5140062

An Implementation Plan for NFS[1] at NASA's NAS Facility
(Revision 3.1)

Terance L. Lam
Computer Sciences Corporation
Numerical Aerodynamic Simulation Division
NASA Ames Research Center
October 10, 1990

ABSTRACT

This document discusses how NASA's NAS can benefit from NFS. A case study is presented to demonstrate the effects of NFS on the NAS supercomputing environment. Potential problems are addressed and an implementation strategy is proposed.

1.0 Introduction

The Sun Microsystems Network File System (NFS) has become a popular standard because it allows users to transparently access files across heterogeneous networks. NFS supports a spectrum of network topologies, from small, simple and homogeneous, to large, complex, multi-vendor networks. In NASA's NAS computing environment, many different computing resources are available to users. Currently, users have information scattered across many different computer systems. This information has to be copied to a particular computer and then converted to the appropriate formats before it can be used locally. The current NAS nfsmount and nfsumount implementations allow users to mount and unmount remote file systems between workstations. If the NFS utilities could be modified to work with the Cray[2] computers, users would no longer need to perform the tedious file transfer operations manually. Once a remote file system is nfsmounted, it can be manipulated as a local file system. But the effects of NFS on this computing environment should be investigated so that any potential problems can be identified. This paper will outline the basic

---

1. NFS is a registered trademark of Sun Microsystems, Inc.
2. Cray is a registered trademark of Cray Research, Inc.

NFS architecture, identify typical performance problems, and recommend specific solutions.

The first phase of this project is modifying the nfsmount and nfsumount utilities to work with the Cray computers. The second phase verifies NFS functionality on the Crays using Sun Microsystems' NFS test suite. The third part involves a case study to determine the worst-case scenario caused by NFS within the NAS computing environment. These potential problems will be addressed in the last section to make NFS a winning case in NAS.

2.0 Background

In order to study the effects of NFS on the NAS computing environment, some background of NFS, NAS networking environment, its resources utilization, and the nfsmount utilities will be helpful.

2.1 NFS

The Network File System is a utility for sharing files in a heterogeneous environment of machines, operating systems, and networks. Sharing is accomplished by mounting a remote file system on a local file system, then reading or writing files in place. The NFS protocol is designed to be machine, operating system, network architecture, and transport protocol independent. This independence is achieved through the use of Remote Procedure Call (RPC) primitives built on top of an External Data Representation (XDR). The supporting mount protocol allows the server to hand out remote access privilege to a restricted set of clients. It performs the operating-system-specific functions that allow, for example, to attach remote directory trees to some local file systems.

Figure 1 depicts a typical NFS environment: one server supporting several clients connected via Ethernet[3]. The server manages the shared resources such as data files and applications. The server is also responsible for the multiplexing of its resources among the various clients. The server must also maintain and protect the data within these shared resources.

---

3. Ethernet is a registered trademark of Xerox Corporation.

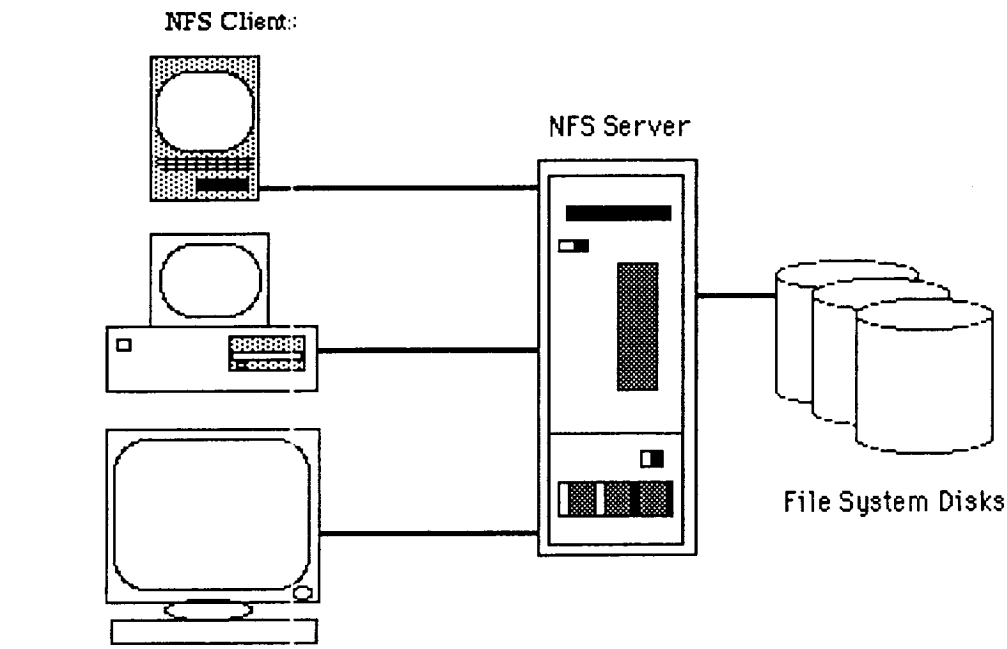

Figure 1.The NFS Environment

Some of the advantages offered by NFS are:

- To the users, all nfsmounted file systems have no apparent difference from a local disk. Users are able to access remote information without knowing the network address of where the data resides. Information on the network is truly distributed.
- NFS offers an extensible set of protocols for data exchange, and allows computers of different operating system to be integrated to the network.
- NFS provides a flexible, operating-system-independent platform for software integration. Software from different vendors can be integrated easily.

- NIS (Network Information Service), a NFS-based network database service, allows the UNIX maintenance commands to be adapted and extended for the purpose of network and machine administration. NIS also allows certain aspects of network administration to be centralized on a small number of file servers.
- NFS inherited the robustness of the 4.2 BSD file system. This stateless protocol and its daemon-based methodology also provides file and record locking capability. Should a client fail, the server can maintain its functional state. Should a server or a network fail, it is not necessary that the clients continue to attempt to complete NFS operations until the server or the network returns to its functional state.

2.2 NAS Computing Environment

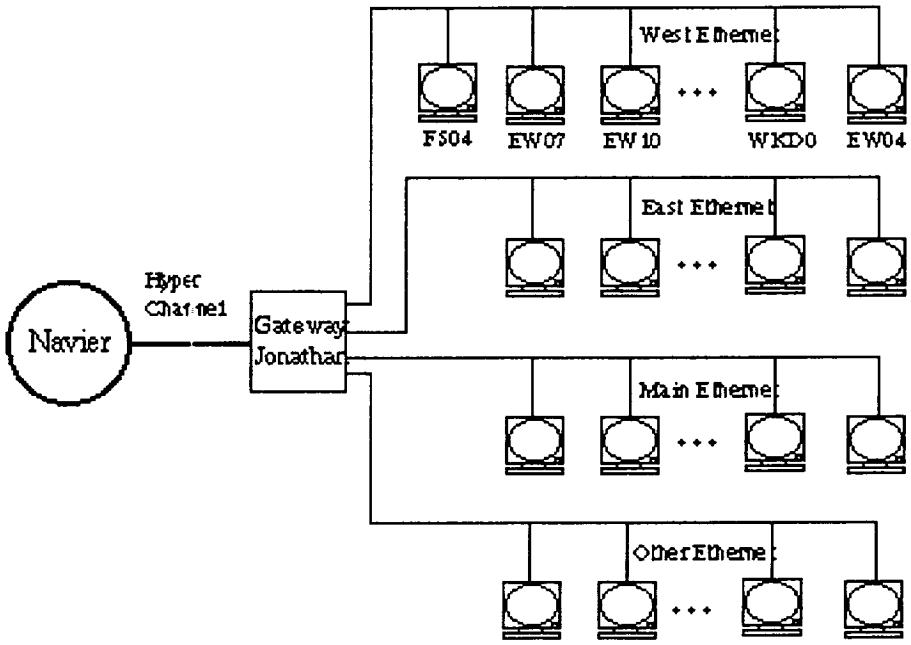

Figure 2.A Simplified NAS Network Diagram.

Figure 2 is a simplified network diagram of the NAS computing environment. Currently, there are more than 200 workstations on the network; most are from Sun and SGI. The majority of these workstations are Ethernet-based. They are grouped into East, West, Main and Other subnets. Each subnet is capable of connecting up to 1,024 computers. These workstations are networked to the Crays through Ethernet (10 Mb/s) and Hyperchannel[4] (50 Mb/s). If engineering workstation EW07 on the West subnet has to communicate with the Crays, it can send information through the West Ethernet, a gateway (Jonathan in this

4. Hyperchannel is a registered trademark of Network Systems, Inc.

case) and then through the Hyperchannel.

A majority of these workstations in NAS are equipped with a local disk drive with capacity of 380 MBytes or less. With disk space taken up by the operating system, about 130 MBytes of usable disk space is left for the users. Due to the limited local storage, users' information often has to be off-loaded to a file server or a remote computer. Also certain data can only be generated on the Cray computers. Each time this data is needed, it has to be manually copied back to the local workstation across the network using RCP or FTP. These file transfers take up users' time, CPU time, and network time.

Currently, NFS is only publicly available between the two Cray computers. If NFS became available to users between the two Cray computers and the workstations, NAS could benefit from the following changes:

- NFS would allow information stored on the Cray file systems to be used locally on workstations in a transparent manner. Users would no longer need to copy data from remote file systems.
- Solution analysis could be performed on a workstation with data retrieved from the Cray file systems. This would free up the remote supercomputer, provided the analysis was more CPU intensive than the I/O intensive.
- A Cray remote file system could be mounted onto several workstations. Different users could share information on the same remote file system. Local copies of this information could be avoided and file system maintenance would become easy because some of these heavily-used data files could be centralized in a certain location and shared by many users.

2.3 Nfsmount and Nfsumount

Nfsmount and nfsumount allow file systems to be cross mounted between workstations without the need of special privileges. The remote file systems are then used as local file systems. The syntax for these utilities are :

```
nfsmount hostname        # e.g. nfsmount  ew07
nfsumount hostname       # e.g. nfsumount ew07
```

Nfsmount will perform the following :

- Find the user's home directory on the remote file system.
- Create a local mount point.
- Mount the remote file system onto the local workstation.

Nfsumount will perform the following :

- Find the user's directory on the remote file system.
- Unmount the remote file system from the local workstation.
- Remove the remote file systems mount point.

These utilities have been modified to allow Cray file systems to be mounted on workstations. The syntax and operation of these commands remain unchanged. The changes are that nfsmount will now mount both the user's home directory and scratch directory. Instead of mounting the file systems to /r/remote_host/wk, nfsmount actually issues the following system calls:

mount remote_host:/u/disk_part/user /r/remote_host/disk_part/user
mount remote_host:/u/scratch/user /r/remote_host/scratch/user

A remote file system can be hard-mounted or soft-mounted. A hard-mounted file system causes client requests to retry until the file server responds. If the server fails to respond, a "nfs server not responding" message will be returned to the client. A soft mounted file system returns a "connection timed out" error after trying a finite number of times and giving up. A soft mount option is used in the current nfsmount implementation so that the client will not hang when the file server is down or the network is broken.

3.0 NFS Performance Evaluation

File systems on the Cray computers can currently be mounted onto workstations for testing, but functionality and performance are a concern because NFS was not functional on Crays' UNICOS[5] prior to release 5.1. It is advised that a plan be implemented to verify NFS' func-

tionality and reliability in future releases of the UNICOS operating system. Also, the effects of NFS on the NAS computing environment should be investigated before it is made available to users.

The NFS performance evaluation is divided into three parts. It first performs a NFS functional verification on the UNICOS 5.1; it then compares the efficiency of NFS versus the traditional methods of acquiring data from the remote hosts. The last part performs a case study to determine the impacts of NFS on the NAS computing environment.

3.1 Functional Test

Sun Microsystems has developed a NFS functionality test suite which can be used to exercise different areas of NFS on a computer. The test suite is divided into basic, general and special sections as shown in table 1.

This test suite was ported to a SGI 4D/60 workstation. All three sections were applied to the file systems on the Cray Y-MP and the Cray -2[6] while nfsmounted to a workstation. In a recent UNICOS upgrade to 5.1.8, NFS on Navier and Reynolds experienced serious failures. When a client accessed a file residing on the NFS server, the file attributes were set improperly, even on a READ_ONLY file system. The owner id and the group id were erroneously set to -1 which indicated that the owner of the file is not recognized. These modifications were removed and NFS was restored to its functional state. Most of the tests in this test suite passed except for symlink, tbl and nroff; therefore, Navier and Reynolds are still considered as having passed the NFS verification. See section 3.1.1 and 3.1.2 for the details.

Basic tests	General tests	Special tests
file and directory creation	small compile	open/unlink

5. UNICOS is a registered trademark of Cray Research , Inc.
6. Cray Y-MP and Cray-2 are registered trademarks of Cray Research, Inc.

file and directory removal		tbl	open/chmod
lookups across mount point		nroff	lost reply on requests
setattr, getattr, and lookup		large compile	exclusive create
read and write		simultaneous large compiles	negative seek
readdir			rename
link and rename			sparse file read/write
symlink and readlink			Holey file test
statfs			

Table 1.Basic, General and Special Tests

3.1.1 Symlink

Creation of symbolic links to Cray-2 and Cray Y-MP files failed. The error returned was :

```
test8: symlink and readlink
test8: (/r/navier/nb/lam/test/testdir)
    can't make symlink file.0 : io error
```

Because UNICOS 5.1 is a System V[7] Release 3 based operating system, symlink is not supported in this release of System V. Refer to the AT&T System V Release 3 Definitions (SVID3) for more details. Systems failing the symlink test can still be considered as passing the test suite. Appendix A and B are lists of tests and results of Navier's and Reynolds' verification.

3.1.2 tbl and nroff

Tbl and nroff requests to Navier and Reynolds also failed. The errors were:

```
tbl
stat: bad data format in tbl.time (Permission denied: tbl)

nroff
stat: bad data format in nroff.time (No such file or directory: nroff)
```

These tests failed simply because text formatting utilities are not available on these supercomputers.

3.2 NFS Case Study

File servers make their file systems available to clients by satisfying several types of requests. These include reading data, writing data, looking up files and returning file status. This is not very different from requests made by a local time-sharing user on a local file system except that requests have been directed through a network and some layers of protocol. In the case of Sun's NFS, for example, requests pass through NFS, RPC (remote procedure call), XDR (external data representation), UDP/IP and an Ethernet (and/or other media), in addition to the normal file system mechanism of the server. Given the diversity and complexity of NFS environment, isolating problems can be difficult. In this case study, a "black box" approach is accomplished by running the nfsstones benchmark on a varied number of clients. The detailed analysis of individual layers of the networked file system and its underlying protocols are avoided on purpose. It looks at the NAS computing envi-

---

7. System V is a registered trademark of AT&T.

ronment as three components: the NFS server (Navier), the clients (4D workstations) and the networks.

This study attempts to gain an understanding of the worst case scenario of NFS in NAS. NFS will be stressed in this case study to identify which one of its components in this environment will be affected the most so that corrective actions can be suggested. NFS activities probably would not be as heavy as that in this case study. Before we go on with the case study, some background of the nfsstones benchmark, TTCP and traffic utilities are necessary. These tools are used to determine the capability of each component in this environment.

3.2.1 Nfsstones

Nfsstones is a network file server performance benchmark developed by Encore Computer Corporation. This program is designed to be portable between different NFS platforms. Nfsstones can be thought of in terms of NFS operations per second, where NFS operations represent a mixture of requests composed of lookup, read, readlink, getattr, write and create, etc. This benchmark emulates the NFS model, presented by Sandberg[8], which was tuned to reflect what is believed to be an NFS environment under normal usage (although compressed into a small time). The nfsstone developers believe that the figures obtained empirically by observing kernel meters after a single run of this nfsstones benchmark are close enough to match Sandberg's figures obtained by compiling the nfsstat statistics.

NFS operation	Sandberg %	nfsstones %
lookup	50	53.0
read	30	32.0

8. Sandberg, R., "The Sun Network File system: Design, Implementation and Experience", Sun Technical Report. A version also appeared in the USENIX Summer 1985 Conference Proceedings, pp. 119-130, although not with the appendix of NFS operations referenced.

readlink	7	7.5
getattr	5	2.3
write	3	3.2
create	1	1.4

Table 2. Distribution of NFS Request in Nfsstones

This benchmark program was ported to the SGI 4D workstations, and used as a means to measure the NFS performance. The reason for choosing this program as the measurement tool is simply that it creates a tremendous number of NFS requests. The nfsstones is used as a tool to stress the server in a manner which is reasonably representative of the kind of load seen during very heavy usage.

The typical nfsstones performance on a 4D/70 and the Cray-2 are shown in table 3. A 4D/60 was used as a NFS client to deliver NFS requests at full capacity. Both the Cray-2 and the 4D/70 are faster than the 4D/60; they are capable of accepting all traffic delivered by the 4D/60. Using a Cray-2 as the NFS server, the performance is 64 nfsstones per second. Using a 4D/70 workstation as the server, the result is 55 nfsstones per second. The performance difference is a result of the higher NFS server performance of the Cray-2 over the 4D/70. We will only see a performance difference as the number of NFS clients increases, that is, the number of NFS requests increases.

Client	SGI 4D/70 as server	Cray-2 as server
4D/60	55 stones	64 stones

Table 3. nfsstones performance

3.2.2 TTCP

TTCP (Test TCP Connection) is a public-domain program developed and modified by T.C. Slattery of USNA, Mike Muuss of BRL, and Silicon Graphics, Inc. This program makes a connection on port 5001 and transfers fabricated buffers or data from stdin. It transfers data to a remote host using a protocol (TCP or UDP) specified by the user, and returns transfer statistics. TTCP was ported to the 4D workstation and Navier, and loop-back tests employing UDP protocol were applied to WK202, EW07, WKD0 and Navier. A loop back test sends data from one computer to itself going through all the network protocol layers. It measures the I/O transfer rate of a computer without actually sending data through the network. The transfer rate is the theoretical maximum rate at which a computer can send data to a computer network. Table 4 shows the results of these loop-back tests.

Computer	KB/sec	Mb/sec
4D/60	638	5.12
4D/70	1,432	11.46
4D/320 VGX	2,843	22.74
Cray-2	8,192	65.50

Table 4. I/O Transfer Rate for the SGI 4D/60, the 4D/70, the 4D/320, and the Cray-2.

3.2.3 Traffic

Traffic is a SunView program that graphically displays Ethernet traffic. It gathers statistics from etherd (8C), running on a host machine. The tool is divided into subwindows, each giving a different view of the network traffic. This program is capable of displaying information on traffic load, size, protocol, source and destination. We are only interested in the traffic load; this feature will be discussed.

Traffic load is represented as a strip chart. The maximal value of the graph represents a load of 100%, that is, 10 Mb/s on the Ethernet. The West Ethernet traffic was monitored by FS04 at different times during normal business hours. Normal business hours are between 9:00 am to 5:00 pm when the production machines are in interactive use. Figure 3 shows one of the traffic displays sampled during normal business hours. The data shows that the nominal West Ethernet utilization stays below 10% of capacity most of the time.

3.2.4 Case Study Set Up

Navier, a Cray-2 supercomputer with 4 CPUs each running at 250 MHZ, was used as the NFS server in this study. A total of 8 SGI 4D clients in the West Ethernet were employed to run the nfsstones benchmark simultaneously, in order to produce the worst-case scenario. Each SGI workstation has an ESDI hard disk with an I/O transfer rate of 1.5 MB/s (12 Mb/s). While these workstations were running the benchmark, the West Ethernet was monitored by FS04's traffic utility. Network statistics were retrieved along with the nfsstones data. These workstations started their execution one by one, so that the network work load would correspond to the number of NFS clients present. This test was carried out during normal business hours.

3.3 Case Study Results

NFS performance problems can usually be broken down into four areas: client, network, server bottlenecks, and NFS itself. If the potential problem areas can be identified among these components, these problems can be isolated and addressed properly. The results of this study, along with its potential problems will be discussed in the following sections.

3.3.1 Client Bottlenecks

Figure 4 is a graph of traffic during production hours. This graph shows the West Ethernet utilization versus the number of NFS clients present. At the early stage of the test, no NFS client was running the benchmark yet. The less-than 10% traffic load was the nominal usage on the West Ethernet. When one workstation started the benchmark, the traffic load increased to 20%. As more clients participated in the test, the traffic load increased. When more than 4 workstations were involved in the test, the average work load sustained a level of 40% and peaked at 60% occasionally. The West Ethernet was stretched to its limit, constantly receiving at least 4 Mb/s of traffic at this time. It took only eight SGI 4D/60 workstations to saturate the Ethernet.

The 4D/60 is the slowest of the SGI 4D machines that was used in this test. It can absorb incoming I/O at a rate of 5.12 Mb/s as shown in the loop-back test. Although the Ethernet transfer specification is 10 Mb/s, 5 Mb/s is a more practical and acceptable figure, because network usage can always be translated into network delay as shown by B. Lyon and R. Sandberg[9]. With the rate that the 4D/60 can absorb incoming I/O, it does not seem that the client is a bottleneck in this environment, especially since each of the workstations has its local hard disk. Certain amounts of information can be stored locally; it largely reduces the overall load on the server and the network. With the availability of the WKSII, which has an I/O transfer rate four times as fast as the 4D/60's, the clients are not going to be the bottleneck in this environment.

---

[9] B. Lyon, R Sandberg. " Breaking Through the NFS Performance Barrier", Legato Systems, Inc. commercial publication, 1989.

Figure 3.Nominal West Ethernet utilization during normal business hours.

Figure 4.West Ethernet utilization during Nfsstones test.

3.3.2 Server Bottlenecks

Figure 5 is a plot of nfsstones performance on Navier in this time interval. Curve 1 is the average nfsstones performance versus a varying number of NFS clients present; curve 2 is the total nfsstones per second delivered by Navier. When a single workstation was running, it achieved a result of 64 nfsstones per second. As the number of NFS clients increased, the average performance decreased and the total nfsstones delivery increased. When 8 clients were present, Navier's average performance dropped to a low of 28.1 nfsstones per second. Its total nfsstones delivery leveled off at 220 nfsstones per second. Although Navier is a supercomputer, it is still a limited resource. 220 nfsstones per second was the maximum total that Navier delivered in this test. The average time per server request remained at a rate of 880 nanoseconds.

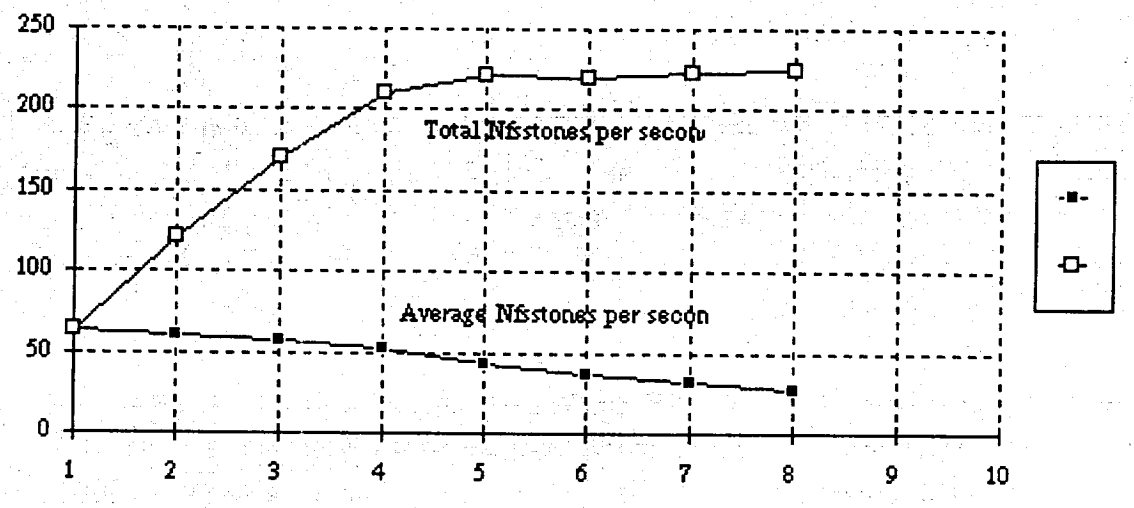

Figure 5.Cray-2 Nfsstones Performance.

A similar study has been conducted at Legato Systems, Inc., using 8 diskless Sun 3 clients[10]. It is found that 30 NFS calls per second is a fair representative of an average NFS network. At this load, average response time for an NFS operation is 47 ms. Comparing this result with the case study at NAS, the average performance achieved in this environment is much better than that of an average NFS environment.

The server CPU could be the bottleneck in an NFS environment, but it is not the case here. Rather, the server's I/O subsystem is usually the primary cause of poor NFS performance. The speed of the disk is the limiting factor on most NFS servers. CPUs in the class of the Cray Y-MP and the Cray-2 can keep up with the rate of NFS requests that the Ethernet can deliver. Slower machines are not able to do this, and even moderate loads on a server could swamp its CPU. The Cray-2's 65.5 Mb/s I/O capability has no problem in catching up with the NFS requests that an 10 Mb/s Ethernet can deliver. As it is shown in Table 3, there is only a 10% difference in performance when a Cray-2 is used as a server instead of a 4D/70. The Cray 's 250 MHZ CPU is at least 30 times faster than the 12 MHZ CPU on a 4D/70. With 4 CPUs on the Cray-2, the total performance is 120 times faster than a 4D/70, but there is only a 10% improvement in the NFS performance. This is simply because the server is not a bottleneck in an NFS environment.

The server is unlikely to cause any potential problem, unless each workstation has a dedicated channel requesting services from the Crays. Otherwise, we should be more concerned about Crays flooding the Ethernet easily.

3.3.3 Network Bottlenecks

The network used to communicate between the client and server does not normally cause a performance bottleneck. There are, however, two conditions to look out for: network delays and high retransmission rates. If the Ethernet is over-utilized, the client will experience longer delays waiting for a free slot to send requests in. Ethernet utilization over 50% is often indicative of excessive network delay, as recommended by the Network and Communication Group at NAS. Another factor contributing to excessive delay is network topology. If clients are located many hops away from servers, their requests may experience

---

10. ibidem

long delays.

In the period these workstations were running nfsstones, netstat reported the statistics in table 5. The collision rate on the West Ethernet was monitored at 0.8% in a one-week test period. The West Ethernet collision rate increased to 9.6% after running nfsstones. This drastic increase in collision rate indicates that the probability of packet collision is high. NFS is a UDP based protocol. This simple but unreliable protocol sends a packet but does not guarantee that it will be received. When 8 NFS clients were running on the West Ethernet, a tremendous amount of network traffic was created. Consequently, the probability of collision increased greatly since the network collision rate is directly proportional to the network load. If a NFS client's request is not acknowledged, NFS retransmits the request. A high retransmission rate can created even more traffic on the network and make the situation worse.

	Collision rate
No nfsstones running	0.8%
8 clients running nfsstones	9.6%

Table 5.Netstat statistic on the West Ethernet

In the worst-case scenario, assuming all 31 workstations on the West Ethernet are 4D/60s, this subnet can generate traffic of 153.6 Mb/s. It exceeds the 10 Mb/s bandwidth of Ethernet. It is not possible for Ethernet to catch up with this number of NFS requests. However, this is the worst-case scenario; it is unlikely that all workstation users will be generating this kind of network traffic all at the same time.

Network utilization at NAS peaked at 60% (6 Mb/s) in this test during normal office hours. 50% or more network usage is an indication of an overloaded network. As the number of NFS clients increases, the traffic on the Ethernet will increases. Each client competes for

network usage and blocks the others. For instance, rlogin processes that require instant-interactive response are impacted most.

Therefore, although the NFS impact on the supercomputers is minimal, its impact on the network and the clients are much more severe. The clients will experience a long delay in sending a request or waiting for a service. This potential problem has to be resolved to improve the network throughput.

3.3.4 NFS Bottlenecks

The nature of NFS itself causes performance bottlenecks at the server. This simple stateless protocol requires a client request to complete before the client can be acknowledged. If a client does not receive this acknowledgement, it retransmits that request to the server. This protocol guarantees all client requests actually complete and modified data is safely stored. It requires data to be synchronously committed to disk; therefore, a server cannot easily cache modified data in volatile storage. This very desirable property of crash-survivability causes these performance problems:

- all NFS operations require disk I/O operations,
- these operations have to be performed serially; there is no opportunity to optimize the disk arm scheduling,
- disk write operations cannot be avoided by caching.

These factors contribute to NFS itself possibly being a bottleneck.

4.0 Recommendation

The case study shows that the impact of NFS on the network is more severe than that on the supercomputers. Even with a 10% nominal network usage, the network remains a potential problem. Under excessive network traffic load, whether it is created by NFS, FTP or RCP, it could block others from accessing resources through the same channel. For instance, rlogin and rsh processes that require instant-interactive response are impacted most. The root cause to the problem is the current state of the NAS network, not NFS. To be more specific, it is caused by the Ethernet. NFS is a valuable tool; it should be made available to

NAS users. Although NFS does impact the NAS computing environment in some ways, this situation can be corrected; especially with the ongoing Medium Speed LAN project and the availability of WKSII.

4.1 Network Solution

This Medium Speed LAN project will re-structure the current NAS network configuration. Although the procurement has not been awarded yet, it is scheduled for the near future. Figure 6 is a conceptual configuration of the future NAS network. NASNET will again be divided into several subnets; each has a direct link to the HSP computers. Each Medium Speed LAN is capable of supporting a minimum of 150 SGI scientific workstations; and capable of delivering a minimum userspace-to-userspace transfer capability of 8 Mb/s between the HSP computers and the workstations. The HSP computers are then networked to the MSS 2. The minimum transfer rate on this particular link is 20 Mb/s. The hops in between the HSP and the workstation in the current configuration will be eliminated to diminish the network delays.

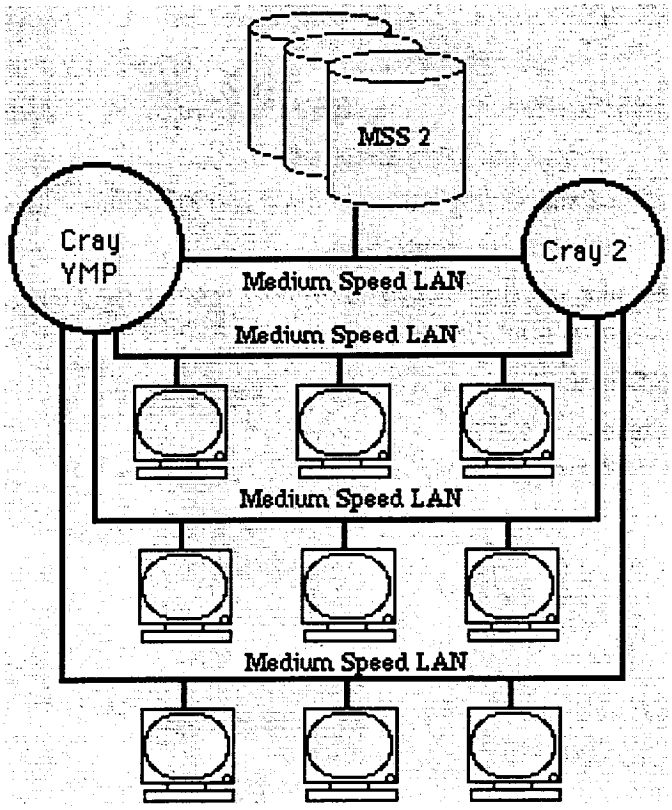

Figure 6.A Conceptual NAS' Medium Speed LAN Configuration.

The future NASNET configuration is highly dependent on the hardware available; therefore, it is difficult to predict the actual configuration at this time. The number of subnets and the number of workstations on each subnet has to be determined when the Medium Speed LAN is available. The actual effects of NFS on the Medium Speed LAN is not clear. A test has been performed on a 4D/60 workstation. A file is read from the disk and then

written back to it. The real-time transfer rate is found to be 7.2 Mb/s. According to the Medium Speed LAN specifications, users theoretically should be able to retrieve data from the remote file systems at a rate faster than reading the same data from the local disk.

If the Medium Speed LAN can not be made available, an alternative solution is further dividing the current NASNET into more subnets. This will reduce the number of workstations and the possible network traffic on each subnet. NASNET currently employes a class B network structure; it can handle up to 256 subnets. Should the subnet performance remain a concern, a subnet with a selected group of users can be created. This subnet can be monitored to determine its performance for a period of time. The results should help determine the number of subnets and the optimal number of workstations on it.

4.2 Workstation Solution

Currently, hard disks found on many workstations are too small. It leaves the workstations running in a network-dependent mode. A large portion of the system software and applications are being relocated to file servers in order to free up local disk space. Workstations frequently have to access this from a file server, increasing the amount of NFS traffic on the network. For example, a user needs to run the PLOT3D application on the workstation; this process requires reading the application from the file server and then sending it through the network. Users may be totally unaware of these transparent file transfers, but it take up network bandwidth which could be useful in other operations.

Expanding the local storage on the workstations would allow operating system and application software to be installed on the workstations. The demands on the server and the network might be reduced, allowing better system and network throughput.

4.3 User Training

Currently, only a limited amount of solution analyses can be performed on a workstation because of the limited resources of the workstations. WKSII makes solution analysis on workstations possible as a result of higher CPU performance, more internal memory and larger local storage space. More applications are expected to be performed on worksta-

tions. Data still may have to be retrieved from the Cray computers.

Users behavior also contributes to the effects of NFS to NAS. Unfortunately, this behavior can not be easily predicted or modelled. Let's consider a user who needs to run PLOT3D on a workstation. The user needs to start up the application on the workstation and read in the data set from the remote computer. There are two ways of doing it. He can nfsmount the remote file system and use the data, or he can copy the file to a local disk, assuming local storage space is available.

The first approach allows the user to retrieve the data in a transparent manner. All the user needs to do is nfsmount the remote file system; the remote data can be used as local data. He may not be aware that sending a 160 MB data file through the network will take up 30% of the network bandwidth and last at least 15 minutes. Sometimes user wants to rerun the application for some reason; say there is a mistake, and the user needs to restart the application. The data set has to be sent through the network again; this will once again create more network traffic. This process will go on until the user has completed with the job.

The second approach is reading the remote data set onto his workstation disk. The user only needs to transfer the data through the network once. After that, this data on the workstation can be repeatedly used until it is not needed any more. The critical assumption is that the user has a disk with enough storage space for the 160 MB data set.

It can be seen that users' behavior plays an important role in the NFS performance and the the network resources; therefore, it is important to educate the users to use these resources properly. Some of the guidelines users should follow are:

- Users should maximize the workstation disk usage by storing the currently in-use data on it. This can be done by cleaning up the local disk as a routine practice. Remote files should be copied to workstations only when necessary. These files should stay on the workstation and be used as long as possible.
- When a new application is developed to run in a distributed mode between the workstations and the Cray computers, the program should read data in from the software running on the Cray computer. This will avoid sending data through the network.

- If an application has to be developed to run on the workstation but requires data to be read in from the remote computer, users should attempt to use workstation resources first. For example, create a small data set on the workstation hard disk and use this data for development testing until the application is ready to be tested with data retrieved from the remote computers.
- Using the Cray supercomputers as big file servers or as a backup device should be avoided.

These are some but not all of the principles that users should follow. When new ideas come up, they should be shared with all users. Of course, it is understood that retrieving data from the remote computers cannot be avoided in certain situations. Should this be the case, users are encouraged to exercise their judgement to avoid any abusive usage of the network resources. If these suggestions can be implemented, NFS activities on the NAS computing environment can be minimized, and NFS impact will become minimal, and the workload on the HSP computers would be off-loaded.

5.0 Conclusion

NFS is a valuable tool; it should be made available to NAS users. It allows remote file systems to be shared by different users. Although the case study shows that the impact of NFS on the current NAS network is potentially high, we should direct our efforts to correct the roots of the problem: the workstation local storage problem and the network problem. We can also re-educate the NAS users by providing NFS guidelines so that the impacts of NFS on the NAS computing environment is minimal. Users should be encouraged to use the computer and network resources intelligently.

Appendix A. NFS Test Result on Navier

BASIC TESTS (directory /r/navier/nb/lam/test/testdir)

The test directory is /r/navier/nb/lam/test/testdir
test1: File and directory creation test
created 155 files 62 directories 5 levels deep in 19.22 seconds
test1 ok.

test2: File and directory removal test

real 20.3
user 0.2
sys 10.9
test2 ok.

The test directory is /r/navier/nb/lam/test/testdir
test3: lookups across mount point
500 getwd and stat calls in 24.67 seconds
test3 ok.

The test directory is /r/navier/nb/lam/test/testdir
test4: setattr, getattr, and lookup
1000 chmods and stats on 10 files in 22.43 seconds
test4 ok.

The test directory is /r/navier/nb/lam/test/testdir
test5: read and write
wrote 1048576 byte file 10 times in 59.16 seconds (177724 bytes/sec)
read 1048576 byte file 10 times in 66.80 seconds (158875 bytes/sec)
test5 ok.

The test directory is /r/navier/nb/lam/test/testdir
test6: readdir
20500 entries read, 200 files in 41.79 seconds
test6 ok.

The test directory is /r/navier/nb/lam/test/testdir
test7: link and rename
200 renames and links on 10 files in 23.9 seconds
test7 ok.

The test directory is /r/navier/nb/lam/test/testdir
test8: symlink and readlink
test8: (/r/navier/nb/lam/test/testdir) can't make symlink file.0 : I/O error

The test directory is /r/navier/nb/lam/test/testdir

test9: statfs
type=1, bsize=4096, blocks=5940480, bfree=151568,
bavail=5788912, files=0, ffree=0, fname=navier, fpack=
1500 statfs calls in 19.85 seconds
test9 ok.

Congratulations, you passed the basic tests!

GENERAL TESTS (directory /r/navier/nb/lam/test/testdir)
if (-d /r/navier/nb/lam/test/testdir) then
rm -rf /r/navier/nb/lam/test/testdir
mkdir /r/navier/nb/lam/test/testdir
endif
cp Makefile runtests *.sh *.c mkdummy rmdummy nroff.in makefile.tst \
/r/navier/nb/lam/test/testdir

Small Compile
8.3 (3.2) real 2.1 (0.1) user 2.0 (0.4) sys

Tbl
stat: bad data format in tbl.time (Permission denied: tbl)

Nroff
stat: bad data format in nroff.time (No such file or directory: nroff)

Large Compile
17.8 (3.1) real 4.5 (0.0) user 3.0 (0.4) sys

Four simultaneous large compiles
37.7 (2.2) real 17.7 (0.3) user 10.9 (0.3) sys

Makefile

11.2 (0.6) real 1.1 (0.0) user 5.8 (0.2) sys

SPECIAL TESTS (directory /r/navier/nb/lam/test/testdir)
if (-d /r/navier/nb/lam/test/testdir) then
rm -rf /r/navier/nb/lam/test/testdir
mkdir /r/navier/nb/lam/test/testdir
endif
cp runtests open-unlk open-chmod dupreq excltest statfs negseek rename holey
/r/navier/nb/lam/test/testdir

check for proper open/unlink operation
nfstesta29367 open; unlink ret = 0
Test completed successfully.

check for proper open/chmod 0 operation

nfstesta29368 open; chmod ret = 0
test completed successfully.

check for lost reply on non-idempotent requests
100 tries, 0 lost replies

test exclusive create, should get: exctest.file2: File exists
exctest.file2: File exists

test statfs for file count, should get positive, different numbers
(known bug in some implementations)
inodes 4606 free 3395

test negative seek, you should get: read: Invalid argument
read: Invalid argument

test rename
Test completed successfully.

test sparse file write/read

Holey file test ok

Special tests complete

All tests completed

Appendix B. NFS Test Result on Reynolds

BASIC TESTS (directory /r/reynolds/rb/lam/test/testdir)

The test directory is /r/reynolds/rb/lam/test/testdir
test1: File and directory creation test
created 155 files 62 directories 5 levels deep in 16.70 seconds
test1 ok.

test2: File and directory removal test

real 36.0
user 0.3
sys 11.7
test2 ok.

The test directory is /r/reynolds/rb/lam/test/testdir
test3: lookups across mount point
500 getwd and stat calls in 56.76 seconds
test3 ok.

The test directory is /r/reynolds/rb/lam/test/testdir
test4: setattr, getattr, and lookup
1000 chmods and stats on 10 files in 19.59 seconds
test4 ok.

The test directory is /r/reynolds/rb/lam/test/testdir
test5: read and write
wrote 1048576 byte file 10 times in 64.36 seconds (163840 bytes/sec)
read 1048576 byte file 10 times in 71.18 seconds (147686 bytes/sec)
test5 ok.

The test directory is /r/reynolds/rb/lam/test/testdir
test6: readdir
20500 entries read, 200 files in 36.73 seconds
test6 ok.

The test directory is /r/reynolds/rb/lam/test/testdir
test7: link and rename
200 renames and links on 10 files in 19.66 seconds
test7 ok.

The test directory is /r/reynolds/rb/lam/test/testdir
test8: symlink and readlink
test8: (/r/reynolds/rb/lam/test/testdir) can't make symlink file.0 : I/O error

The test directory is /r/reynolds/rb/lam/test/testdir

```
test9: statfs
        type=1, bsize=4096, blocks=7150080, bfree=1142704,
        bavail=6007376, files=0, ffree=0, fname=reynol, fpack=ds
        1500 statfs calls in 17.17 seconds
        test9 ok.

Congratulations, you passed the basic tests!

GENERAL TESTS (directory /r/reynolds/rb/lam/test/testdir)
        if (-d /r/reynolds/rb/lam/test/testdir) then
        rm -rf /r/reynolds/rb/lam/test/testdir
        mkdir /r/reynolds/rb/lam/test/testdir
        endif
        cp Makefile runtests *.sh *.c mkdummy rmdummy nroff.in makefile.tst \
                /r/reynolds/rb/lam/test/testdir

Small Compile
        7.1 (2.5) real          1.9 (0.0) user          1.8 (0.4) sys

Tbl
stat: bad data format in tbl.time (Permission denied: tbl)

Nroff
stat: bad data format in nroff.time (No such file or directory: nroff)

Large Compile
        12.1 (1.0) real  4.3 (0.0) user         2.3 (0.1) sys

Four simultaneous large compiles
        39.6 (1.8) real  17.9 (0.3) user        10.9 (0.1) sys

Makefile
        10.6 (0.6) real  1.1 (0.1) user         5.5 (0.2) sys


SPECIAL TESTS (directory /r/reynolds/rb/lam/test/testdir)
        if ( -d /r/reynolds/rb/lam/test/testdir) then
        rm -rf /r/reynolds/rb/lam/test/testdir
        mkdir /r/reynolds/rb/lam/test/testdir
        endif
        cp runtests open-unlk open-chmod dupreq excltest statfs negseek rename holey
/r/reynolds/rb/lam/test/testdir

check for proper open/unlink operation
nfstesta28720 open; unlink ret = 0
Test completed successfully.

check for proper open/chmod 0 operation
nfstesta28721 open; chmod ret = 0
```

test completed successfully.

check for lost reply on non-idempotent requests
100 tries, 0 lost replies

test exclusive create, should get: exctest.file2: File exists
exctest.file2: File exists

test statfs for file count, should get positive, different numbers
(known bug in some implementations)
inodes 4606 free 3395

test negative seek, you should get: read: Invalid argument
read: Invalid argument

test rename
Test completed successfully.

test sparse file write/read

Holey file test ok

Special tests complete

All tests completed