be important users of the VHSRFS, it should be fast enough to allow supercomputers to access data at supercomputer speed.

The second is the effective speed of wide-area networks. Although the fastest data links in common use today are 45 Mb/s, faster links are on the horizon. It will not be long before data networks at 155 Mb/s and even 622 Mb/s are commercially available. The VHSRFS should be fast enough to be able to fully utilize the capacity of these new high-speed networks.

Therefore, the VHSRFS must be able to transfer data to supercomputers across the network at 600–800 Mb/s.

Although a gigabit network can match the speed of a supercomputer I/O channel, it will have a longer propagation delay, since the length of the channel is measured in hundreds of kilometers instead of in meters. The VHSRFS must implement measures designed to minimize the effect of the longer propagation delay.

Shared: The VHSRFS must be shared simultaneously among a number of network-connected machines. These machines may be of different architectures, and may use different operating systems. Further, they may want to make use of the VHSRFS at different times or all at once.

There are two models for remote mass storage:

- A remote disk drive
- A remote file system

The remote disk drive relies on the user's system to provide meaning and structure to the data. User requests to a remote disk drive are in the form of "read or write data blocks M thru N". The extent of processing handled by the remote system is limited to the actual data transfer. While this approach simplifies the remote disk's design, it introduces a serious drawback: every host which uses the remote disk must have the same idea of its content.

In the case of a VHSRFS server, hosts make requests for file blocks as opposed to disk blocks. Since only one machine, the server, needs to know the disk layout, information about data allocation and layout can be hidden from the client machines by the server. In fact, the layout can change without the clients ever being aware. Another advantage of a file system model versus a disk drive model is that the file system can handle issues of data consistency. Where necessary, the file server can arbitrate access and can prevent destruction of data by one host while another is still accessing it.

Expandable: The VHSRFS must be available in a range of sizes. Some databases might need only

a few gigabytes of storage, while others might need a terabyte or more. Adding more storage (e.g., when a database overflows the available storage) to a VHSRFS should be straightforward.

The largest VHSRFS must be able to accommodate databases of tens to hundreds of terabytes.

Standard interfaces: One attraction of the network supercomputer is its ability to take many forms by using many different machines; it is crucial that all these different machines be able to make effective use of the VHSRFS.

To allow this, the VHSRFS must present a standard interface to the network. It must use standard file system protocols and standard network protocols (e.g., ATM, AAL5, IP, TCP, UDP). It must attach to the network through a standard physical interface (e.g., HIPPI, SONET). This will allow the VHSRFS to converse with supercomputers using standard software and readily-available network technologies.

Cost-effective: The VHSRFS must provide a cost-effective method of storing data. It must not be too expensive (e.g., compared to existing supercomputer disks), while still providing performance, expandability, and capacity as noted above. The VHSRFS can help make the overall networking environment more cost-effective by reducing the need for redundant copies of data (and the need for storage for those copies).

3 Design

This section proposes a high-level VHSRFS design capable of meeting the stated requirements. We propose that the VHSRFS be built using a large-memory, high-performance processor with HIPPI-connected RAID (Redundant Arrays of Inexpensive Disks) mass storage technology. This design utilizes readily-available hardware and reduces the risks during system integration.

3.1 Architecture

The proposed VHSRFS architecture (Figure 2) consists of:

- A high-performance processor
- Large memory
- RAID storage subsystem

Most of these parts are available in commercial products. High-performance processors (e.g., superminicomputers, or small supercomputers) are offered by a number of vendors. They are available with enough memory to meet the requirements of a VHSRFS. RAID storage subsystems are offered commercially (e.g., by Maximum Strategy), and HIPPI interfaces for them are available.

One part which is missing is a network interface which can attach the VHSRFS to a high-speed wide-area network. HIPPI interfaces are available on high-performance processors, but there is currently no way to attach such a processor to a wide-area network. A gateway with both HIPPI and SONET/ATM interfaces is not currently available (although efforts in this arena are underway), and SONET/ATM interfaces at 622 Mb/s are not available for high-performance processors.

3.1.1 High-Performance Processor

We claim that the VHSRFS server must have processing power to meet the requirements of shared access, standard interfaces, and very high speed.

The requirement to support shared access is met by implementing a file system protocol designed around the IEEE Mass Storage System Reference Model (MSSRM) [1]. The issues of selecting and implementing file systems protocols are discussed in a companion paper, *Technologies for Gigabit Distributed File Systems* [4]. The MSSRM enumerates several tasks requiring processing capability on the part of the server. These include:

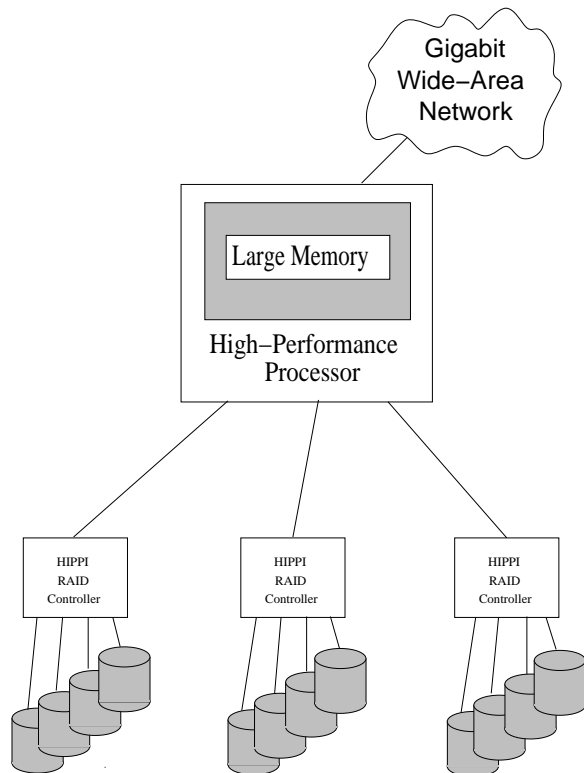


Figure 2: VHSRFS Server

- Mapping client names to VHSRFS storage:** The MSSRM provides a global name space which is location independent. The *Name Server* module in the MSSRM maps names to physical storage locations. The name mapping allows the VHSRFS server to rearrange storage for space or access time needs independent of client knowledge.
- Data buffering:** The VHSRFS server can improve performance by implementing buffering. When clients request data, the server can attempt to predict the clients next request and “read ahead” some amount of data. Output data can also be buffered in an attempt to write one large buffer in favor of several small buffers.
- Access arbitration:** When several clients attempt to access the same piece of data, the VHSRFS server must implement an access policy to guarantee data consistency. This is important for cases where one client is writing data that a second client is

attempting to read.

- Access security:** In a shared environment, the MSSRM needs to provide data security such that unauthorized clients cannot gain access to sensitive data.

The requirement for standard interfaces states that the VHSRFS will present a standard interface to the network, i.e. it will use the TCP/IP protocols. It is recognized that networks with large $bandwidth \times delay$ products, called Long Fat Networks (LFNs, pronounced “elephan(t)s”), introduce TCP performance problems (see [2]). As the network community researches these problems, protocol changes will be introduced to provide performance enhancements. The use of a processor within the VHSRFS will allow the system to rapidly adapt and provide higher overall system performance.

The need for a high-performance processor is a result of the network supercomputer model. We observe that one likely class of users of a large VHSRFS system will be supercomputer-class machines. These machines will be capable of sustaining very large, high-speed data transfers. If the VHSRFS system is to maintain pace with these machines, especially when multiple clients are making simultaneous requests, the VHSRFS server must be capable of accepting and transmitting data to clients at supercomputer channel speeds (600-800 Mb/s). Furthermore, while client requests are received and replied to over the network, the server must be accessing the storage repository at these same speeds. This model implies that the VHSRFS server must be able to interface to a high-speed network based on interfaces such as HIPPI or SONET/ATM, and must also support very large amounts of data storage at supercomputer transfer rates (such as HIPPI connected disks). We make the observation that a general purpose supercomputer, that is, a supercomputer with fast scalar performance but without the need for vector processors or parallelization, meets the above requirements.

3.1.2 Large Memory

The VHSRFS server requires a very large memory pool in order to perform at the required

level. Large amounts of memory are required for buffering, both for the storage subsystem and the network interface.

The server's performance will be limited by the time it takes to read and write data from the storage system. Common file system techniques employ buffering to reduce this time. When read requests are received, the server can often correctly anticipate that the next read request from this client will be the data following the data just read. If the server were to read the data ahead, that is, read into memory more data than was requested for the transfer, there is a good chance that the next read request could be satisfied from buffer memory rather than requiring storage access. In the same manner, the server can anticipate that a write request will follow the last write request and delay physically writing the data to storage until a larger buffer has been filled.

Similarly, file system throughput can be enhanced by buffering the name server translations. As client requests are received, the name server module maps client names into storage locations. By caching frequently-used and most-recently-used translations, translations can be satisfied from table lookups rather than by a (perhaps) long computational process.

The network interface also requires large amounts of buffer storage. The network supercomputer model has multiple supercomputers accessing a small number of shared data archives. This implies that the VHSRFS must first support a large number of client connections, and that data transfers will tend to be very large to satisfy supercomputer needs. From a protocol standpoint, this leads to the need for a large amount of memory buffer to support receive and transmit queues, fragmentation and reassembly queues, and retransmission queues. If we consider some of the proposed TCP extensions to improve performance on LFNS [2], then we also have additional needs for large memory. One of the observed limitations is the inability to keep the network path fully utilized. The solution, that of increasing the window size, means that an even larger amount of data may be queued per each network connection. Congestion and data loss in the network also leads to the necessity for large memory. If any one cell (ATM layer) or fragment/packet (IP layer) is lost, the entire

window of data may need retransmission. If the window has been scaled to improve throughput, the data required to be held in memory similarly increases.

How much memory is required for all these buffers? We estimate that file system buffers are likely to absorb about a megabyte of memory per active user. We estimate that the retransmission queue for the network interface will require 3–5 megabytes, and that the transmit and receive queues could require another megabyte per active user. For a moderate number of users (e.g., 10 users), the VHSRFS would require around 25 megabytes of buffer storage.

3.1.3 RAID Storage Subsystem

A RAID-based storage subsystem is recommended because of its ability to provide high-performance, fault-tolerant data storage at a reasonable cost.

RAID is a concept which an array of disk drives working in parallel is used to improve data throughput, improve reliability, and reduce cost. Whereas a single disk might provide an 8-bit data path with a 2 MB/s data throughput, four of those same drives working in parallel can provide a 32-bit data path with 8 MB/s of data throughput. Since a larger number of components leads to a higher risk that one of those components will fail sooner than a single component would, RAID also employs some type of data redundancy to provide system level fault tolerance. This redundancy means that a RAID subsystem can maintain a high MTBF, even though the MTBF of the sum of its parts is low.

There are five different types, or levels, of RAID.

RAID level 1 or RAID 1, provides fault tolerance through disk mirroring. At this level, two identical copies of the data are kept on two physically separate disks. Failure of one disk leaves the "mirrored" copy intact. The disadvantage of RAID 1 is 50 percent utilization. For every disk used to store data, another full disk is needed to mirror it. This means that a level 1 RAID subsystem is expensive, and does not provide a significant improvement in throughput.

RAID 2 uses data striping with an interleaved Hamming code. Here, each bit of data is written to a separate disk. Interleaved with the data bits is a Hamming code capable of correcting single bit errors and detecting double bit errors.

Disadvantages of RAID 2 include the need to compute complex Hamming codes, and the large number of check bits needed. With 10 data drives, four check bit drives would be needed, for 71 percent utilization. As the number of data bits drops below 10, the utilization can drop below the 50 percent level.

RAID 3 draws on the observation that many of today's drives already incorporate ECCs (error-correction code's) making Hamming codes redundant. Instead, RAID 3 stores only enough information to correct a single failed drive. RAID 3 uses bit level striping with one check drive containing an XORed parity bit. If any one drive fails, the data can be reconstructed by XORing the remaining data bits with the check bit. One disadvantage of RAID 3 is that each drive transfers one sector for a N -wide data transfer unit. As N starts to grow to even small numbers, the transfer unit size grows and small I/O begins to lose performance. The penalty is greater for small writes because the entire transfer unit must be read, a small portion modified, and the entire transfer unit written back to disk.

RAID 4 attempts to overcome the read performance limitation of RAID 3 by using sector striping instead of bit striping. With sector striping, unrelated sectors may be read off different drives simultaneously. That is, when reading, the drives in the array may be at different sector locations simultaneously. RAID 4 still imposes the write penalty since there is only one check disk. This limits the system to performing write operations one at a time.

RAID 5 removes the write bottleneck from RAID 4 by distributing the check disk function among several drives. In RAID 5, disk one may be the check disk to sector one while disks two through five are the data disks. The check disk for sector two might be disk three, while disks one, two, four and five are the data disks. Since the XOR nature of the check disk requires that only two disks be written too, the data disk and the check disk, if the write operation involves different disks, several write operations may take place simultaneously.

We recommend that a VHSRFS employ a level 5 RAID storage subsystem. It offers the best performance, and commercial products are available (e.g., from Maximum Strategy).

3.2 Satisfying the Requirements

This section briefly covers the stated requirements and how they are met by the proposed VHSRFS design.

Very high-speed: A VHSRFS design requires high speed in multiple areas. Tasks to be simultaneously carried out at supercomputer speeds include:

- Accept input requests and data from the network
- Perform file system protocol processing
- Transfer data to and from mass storage
- Send responses and data to the network

As wide-area networks capable of supporting supercomputers (i.e. 600–800 Mb/s) are implemented, the overall demands on the server also increase.

The selection of a high-performance processor with large amounts of memory allow the VHSRFS server to concurrently handle high network demands while supporting very fast storage access.

Shared access: To support the shared access requirement, a high-performance processor is recommended. Use of a processor allows the VHSRFS to become a programable system. This programability is the basis for implementation of network and file system protocols, something that is not possible on high-speed disk controllers. Programability allows for the growth and development of protocols as the VHSRFS concept matures. It is recognized that current TCP/IP protocols have limitations for high *bandwidth* \times *delay* networks. Extensions have been proposed but will need to be implemented to surmount these issues. Similarly, there are likely to be issues in file system performance which will need to be dealt with.

Standard interfaces: A goal of the VHSRFS is to provide a new service to existing users. The use of standard protocols, both network and file system, will achieve this goal. The VHSRFS needs to support the TCP/IP protocol suite for network access. Physical interfaces need to support the high bandwidth of the network and may include both HIPPI and SONET/ATM interfaces. A standard file system protocol will allow the VHSRFS to be shared among a large number of remote hosts. Standard interfaces also allow connection with more traditional networks, such as FDDI, in order to provide access to tape drives and other low-performance storage media.

Expandable: A goal of the VHSRFS is to support databases of tens to hundreds of terabytes in size. RAID technology allows the storage capacity to grow by the addition of inexpensive disks. Use of RAID technology also allows the storage media to provide the high-bandwidth data transfers required. As the demands on the VHSRFS server grow, the use of a large-memory, high-performance processor will allow I/O bandwidth to grow, along with providing support for higher demands on the file system and network protocols.

Cost-effective: Cost objectives can be met with the selection of RAID storage supported by a high-performance processor using standard interface technology. The use of RAID provides a reliable, low-cost, high-bandwidth storage media. The VHSRFS server requires a processor capable of maintaining supercomputer speeds, but does not require the special hardware, such as vector processing, often associated with supercomputers. By selecting TCP/IP as the network protocol, and using HIPPI as a channel both for the network and for the storage system, costs can be controlled by not requiring the development or purchase of unique, one-of-a-kind, software or hardware.

4 Summary

Gigabit networks promise to enable a number of applications that are currently infeasible. To support these applications, a new shared very high-performance remote file system is required.

A VHSRFS should be very fast, sharable, expandable, have a standard network interface, and be cost-effective. We propose a high-level design for a VHSRFS based on a large-memory, high-performance processor and RAID technology. This design has provisions to meet these requirements. Most of the components of the design are available commercially.

A Acronyms

AAL	ATM adaptation layer
AAL5	ATM adaptation layer 5
Gb/s	gigabits per second
HIPPI	high-performance parallel interface
I/O	input/output
IP	Internet Protocol
Mb/s	megabits per second
MB/s	megabytes per second
ms	milliseconds
MSSRM	mass storage system reference model
MTBF	mean time between failures
RAID	redundant array of inexpensive disks
SONET	synchronous optical network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VHSRFS	Very High-Speed Remote File System

References

- [1] Coleman, S., and S. Miller, Editors. *Mass Storage System Reference Model*. IEEE, Ver. 4, May 1990.
- [2] Jacobson, V., R. Braden, and D. Borman. *TCP Extensions for High Performance*. RFC 1323, May 1992.
- [3] Alford, Roger. *Disk Arrays Explained*. BYTE, October 1992.
- [4] Spengler, Michael K., and Timothy J. Salo. *Technologies for Gigabit Distributed File Systems*. Minnesota Supercomputer Center, Inc.