Configuring A Client/Server File Service Network

Chuck Kollars, Sun Microsystems
<ckollars@East.Sun.COM>
February 1991, This Revision April 1992

INTRODUCTION

Purpose of This Document

You know generally how many desktop clients there will be and how much and what kind of work they'll be doing. But how many servers do you need with how many disks? And where should the wires go? This document will help you answer those questions.

Audience

This document is directed at experienced administrators and planners. Although it was written in the context of workstations and servers made by Sun, it applies to any systems that use NFS.

Anyone who uses the techniques described in this document should understand they are provisional. All configurations generated by these techniques should be sanity–checked against past experience. If the answer given by the technique appears to be wildly incorrect, it very well may be just that.

Limitations

This document is concerned with file service using NFS. It doesn't address the configuration of compute servers, database servers, network infrastructure servers (nameservice, routing, email, etc.), or multi-user servers.

This document focuses on the design of a brand new network. The techniques can be applied to existing networks as well, but this document doesn't talk about how to do so.

The techniques in this document use average load values over a 200 second period of heavy activity. This is reasonable in most situations. However if activity occurs in short sharp bursts, or activity of many workstations is synchronized, using a 200 second interval will underestimate the peak load and under–configure the servers and networks.

Contents

BACKGROUND INFORMATION

Measuring and Using *NFSops*

Client computers work together very closely with server computers. Often the client computers are desktop workstations that communicate with the user through a WIMP (windows icons mouse pointer) interface. Server computers may provide file service, database service, or compute service as well as administrative services such as email and naming. Frequently server computers provide shared file service to the client computers via NFS (Network File System).

The relatively new metric NFSops provides a measure of how much NFS work a server can do. A benchmark tool called either *nhfsstone* (pronounced n-f-s-stone, the 'h' is silent) or *LADDIS* (an anronym representing the companies that participated in its development) running on a client computer sends NFS requests over a network to a server computer under test, and measures how many requests get answered how quickly. In its most complete form, the results of LADDIS/nhfsstone are presented as a line graph with number of NFSops on the horizontal axis and response time on the vertical axis.

Not all NFS requests are equal. Some involve large amounts of data and other don't; some reference frequently used information that's probably in a cache and others don't. So an NFSops rating isn't meaningful unless you also know the *mix* of NFS operations. Most publicly available test results were measured using one or more of the four mixes presented in Table 1. (Although publicly available test results should include four sets of results corresponding to these four mixes, they may not. You may need to interpolate from the limited set of results that are available. Results change substantially depending on the operations mix, especially the proportion of read+write requests. Interpolating from incomplete specifications is often difficult. You may have to request more information from the vendor.)

- ♦ The NetBench mix is representative of the mix of NFS operations requested by a diskfull client running a mix of Unix shell commands including make, cp, ls, rm, and find. It's the mix generated by the NetBench test that generates a NetStone rating. This document refers to NFSops using the NetBench mix as NFSops_{NB}.
- ♦ The Legato mix, also sometimes called the diskless mix, is representative of the mix of NFS operations requested by a diskless client running a then–current OS and real–world applications a couple years ago. It's the default mix used by LADDIS/nhfsstone. This document refers to NFSops using the Legato mix as NFSops_{LG}.
- ♦ The Brown mix, also sometimes called the dataless mix, is representative of the mix of NFS operations requested by a dataless client running real–world applications. A dataless client is one that has a local disk, but uses it only for swap, or for root and swap, or for swap and /usr. This document refers to NFSops using the Brown mix as NFSops_{BW}.
- ♦ The SunMark mix is representative of the mix of NFS operations requested by a dataless client running shell commands typically used in a third–generation software development environment. It's the mix generated by the SunMark test that was used to generate some client/server ratio estimates. This document refers to NFSops using the SunMark mix as NFSops_{SM}.

A second way not all NFSops are equal is data buffer size. One NFS read operation with rsize=8KB moves a lot more data than one with rsize=2KB. Most NFS client implementations default to an 8KB buffer size since it's the largest size supported by most server implementations. If you're using a larger buffer size to get better performance out of a specialized server, or a smaller buffer size to compensate for underpowered Ethernet bridges or slow Ethernet interfaces, there will be some additional error in your result. Proceed as though your buffer size were the standard 8KB anyway. Published performance numbers using nonstandard buffer sizes are rare and adjusting numbers to compensate for a different buffer size is difficult.

A third way not all NFSops are equal is response time. LADDIS/nhfsstone can be used to produce a line showing the relationship between average response time in milliseconds and throughput in NFSops. Yet vendors who are touting the capacity of their servers often state their NFS performance as one number. Some old statements used a 70 millisecond response time cutoff, others used a 50 millisecond cutoff. Most recent statements quote a peak value rather than using a response time cutoff. This is usually reasonable since the entire results line for most new fast servers lies below any meaningful response time cutoff.

Some users have much more stringent average response time requirements than others. For example getting good interactive response from an image processing application running on a diskless workstation may require that the average NFS response time of the server be only 25 milliseconds. If you have stronger (or weaker) response time requirements, you can read off a different value of NFSops. This relationship between response time and throughput applies only to NFS servers, not to clients or networks. Clients will generate the same load unless the server response time is so slow that client performance is degraded. Communicating an NFS request over an Ethernet will take only a few milliseconds (assuming the Ethernet isn’t overloaded or faulty), which is negligible compared to the response time of the server itself.

The mix of requests generated by your workload is probably similar to one of these four mixes. To use your exact mix would take a lot of work and the result probably won’t be more accurate. So pick the standard mix that’s closest to your own mix and use it.

TABLE 1 – THE 4 “STANDARD” NFS OPERATION MIXES

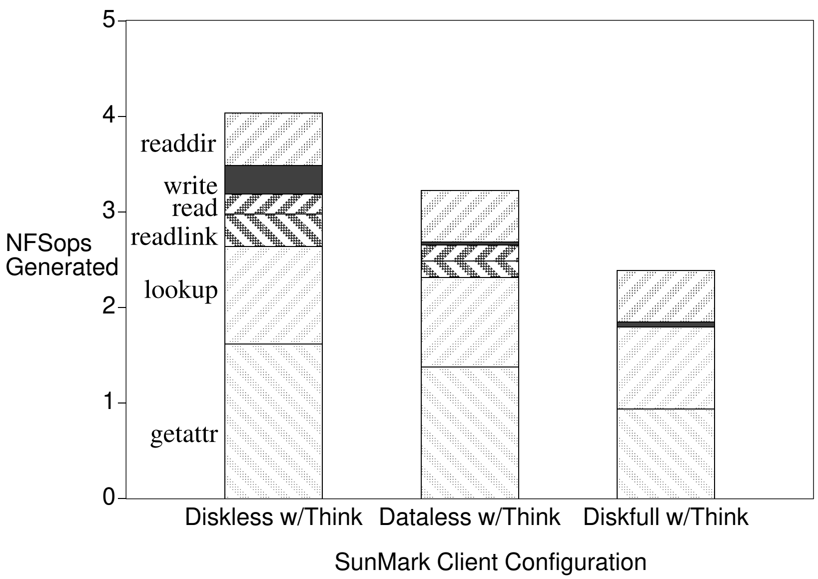
NFS Operation	NETBENCH Mix %	LEGATO Mix %	BROWN Mix %	SUNMARK Mix %
null	0	0	0	0
getattr	26	13	23	41
setattr	0	1	1	0
root	0	0	0	0
lookup	7	34	45	29
readlink	7	8	8	5
read	21	22	13	5
wrcache	0	0	0	0
write	24	15	2	1
create	2	2	1	1
remove	1	1	1	1
rename	0	0	0	0
link	0	0	0	0
symlink	0	0	0	0
mkdir	0	0	0	0
rmdir	0	0	0	0
readdir	9	3	5	16
fsstat	3	1	1	1

Older and Newer Mixes

To give you some feel for how NFS operation mixes have changed over time and hence how to interpret old test results, Figure 1 presents the NFS load generated by one client machine running the old MCS SunMark test.

Figure 1

NFS Load Generated By SunMark



The NFS Equation

Our goal is to determine for a specific situation how many NFS clients an NFS server can support. Assuming the NFS data transfer size is fixed, a reasonable approximation is this four–parameter conceptual equation:

`clients-per-server =` *f*`(mix,response-time,load-per-client)`

Values for three of the four parameters can be determined easily. The `mix` can be either measured or estimated, and a similar "standard" mix used in the rest of the calculations. Average `response-time` can be either a user's requirement or a fixed value such as 50 milliseconds. Generated `load-per-client` can be determined or estimated several different ways.

Once three values are available, the fourth can be calculated. Being able to calculate a realistic specific `clients-per-server` means being able to realistically configure a specific network.

Load Generated per Client

The clients–per–server values calculated for different situations will vary by one or two orders of magnitude! That's why most client/server ratio recommendations published by vendors have been either "way off" or so wide they were next to useless.

<u>This document's premise is that if you can estimate `mix` and `load-per-client` for a specific network, you can calculate a reasonable and useful `clients-per-server` value for that network.</u>

The House Wiring Analogy

Laying out a network of client and server computers that use NFS to share files is not significantly different from laying out the electrical wiring in a house.

There are good rules of thumb for wiring a house: Use 15 amp fuses with number 14 wire, 20 amp fuses with number 12 wire; the sum of individual fuses shouldn't exceed twice the size of the main fuse; the toaster, coffee maker, microwave, clothes iron, and hair dryer shouldn't all be on the same circuit; and so on.

Such a simple approach hasn't yet been much used for laying out client/server networks. One reason is there has been no common metric. We measure electricity in *amps*, but what do we measure NFS service in? The other reason is there has been little information on just how much load a client generates. Any number of books will tell us that a clothes iron draws about 7 amps; but how much load does a dataless desktop machine running office automation applications generate?

In this document we aim to fill both gaps. Our metric for NFS service is *NFS Operations Per Second* (abbreviated *NFSops*). NFSops are only slightly more complicated than amps in that a mix must be specified. NFSops_{NB}, NFSops_{LG}, NFSops_{BW}, and NFSops_{SM} are like amps_{green}, amps_{red}, amps_{blue}, and amps_{yellow}. And our approach to determining generated client load is to collect information from many sites and publish it. The collected list of actual generated loads is included in this document.

CONFIG STEP 1 – DETERMINE THE CLIENTS (DESKTOPS)

For Each Workload

If the installation includes different types of desktops, each with their own workload and usage pattern, group each type of desktop together and figure out how much load those desktops will generate. If the groups of desktops will all share common servers, add the results for each group of desktops together to get a grand total. If each group will have its own servers, keep the numbers separate for now even if they share an Ethernet.

How Many Clients

Determining the number of client systems is usually simple. While the number of servers required and the type and location of network wiring are still open questions, the number of seats to be served is well known. If different clients will have significantly different workloads or usage patterns, group similar seats together and handle each group separately.

How Much Load Generated by Each Client

Knowing the NFSops mix and the overall NFSops load generated by one client, you can calculate the total load all the clients together to generate.

To avoid re–measuring the server for every NFS operations mix, use one of the four "standard" mixes. Find out the actual mix generated by your group of client systems. Then choose the standard mix that's most similar to your specific mix: $NFSops_{NB}$, $NFSops_{LG}$, $NFSops_{BW}$, or $NFSops_{SM}$. It's most important to match the percentage of read+write(+wrcache) operations, as they include large amounts of data and put a much heavier load on disks and networks than the other operations.

The hard part is figuring out how much load one client generates. Real–world clients vary so greatly that "rules of thumb" don't work. Here are three different ways to determine how much load your clients will generate. Choose one of them. The first one is most accurate and most difficult; the last one is least accurate and easiest.

Alternative 1: Set up one test client and one test server in a lab

Alternative 2: Measure a similar client at the same or a similar site

Alternative 3: Look up a similar client in a list of known actual loads

Load Generated by Each Client – Alternative 1 of 3
Set Up One Test Client And One Test Server In A Lab

The first way to find out the actual load and mix generated by your group of clients is to set up a sample client configured the proposed way and with the actual proposed workload running on it. Use scripts, journalling, or a human to generate actual workload activity. Then use the utility program nfsstat to determine the level and mix of NFSops being generated.

```
shell> # *** load generated by an existing client since it was last booted ***
shell> nfsstat -nc #display level and mix generated since last system bootup
shell> uptime #display how long the system has been up
shell> #Calculate average level generated since bootup as "client NFS calls" divided by
shell> # total number of seconds up (not counting inactive "nightime" hours).
```

```
shell> # *** load generated by an existing client during a 200 second test period of activity ***
shell> foreach x (0 1 2 3 4 5 6)
? nfsstat -nc #display previous interval
? sleep 200 #while normal processing occurs
? end
shell> #Calculate client NFS calls in each interval by subtracting previous interval's value.
shell> #Calculate average level generated in each interval by dividing "client NFS calls" by
shell> # the length of the interval in seconds.
shell> #Figure an overall average by throwing out the lowest and highest of the
shell> # six values, then averaging together the remaining four.
```

The output will give an absolute number and a percentage for each NFS operation (null ... read, wrcache ... symlink, mkdir ... fsstat). Record the percentages—they are your *mix*. Compare your mix to the four standard mixes to see which one is most similar. It's most important to match the percentage of read+write(+wrcache) operations. These operations include large amounts of data and put a heavier load on disks and networks than the other operations.

In the output from each 200 second interval, take the very first number (the one labelled "calls"), and divide it by 200 to get the number of NFS operations requested per second in each period. Throw away the highest and lowest values, and average the remaining four together to get an overall NFSops rating for your workload. (Check the second value in the output of each 200 second interval [the one labelled "badcalls"] to be sure it's very close to zero. If it isn't, there's something wrong with your test setup and your results may not be meaningful.)

Load Generated by Each Client – Alternative 2 of 3
Measure A Similar Client

The measurement and calculation techniques are exactly the same as in Method 1. The difference is you'll be measuring a real client on site rather than a test client in a lab. If possible measure an existing client in the "next office." If you can't do that, try to measure one in the "next department" or the "next building."

The final configuration will depend critically on how similar the workstation you measure is to the new ones you're configuring. "Sorta like" isn't good enough; a poor estimate here may result in a final configuration that doesn't work.

Load Generated by Each Client – Alternative 3 of 3
Look Up A Similar Client

Find the line in the table of known actual workloads in Appendix B of this document that's most similar to your proposed client.

Consider hardware configuration, workload style, and administrative policies. Hardware configuration means what kind of system and how much RAM it has. Workload style means which principal applications will be used. Administrative policies means whether home directories are stored locally, how shared file systems are mounted, and so on.

The table of known actual generated loads in this document will be updated periodically. To get the latest table of known actual workloads, fetch a fresh copy of this document.

Peak Usage

Now that you've determined both the *mix* and the *overall* NFSops load generated by one active client, determine what the peak usage for this group of clients will be. Peak usage is measured as *max percentage of seats active simultaneously* and defined as *walk around at the busiest time and count what percentage of users are actually keying or mousing*.

In some situations all of the workstations will be in use sometimes. An example is the seats in a public room on a university campus. Peak usage is 100%. (You can use 100% in the calculations in two other situations. One is where the peak usage is so close to 100% it's not worth worrying about the exact value. The other is where the alternate periods of activity and inactivity are shorter than the sampling interval and so are already included in the load.)

In other situations the peak usage is significantly less than 100%. One situation is where the workload includes substantial amounts of "think time" when the user is waiting and little or no NFS activity is being generated. Another situation is where every office has its own workstation. If the workstations merely decorate the desktops, peak usage may be only 20%. On the other hand, if most people read their email at the same time, say 10:15 am, peak usage may be 80% even though workstation duty cycle is much lower.

Total Generated Load

Depending on which of the four standard mixes was most similar to your mix, calculate the total $NFSops_{XX}$ that this entire group of clients will generate at the busiest time:

$$NFSops_{XX}\langle group\rangle = NFSops_{XX}\langle client\rangle * \#Clients * PeakUsage$$

For example, if one client running the proposed workload generates 8.3 NFS operations per second with a mix most similar to the "Brown" mix, there are 20 similar seats, and on occasion 90% of the clients will be active at the same:

$$8.3 * 20 * 0.90 = 149.4\ NFSops_{BW}\langle group\rangle$$

CONFIG STEP 2 – DETERMINE THE SERVERS

What Size Servers

Often the size of server is dictated by space considerations and organizational considerations. For example a rack–mounted server won't fit under a desk while a deskside server costs too much for a department of only three people. Once it's decided what size server to use, you can figure out how many of them are needed.

A desktop server (a workstation with additional peripherals) can serve a small group of clients. Small or large deskside servers handle larger groups of clients. And data center servers handle the largest groups of clients. If you need very large amounts of online storage, or if you intend to put the servers in a special room (e.g. locked, raised floor, air conditioned), choose a data center server.

If each small group maintains its own server, they will probably want desktop or deskside servers. If a central group (MIS, IR, etc.) maintains servers for all groups, they may want to install all data center servers in one room.

How Many Servers

Look up the NFSops$_{NB}$, NFSops$_{LG,}$ NFSops$_{BW}$, and NFSops$_{SM}$ ratings of the desired server. To calculate how many servers you need, do a division:

```
Servers = roundup(NFSops_XX<group> / NFSops_XX<per server>)
```

To continue the example from the previous chapter, if you've chosen a deskside server rated at 100 NFSops$_{LG}$, you will need two servers.

```
2 Servers = roundup(149.4 / 100)
```

Most published NFSops$_{XX}$ ratings of servers assume the machine is dedicated to providing file service. If you expect the server machine to do other things too, such as provide name service, route email, interconnect networks, or answer database queries, de–rate the NFSops$_{XX}$ per server.

The above calculations don't include the need for redundancy or high availability. These other considerations may militate for a different model of server, or for more servers, or both.

How Many Disks on Each Server

The number of disk spindles you need is dictated by how much disk space you need and by how much performance you need. Suppose you need 350KB for the server OS, 20MB for mail and print spooling, 750KB each for sixty users, and 42MB each for twenty-five diskless clients, and each disk spindle holds 669MB:

```
Spindles = roundup(total needed / space per spindle)

0.350 + 20 + 60*0.750 + 25*42 = 1115MB needed

2 Spindles = roundup(1115 / 669)
```

NFSops$_{XX}$ performance depends on how many disk controllers or I/O channels are in the system. Of course system configuration rules may limit your flexibility. If a controller is built into each disk, you may need to add more spindles to get more performance even though you won't be using all the space on each spindle. If systems are limited to only two or three spindles per controller or channel, you may need to add another controller even though you already have the required performance.

Try different configurations until you find one that satisfies both your space requirements and your performance requirements. Try configurations with every available disk size, as the disk size that made the most sense when you were looking just at space and price may no longer make the most sense when you look at performance too.

When you install a file server, be sure the NFS load is fairly split among the disk controllers and spindles. For example the root and swap space for a large number of diskless workstations should be distributed to more than one controller or spindle, and the /usr space and the space for the server itself should be on a different controller or spindle than the root and swap spaces of the diskless workstations.

How Many Ethernet Interfaces

The NFSops rating of a particular server should include details of that server's configuration such as how many disk controllers, whether or not there's an NFS accelerator board, and the type and number of Ethernet interfaces conveying NFS load.[1] If the server you've chosen was configured with three Ethernet interfaces, then you need at least three Ethernet interfaces. You may want to include one more interface to provide connectivity to the broader internetwork.

Each Ethernet interface must be attached to a different Ethernet cable. So if you have to have more than one Ethernet interface on your server machines, you will also have to have more than one Ethernet, and some sort of interconnection between them. If you have two groups of clients that will both be handled by the same large server machine, you can run a separate Ethernet to each group of clients, and let the server machine itself route packets between the two Ethernets as necessary. Note that this routing function uses up some of the server's CPU power and also some of its Ethernet interface bandwidth, so adjust the NFSops$_{XX}$ rating of the server downward accordingly.

Until someone gets a better fix on how much "accordingly" is, adjust the server's NFSops$_{XX}$ capability downward by 10% if it provides name service, by 20% if it provides packet routing service, and by 10% if it provides email service. Adjust the capacity of the Ethernet media downward similarly.

Many modern Ethernet interfaces, particularly built–in ones, have about the same bandwidth as the Ethernet cable itself. If you need two Ethernets for performance reasons you also need two interfaces for performance reasons, and vice versa. Similarly if one Ethernet provides sufficient performance, one interface will too.

Unfortunately the performance of older Ethernet interrfaces, particularly add–on ones, varies considerably. They either place a high load on the server machine's CPU, or have inherent throughput and latency limitations, or both. If you have to use one of these older add–on interfaces, put most of the NFS load on the built–in interface and use the add–on interface to provide connectivity to the broader internetwork.

You shouldn't have to worry about these older interfaces when configuring a new network. Vendors don't publish ratings of configurations that include them. You might have to deal with these older interfaces when you're upgrading or expanding an existing installation.

There are two aspects to the performance of an Ethernet interface: throughput and packet rate. Ideally they're reciprocals of each other, related by the serial bit rate of the media itself. In practice many interfaces have either a throughput limit or a packet rate limit or both. For example an interface may have a packet rate limit of 100 packets/second, which means you won't get a lot of data through it unless you use large packets. Or an interface may have a throughput

*1. It's possible to think of each individual Ethernet interface as having its own NFSops$_{XX}$ throughput rating. And it's possible to calculate how many Ethernet interfaces you need independent of how many disk controllers or how many MIPS you need. But just because it's possible doesn't mean it's worth the effort. You should have access to published NFSops ratings for all reasonable server configurations, including configurations with more than one Ethernet interface.

limit of 50KBytes/sec, which means you won't get a lot of transctions through it unless you use small packets. The throughput limit of Ethernet with only two active stations is 1.25MBytes/sec, and the packet rate limit of Ethernet is 14,880 packets/sec. The practical throughput of an Ethernet with many nodes on it is 35% of its serial bit rate, or 0.44MBytes/sec.

As an example of network interface restrictions, Sun sanctions serving diskless workstations only off a) the server's built–in Ethernet interface, b) any SBus Ethernet interface, and c) a "3E" VME Ethernet interface in a SPARCsystem 4nn.

CONFIG STEP 3 – DETERMINE THE NETWORK (WIRING)

How Many Ethernets

Compare your expected total $NFSops_{XX}$ load to the load limit of an Ethernet. If you have more traffic than one Ethernet can carry, configure multiple Ethernets.

TABLE 2 –APPROXIMATE ETHERNET MEDIA CAPACITY

	$NFSops_{NB}$	$NFSops_{LG}$	$NFSops_{BW}$	$NFSops_{SM}$
Ethernet Capacity	105	125	275	550

The Ethernet $NFSops_{XX}$ capacities in Table 2 assume an 8KB NFS data transfer size, assume the total load is being generated by many different machines, assume there is no other traffic, and assume a maximum Ethernet utilization of 35% of its serial bit rate. One LADDIS/nhfsstone test client on an isolated network can generate much higher Ethernet usage, for example 280+ $NFSops_{LG}$. Do not expect and do not attempt such high load levels in a production environment.

Bandwidth is one Ethernet limitation. A second limitation is that not many more than 100 (some people say only 50) TCP/IP systems should be connected to one Ethernet. When there are too many systems per Ethernet, difficulties such as broadcast storms, slow simultaneous booting, and poor nameservice load balancing become significant.

Interconnecting Ethernets

When interconnecting Ethernets, you may be presented with a choice between “bridges” and “routers.” Bridges handle all network protocols. Classic routers handle only one network protocol, for example TCP/IP.

Bridges can be added or removed any time any place without administrative side effects such as having to change the addresses of some hosts. They are sometimes used to isolate the traffic between a set of diskless clients and their file server from the main network.

On the other hand bridges can’t make intelligent decisions about which packets to forward. They blindly forward all broadcast packets to all networks. The resulting high level of broadcast traffic can degrade the performance of the attached hosts. Unfiltered broadcasting can overrun administrative boundaries, so a client may discover and try to communicate directly with a server that’s hundreds of miles away.

Another problem with bridges is that many inexpensive models interfere with NFS operation. If you decide to use bridges, get a top of the line model that can both filter and forward packets at or near media speed. If you try to save money by buying a less expensive (slower) model bridge, you’ll find you have to reduce the NFS buffer size to get NFS to work through the bridge.

If all the nodes on the network use the same protocol stack (ex: TCP/IP), you can use simple routers. In fact you can probably dedicate some of your systems to act as routers. You can even have some systems act as routers in addition to their primary purpose, probably file service. If on the other hand the network must carry packets from more than one protocol stack, simple routers won’t work. You will need to use either bridges or multi–protocol routers or a hybrid of the two.

A solution to the “bridges or routers” quandary is to use multi–protocol routers that fall back to bridging packets they don’t know how to route.

A second possible solution to the “bridges or routers” quandary is to use bridges at the lowest level and routers at all higher levels. The lowest level should never have more than 5 segments connected with bridges. The problem with this solution is that network administrators have to learn how to monitor and control both bridges and routers.

The term *brouter* refers to at least three different kinds of box that are neither a pure bridge nor a pure router. When you hear it, ask more specific questions. *brouter* may refer to a smarter bridge that supports multiple, redundant, and parallel links. *brouter* may refer to a multi–protocol router that falls back to bridging packets it doesn’t know how to route. Finally *brouter* may refer to a box where each port can be configured to act as either a bridge or a router.

Problems have been seen with even the most recent “brouter” products when used in very large (>30 segments) networks. It’s not clear whether the problems were due to misuse of the brouters or to an inherent inability of the brouters to scale up to very large networks. If a significant amount of network traffic escapes the “router” function and falls into the bridge function, the same scaling problems that plague pure bridges will affect even the best multi–protocol router/bridge.

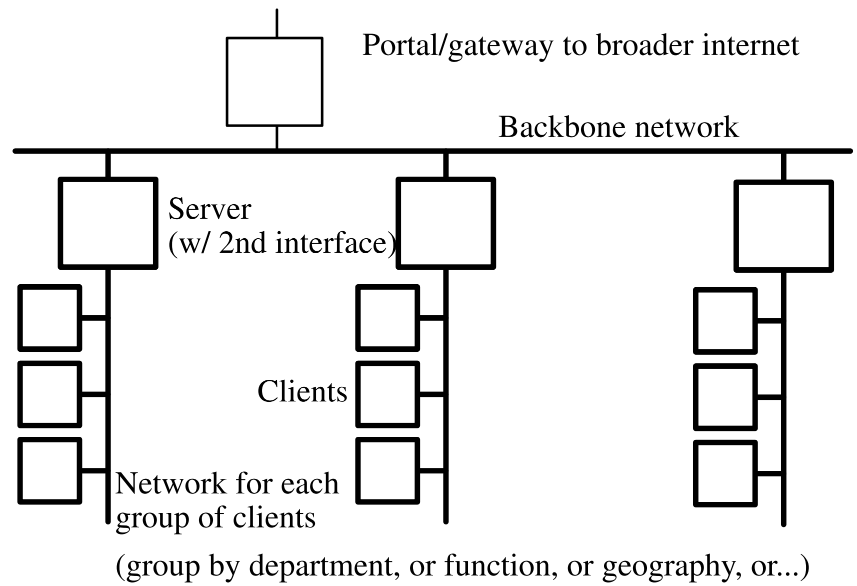
If more than 3% of the packets on any network segment are addressed to the Ethernet broadcast address (all ones— ff:ff:ff:ff:ff:ff), you have a potential problem. If the majority of these broadcast packets are coming onto the segment via a bridge or router or gateway, the network topology is causing (or at least not preventing) the problem.

Topologies for Interconnecting Ethernets

Here are four topology models for interconnecting Ethernets. Each one addresses a particular situation or requirement. They can be mixed or combined.

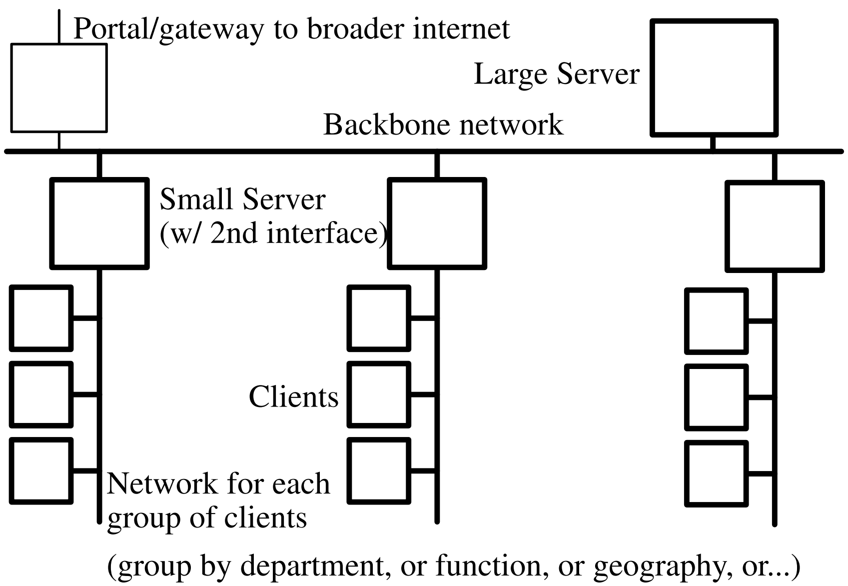
For a large installation with many groups of clients and many networks, use the *basic hierarchy*. Put each group of clients and its associated server on one network. Put either a second Ethernet interface or an FDDI interface into each server machine, and connect those interfaces to a “backbone” network. Most traffic will be between a client and its server and will stay on one network. Yet there is full connectivity so any machine can reach any other machine.

Figure 2
Topology 1 of 4 – Basic Hierarchy



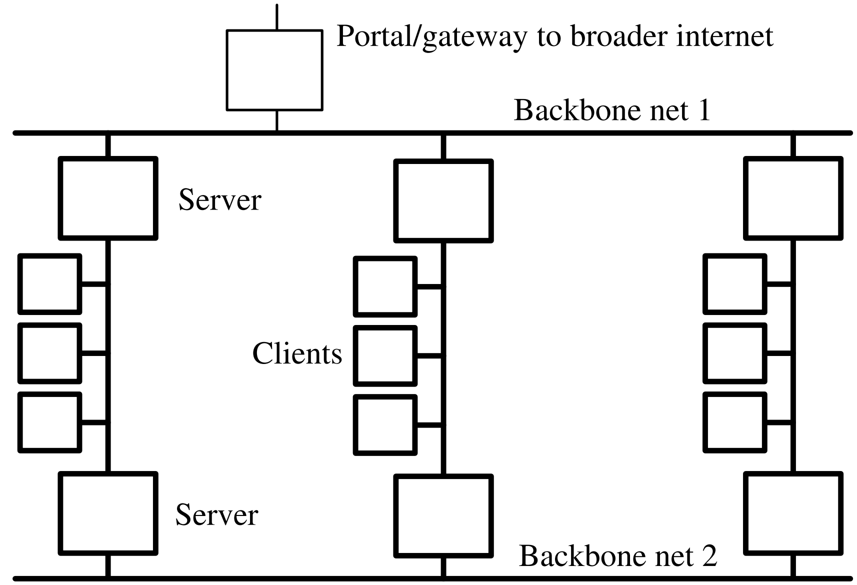
If you’re using diskless clients, or if there is a clear separation between “departmental” data needs and “global” data needs, use a *two–tiered* topology.

Figure 3
Topology 2 of 4 – Two–Tiered



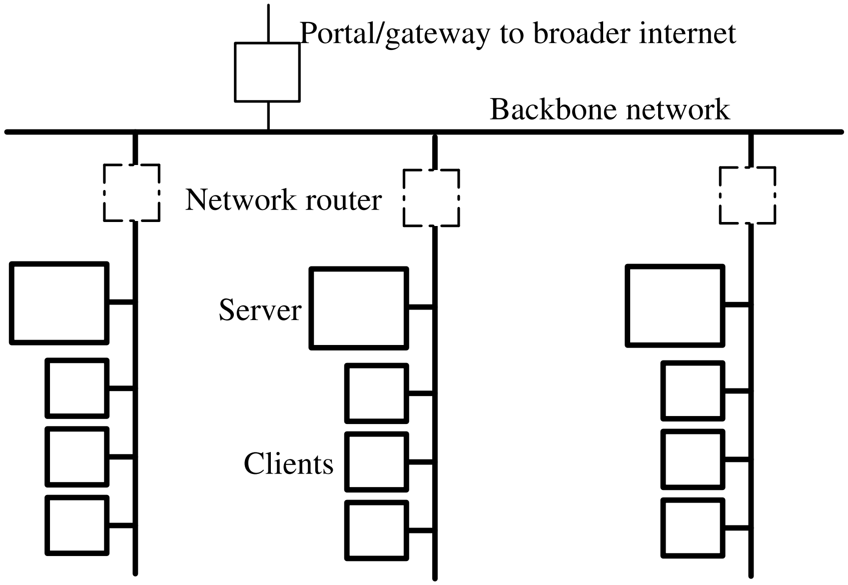
If you're very concerned about reliability, associate two servers with each group of clients rather than just one. And use these additional servers to connect each network to a second backbone network. With this *redundant connectivity* topology no single failure can disconnect a client from the rest of the network.

Figure 4
Topology 3 of 4 – Redundant Connectivity



If you prefer, you can use separate boxes to interconnect the networks rather than adding a second Ethernet interface to each server. This removes some non-NFS load from the servers. The separate boxes that connect this *dedicated infrastructure* topology could be "routers" or "brouters" from a networking vendor, or they could be additional workstations fitted with a second Ethernet interface and dedicated to this use.

Figure 5
Topology 4 of 4 – Dedicated Infrastructure



These topology choices are not mutually exclusive. You can combine them in many ways. Here are two examples:

Figure 6
Combined Topology – Hierarchy and Two–Tiered

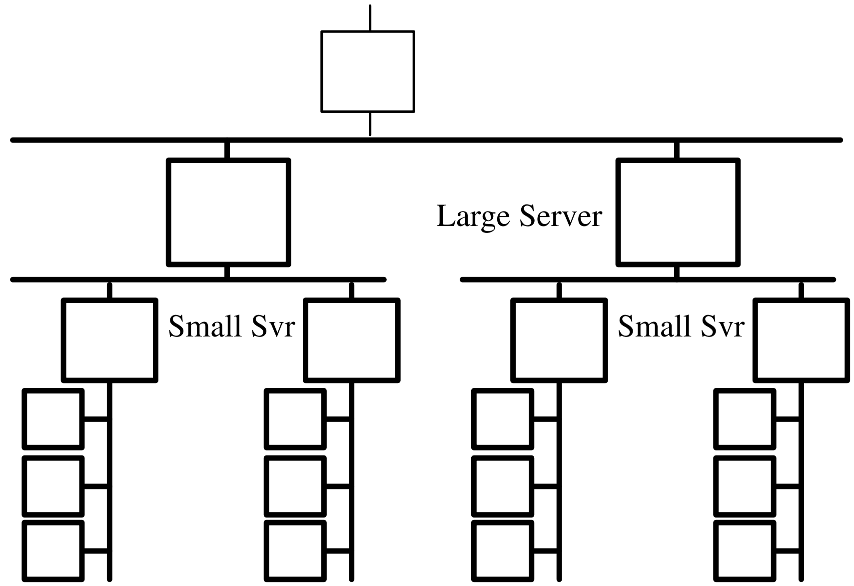
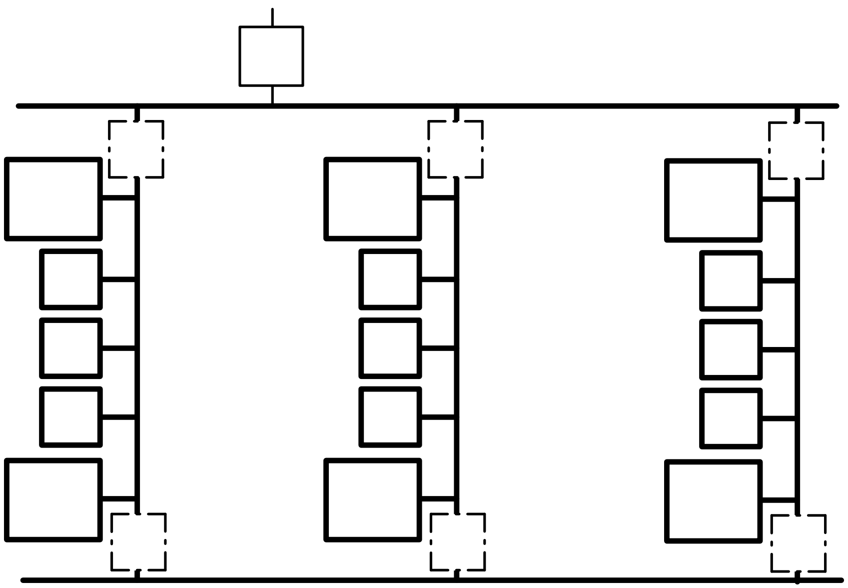


Figure 7
Combination Topology – Redundancy *<u>and</u>* *Dedication*



Networks are often connected in a hierarchy of increasingly larger internetworks. Your group of networks in one building may in turn be part of an even larger group of networks spread over a whole campus of buildings. If your group of networks is part of a larger group of networks, you should dedicate a system to be your connection to the next higher layer. This connection to the outside should insulate your networks from problems that occur elsewhere. So use a sophisticated router or brouter rather than a simple bridge.

What Kind of Media

Ethernet/802.3 networks run over several different kinds of wiring, just as several different kinds of pipe are commonly used to plumb houses. Network performance is essentially identical for all types of wiring, just as water flows equally well through iron, copper, or plastic pipe. All the networks run at the same 10 Mbits/sec serial bit rate, and all use Carrier Sense Multiple Access with Collision Detection (CSMA/CD) as their arbitration method.

There are three common kinds of Ethernet/802.3 network wiring:

- ♦ Thicknet, also called thick coax or yellow cable, was the first Ethernet media. It isn't much in vogue any more. It's heavy and cumbersome, especially in buildings that don't have drop ceilings. It's expensive to buy and labor–intensive to install. And it's hard to trouble shoot.
- ♦ Thinnet, also called thin coax or black cable or cheapernet, is lighter and more flexible. Some hosts and host adapter cards have the transceiver onboard, so that the backpanel contains a BNC connector that attaches directly to the coax rather than a 15–pin connector for a transceiver or transceiver cable. This direct attachment with no drop cable has its drawbacks. Carrying a loop of coax all the way to the host means two cables going to the host rather than just one. And if a user accidently disconnects the main cable from the 'T' connector, all other stations on that cable segment will be affected.
- ♦ Twisted pair, also called TP or UTP, is the newest Ethernet media. While its materials cost may seem higher than thinnet, its labor costs are low and its maintainability is excellent. The wires form a star pattern with each host directly connected to a hub. This topology makes installation and maintenance much easier, and makes it harder for a problem with one host to affect other hosts. It also makes troubleshooting easier since different sections of the network can be easily isolated just by disconnecting and

reconnecting hubs and since many hubs have network management capability built–into them. Another advantage of twisted pair Ethernet hubs is that the wiring that interconnets the hubs isn't necessarily Ethernet. You can have twisted pair Ethernet from the system to the hub in the closet, and at the same time use fiber, perhaps at FDDI rates, to interconnect the hubs.

> From the point of view of a host, the network operates the same even though the topology looks different. The host will see the same 10 Mbit/sec interface and the same collision detection signal that it's used to. The fact that the cable topology is actually a star rather than a bus, and the fact that the collision detection signal is generated by a hub, are transparent to the host.

> Several vendors introduced twisted pair products before the standards process was complete, so not all existing twisted pair products interwork with each other. If you need to know what works with what, ask your network vendor.

There are also more exotic kinds of Ethernet/802.3 network wiring, including fiber optic, infrared, microwave, and radio.

In the past the choice of which type of wiring to use for a network has been subject to the same kinds of tradeoffs as choosing which type of pipe to use for a house: labor costs vs. materials costs; amateur vs. professional installation; and materials vs. maintenance costs. Today the choice is clear: all new Ethernet installations should use twisted pair wiring for home runs (from offices to wiring closets).

Another similarity between network wiring and house plumbing is that media can be mixed. The plastic pipe in the new section of a house may be connected to the iron pipe in the old section via special couplings. Likewise new twisted pair may be connected to existing thicknet via a twisted pair hub with a transceiver cable connector on it.

Today you'll probably need to put a small external box between each system and the network no matter which kind of wiring you choose. For thicknet the box is clearly a transceiver: a transceiver cable connects to the 15-pin connector on one side and the other side clamps around the coax. For thinnet too the box is a transceiver even though many people refer to it as a "thick to thin converter." Recently thanks to the continued miniaturization of the electronics thinnet transceivers so small they can be connected directly to the host backpanel without an intervening transceiver cable have appeared. For twisted pair the box contains a specialized transceiver that communicates with a hub rather than a piece of coax. (Many people naively assume the box contains nothing more than a twisted pair line driver [balun] and refer to it as a "thick to twisted converter.") Boxes with a thinnet connection on one side and a twisted pair connection on the other side have also appeared recently. Inside these "thinnet to twisted pair converters" is a miniature Ethernet repeater.

Sometime in the future you can expect that all new systems will have a twisted pair Ethernet interface built in. The 15–pin 'D' shaped transceiver connector is going the way of the dinosaur.

For very small test networks a "multiport transceiver" box in loopback mode with a transceiver cable from the box to each host works well. For larger laboratory or test networks thinnet is a good choice. When wiring a new building, use twisted pair. Twisted pair is also a good choice for expanding an existing building wide network, especially if the building has wiring closets.

If you're looking at a large installation, call in a network wiring expert. I unclog my own sinks, but when I'm remodeling I call a professional plumber. If you really want to do it all yourself, ask *how much would it cost my company for the network to be non–operational for one day?* Then ask *how much do professionals charge for their cable installation and/or maintenance service?*

EXAMPLE CONFIGURATION

Here's an example showing how you can use $NFSops_{XX}$ to be sure that all parts of your network can handle the expected file sharing load.

There are two groups of client systems. Group A has 80 dataless systems, each of which supports one user. The systems all run the same automated office workload, which generates 6.1 $NFSops_{BW}$. Each user is allocated 10MB of online disk storage. The systems are used only part time, and different users are active at different times of the day, so Peak Usage is 20%.

Group B has 20 diskfull systems which are shared by 75 users. They all run the same CAD/CAM workload, which generates 4.4 $NFSops_{NB}$. Each user is allocated 40MB of online disk storage. Most systems are in use most of the time, so Peak Usage is 90%.

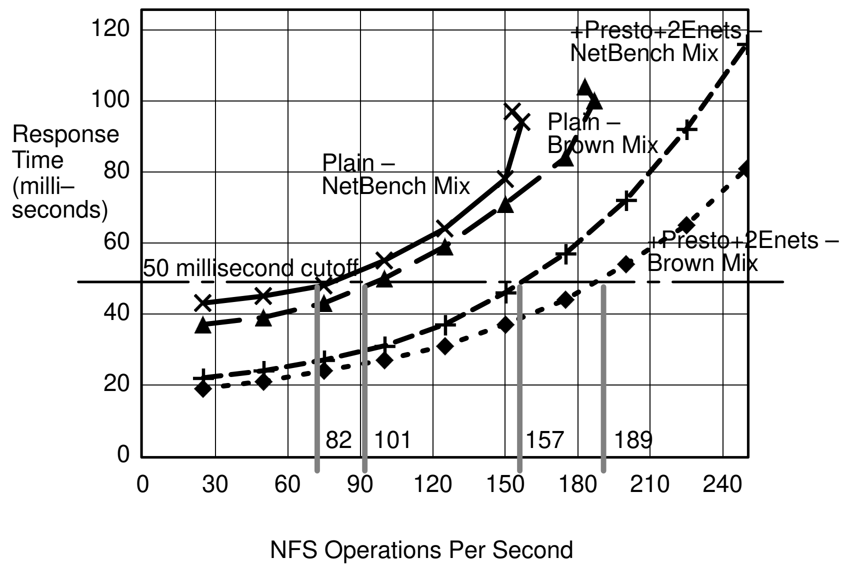
Calculation tells us that the client systems in Group A generate a total load of 98 $NFSops_{BW}$, and that the Group A users need a total of 800MB of online disk storage for individual use. Likewise the client systems in Group B generate a total load of 79 $NFSops_{NB}$, and the Group B users need a total of 3000MB of online disk storage for individual use.

Suppose only two server models are available. And suppose the NFSops ratings for the two server models are as shown in Figure 8 . (The data in Figure 8 was made up for this example.)

Suppose all the servers provide all network services in addition to file service. De–rate the servers by 10% since they're providing name service, another 20% since they're providing packet forwarding service, and another 10% since they're providing email routing service.

Figure 8

Imaginary Server Specs (Used In Example)



Group A wants to use at least two file servers so that even if one fails, half the users can keep working. Group B is willing to use only one file server if a single large one can satisfy their performance and storage requirements. The two groups don't want to share file servers.

Configure two smaller servers for Group A. The smaller server, rated at 101 $NFSops_{BW}$ @ 50 milliseconds, meets the NFS requirement and has enough power left over to provide name service and route email and forward packets. One 669MB drive on each server meets their space requirements.

A single larger server with two 1 GB disk spindles will satisfy Group B's needs. The larger server, rated at 157 $NFSops_{NB}$ @ 50 milliseconds, meets the NFS requirement and has enough power left over to provide name service and route email and forward packets.

Interconnecting Group B's clients and server will require two Ethernets, both because the expected 98 $NFSops_{NB}$ total load slightly exceeds the capacity of one Ethernet and because the lines for the larger server in the graph assume both additional Ethernet interfaces are being used. One Ethernet has enough capacity to interconnect all of Group A's 2 servers and 80 clients. (If there's any possibility group B will grow in the next couple of years, use two Ethernets with one server and 40 clients on each right from the beginning.)

Figure 9 shows the final recommended network configuration for the example. Figure 10 shows an actual network configuration.

Figure 9

Final Recommended Network Configuration for Example

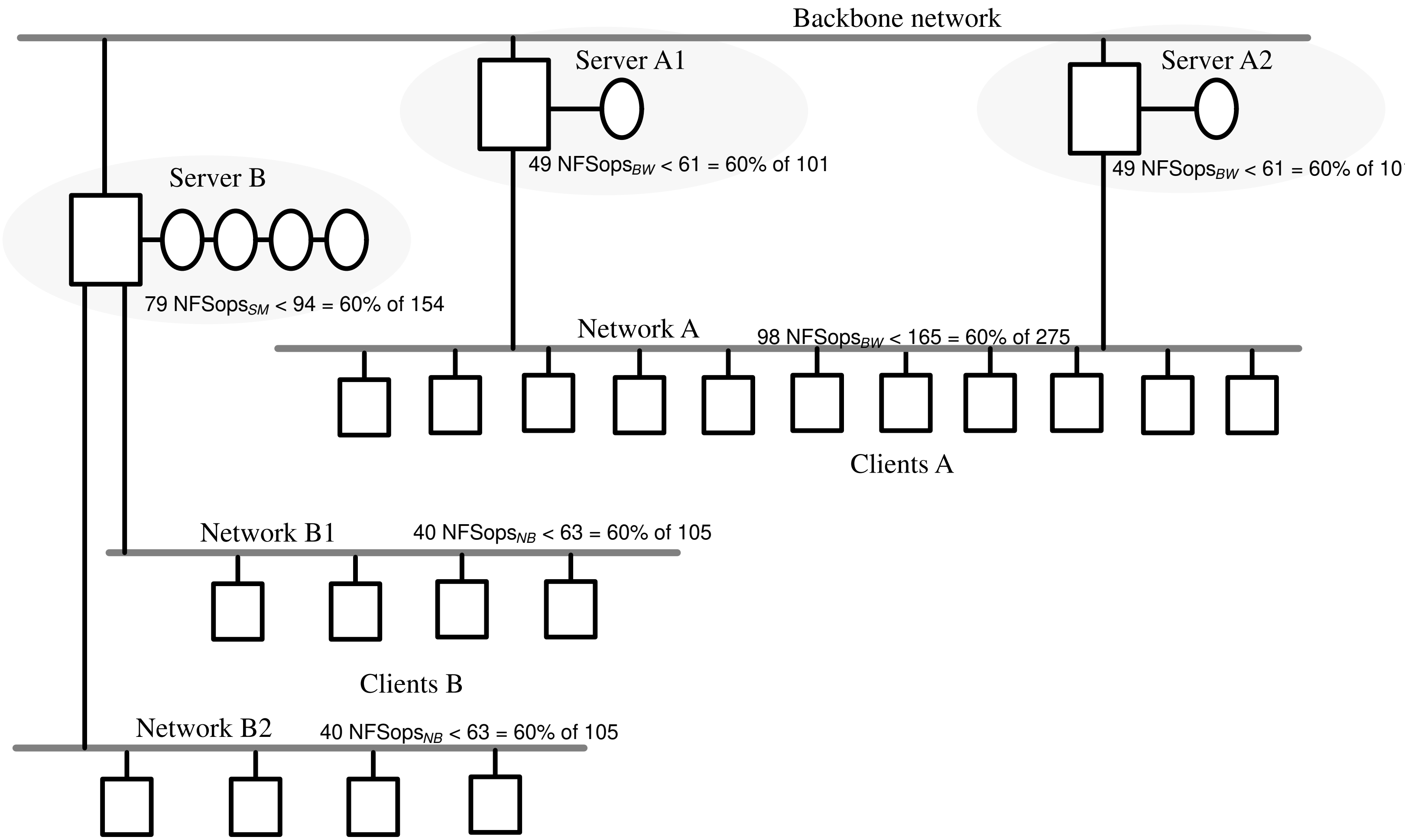
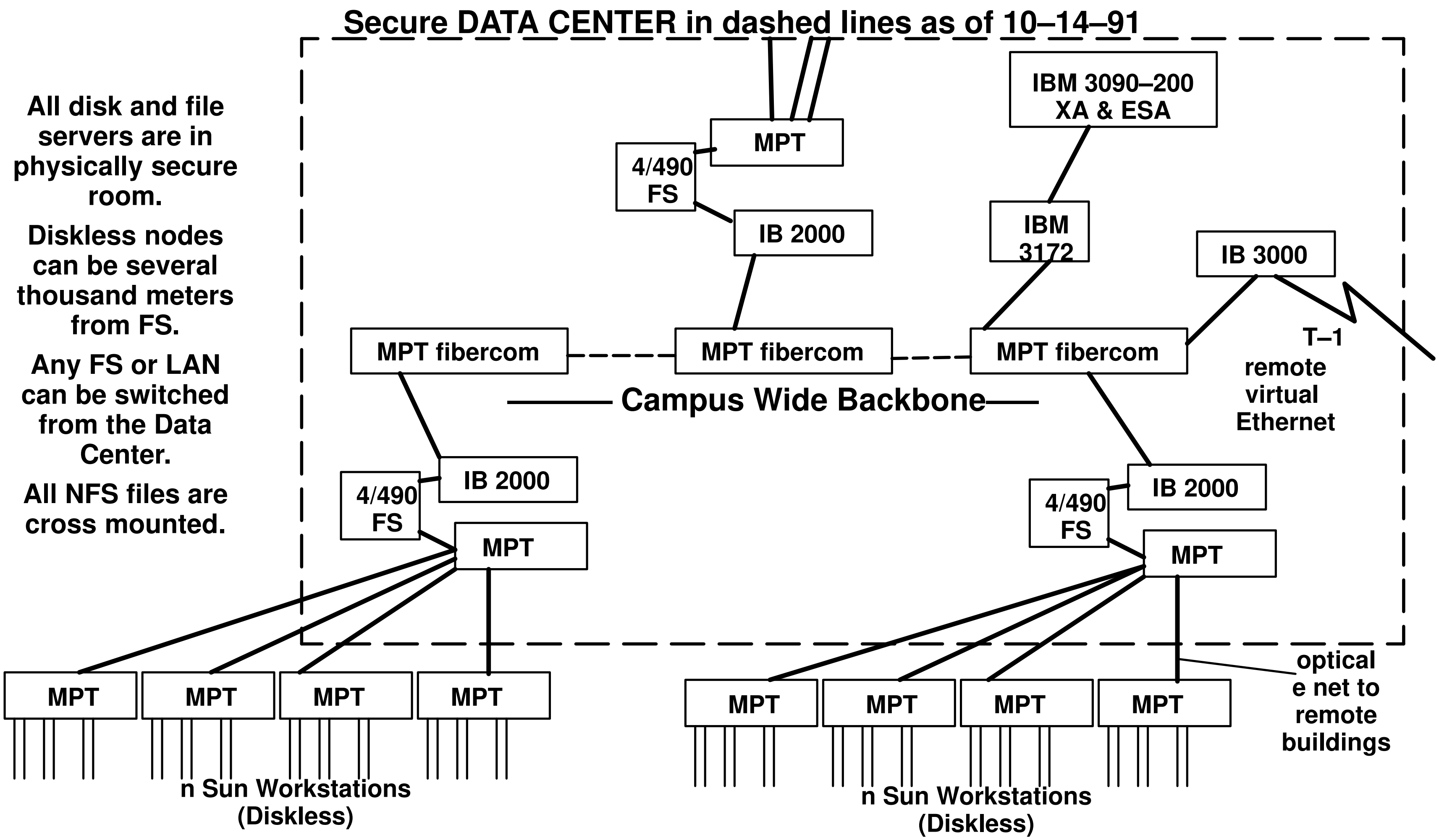


Figure 10

An Actual Network Configuration



Some 4/490 File Server's have over 40 diskless workstations on one LAN. The CISCO (or others) eliminate all gateway traffic on the 4/490 FS increasing available disk and cpu bandweith.

APPENDIX A – LADDIS/NHFSSTONE BENCHMARK

History

Nhfsstone version 1 was created at Legato. It synthesized client load by making file system calls. It has been widely distributed.

After several minor additions and one major change it became Nhfsstone version 2. The major change is that nhfsstone version 2 synthesizes client load by making NFS/RPC calls directly rather than by making file system calls. Although it's been used extensively internally by some vendors, it's never been widely distributed.

With more changes to both the code and the rules for using it, it became LADDIS. When it is complete LADDIS will be added to the SPEC suite. Currently PRE-LADDIS is available from SPEC under license for beta testing and feedback. You may see a few numbers generated by preliminary versions of LADDIS even though all SPEC members have agreed not to publish LADDIS results until LADDIS is finalized.

Nhfsstone is pronounced n-f-s-stone. The silent 'h' continues the tradition of w<u>h</u>etstone, d<u>h</u>rystone, etc. LADDIS is an anronym representing the six companies that participated in its development: Legato, Auspex, Digital, Data General, IBM, and Sun.

Metrics

LADDIS/Nhfsstone reports its results as "NFS Operations Per Second" versus response time. We try to abbreviate "NFS Operations Per Second" as *NFSops* consistently . You will also see the acronyms NFSOPS and NOPS.

The abbreviation NFSIOPS looks very similar to the abbreviation NFSOPS but is <u>not</u> the same. A conventional way to rate hard disks is in terms of Input/Output operations per second, usually abbreviated IOPS. After replacing a hard disk with a remote disk it would to make sense to talk about NFSIOPS. This metric would count any operation that actually caused disk activity, which is <u>not</u> the same as the NFSops measured by LADDIS/nhfsstone.

The reality of NFSIOPS is even worse than the theory. Different folks define NFSIOPS differently. Some define it as all NFS *read* and *write* operations but none of the other kinds of NFS operations. Some define it as the actual number of calls to the local disk driver on behalf of NFS. And some convert NFSops to NFSIOPS using a mystery formula.

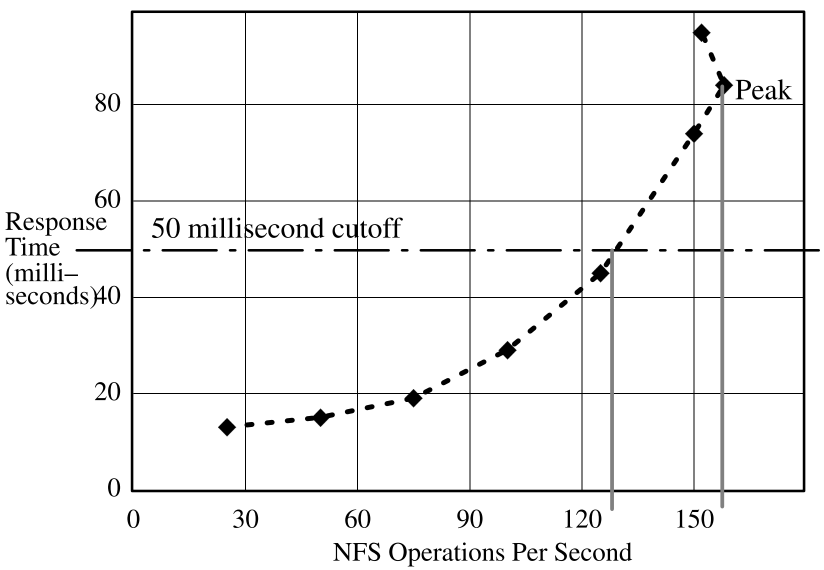
Results

Figure 11 shows how LADDIS/nhfsstone measurements of a particular server published by its vendor might look. The plotted line shows average NFS request response time against the total number of

NFS operations being generated and responded to. Most but not all graphs put load on the horizontal axis and response time on the vertical axis.

Figure 11

Imaginary Published Server Specs (One Mix)



There is a strong push to reduce the line to a single number. One way to do this is to choose a "cutoff" response time, then read the load at the point where the test line crosses the cutoff line. Some published results used a cutoff of 70 milliseconds. 50 milliseconds was another commonly used cutoff. This document uses a 50 millisecond cutoff because it's a nice round number. Great client performance really requires a latency of less than 40 milliseconds. An even lower latency of 25–30 milliseconds is recommended for use with diskless clients. Particular applications, configurations, or responsiveness needs may require using some other specific cutoff.

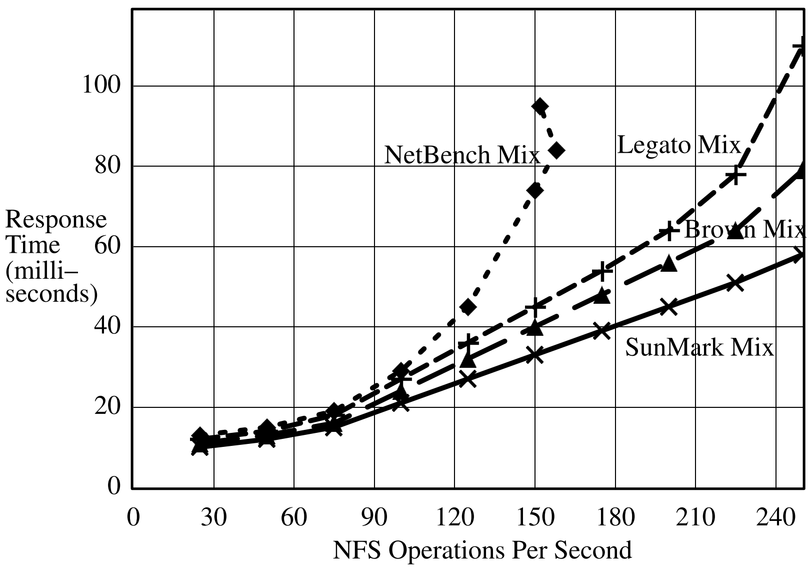
Try to obtain the entire line graph. If you have the entire graph you can choose later what average response time cutoff to use, and can even change your mind and recompute. If you have the entire graph you can see the shape of the curve and know how the server will behave if you overload it.

The recent trend is to reduce the line to a single number by quoting the "peak" NFSops value, the highest one regardless of response time. This practice, which would have been misleading in the past, is usually reasonable today. Most new model servers are so fast that the entire line falls below any reasonable response time cutoff. So the "peak" value is a fair measure of their capability. Nevertheless insist that the quoted capacity include the response time as well as the operation rate, e.g. *X*NFSops@*Y*milliseconds.

There should be one line on the graph for each NFS operations test mix. Figure 12 shows test results for a particular server using all four “standard” NFS operations mixes.

Figure 12

Imaginary Published Server Specs (Four Mixes)



Limitations of the Benchmark

LADDIS/nhfsstone measures the average response time for all NFS operations in a particular mix over a period of time. It does not measure the response time to different operations separately. Nor does it measure the variation in response time.

Its results are useful for *capacity planning*. Its results are not useful for predicting either the instantaneous responsiveness of a particular workstation or the time required to perform a particular network–intensive operation such as copying a file.

Obtaining the Benchmark Program

You can lay out networks without ever running LADDIS/nhfsstone yourself. Vendors publish lots of numbers. The rest you can do with nfsstat and a pocket calculator.

If you want to run LADDIS/nhfsstone yourself, you can obtain a copy several ways.

- ♦ send email
  `To: request@legato.com`
  `Subject: send unsupported nhfsstone`
  (or with bang addressing—
  `To: uunet!legato.com!request`
  `Subject: send unsupported nhfsstone`
- ♦ use ‘archie‘ to locate an archived copy on the Internet and fetch it with anonymous FTP
- ♦ get a copy from SPEC — start by sending email
  `To: spec–preladdis–beta–test@riscee.pko.dec.com`
  `Subject: Request SPEC PRE–LADDIS Beta Test License`

Issues

LADDIS/nhfsstone can give very different results depending on how it is run. Here are some issues:

One issue is the number load generators on each network and the number of child processes on each load generator. Tests show that response time gets slightly longer when there are more child processes. It’s not clear that the default number of child processes is correct. Should the number of child processes be specified per load generator, per server under test, or per server network interface? If it’s necessary to use more than one machine to generate the desired load, what should be kept constant: the number of children per machine or the total number of children? Currently adding a load generator means using more server disk space. But the larger server data space can affect server caching and paging. So the results with two generators each producing X/2 load can be significantly different from the results with one generator producing X load. This issue can be finessed by decreeing that there should be only one load generator per Ethernet.

LADDIS/nhfsstone builds its test disk data on the fly by random operations, and so takes a while to settle down to a steady state. If a test run is too short, the results will reflect startup effects. You should run all nhfsstone tests for at least 300 seconds. LADDIS automatically excludes the startup period from its results.

Nhfsstone doesn't fully capture the differences in NFS load generated by different kinds of clients. It allows the mix of operations to be specified. But it doesn't allow either the pattern of operations or the pattern of data transfer sizes to be specified. It assumes that the sequence of operations is random. And it assumes that the pattern of data transfer sizes is heavily skewed toward short transfers with a few very long multi–block transfers. These assumptions aren't necessarily true. For example the load actually generated by a SPARCstation does back–to–back write operations much more frequently than LADDIS/nhfsstone. Likewise the load actually generated by a SPARCstation that's loading executables or shared libraries from a remote system does large block transfers much more frequently than LADDIS/nhfsstone.

Nhfsstone shares the same small exported disk partitions to all clients. This can reduce disk head movement well below real world values or even allow all NFS read requests to be answered out of RAM, resulting in greatly inflated results. LADDIS only partially solves this problem. Although its disk space size is larger than nhfsstone, it's still possible with clever use of striping of a large number of disks to eliminate almost all disk head movement.

Results generated by Nhfsstone version 1, nhfsstone version 2, and LADDIS are not always similar. Depending on the test environment, results from different versions of the benchmark have been seen to differ by more than a factor of two. Nhfsstone version 2 results are often higher than either nhfsstone version 1 results or LADDIS results.

Nhfsstone version 1 results can be distorted by client–side caching. Nhfsstone version 1 attempts to beat the client–side cache by peeking at the kernel to see how many NFS operations actually occurred, then requesting more if needed. Thus any request that's answered out of a client–side cache gets reissued. Despite this attempt to beat the client–side cache, the pattern of requests may become skewed, or client RAM may thrash, or both. If the client–side cache is very very good, nhfsstone may just give up and print a warning message about its results being questionable.

Later versions solve this problem by issuing requests directly to NFS rather than to a vnode, thus bypassing any client–side caches. This makes the benchmark behavior and results correspond more closely to the output of nfsstat. This change also makes it more obvious that the benchmark measures server capacity not client efficiency.

Server–side data caches are not a problem since the behavior LADDIS/nhfsstone measures is the same behavior real clients will see. In other words, LADDIS/nhfsstone measures among other things the effectiveness of server–side data caches. Server–side file name caches however can be a problem. The number and pattern of file name sizes used by LADDIS/nhfsstone may be very different from the number and pattern of file name sizes used by a real client, causing the server–side filename cache to be either much more or much less effective than it would be with a real client. This problem is especially severe with version 1 of nhfsstone. version 1 uses very long filenames in an effort to beat the client–side cache. Unfortunately these very long filenames also reduce the effectiveness of the server–side cache and so cause server performance to be underreported.

APPENDIX B – KNOWN ACTUAL LOADS

Actual load varies as much as a couple orders of magnitude from site to site and application to application.

The Survey

To measure an actual load, run the survey script at a workstation while it's busy. The survey is provided in a form that you can fetch from the same place you fetched this document, then email to users. Users can easily unwrap and run it. It will automatically email its results back to the person who sent it (you).

Please forward a copy of the results by email To: <survey@deitrick.East.Sun.COM>. All the results received there will be summarized and included in the next revision of this document.

Fetch the survey script from the same place you fetched this document. If you fetched this document from a file named *mumble*_doc.ps.Z, fetch the survey script from a file named *mumble*_survey_script.txt.

The Bourne shell script is partially uuencoded to make it easier for typical users to unwrap and run it. If you want to examine the script, uudecode it. The script collects information from a variety of standard utilities such as ps and nfsstat without peeking inside the kernel or modifying the system in any way. It puts itself and all its working files in /tmp so all vestiges of it will disappear from the system the next time /tmp is purged.

The Results

Here are the survey results received to date. To see the latest version of this table, fetch a fresh copy of this document.

<u>You can help grow this table</u>. Fetch a fresh copy of the encoded, email–able survey script and forward it to your end users or administrators. When you get the results back, email a copy To: survey@deitrick.East.Sun.COM. The information you submit will be made available to everyone in the next revision of this document.

Loads Generated by Client Workstations

Reference#	Sample Date–Survey Version	Type of Organization	Type of Use	Frequency of Use	Window System	Primary App	#Applications	System Type	OS	RAM/Swap	#Local Disks	root/swap/usr	Home/Dynlib	Automounter	Route Daemon	nfsds/biods	NFSops	NFS Operation Mix
1	1/18/91–2	High Tech	SW Devel	Much	Sunview	?	6	4/65	S4.1–C	16/20	2	ll?	??	n	Y	8/4	00.00	0, 14, 0, 0, 56, 0, 24, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2
2	1/18/91–2	High Tech	Sales	Much	OW	?	9	4/65	S4.1.1–P	24/40	2	ll?	??	n	n	8/4	00.30	0, 58, 0, 0, 15, 0, 9, 0, 10, 0, 0, 0, 0, 0, 0, 0, 4, 0
3	1/21/91–2	High Tech	Tech Support	Constantly	OW	?	6	4/65	S4.1.1–C	8/16	2	ll?	??	Y	Y	8/4	00.12	0, 22, 0, 0, 9, 16, 45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0
4	1/21/91–2	High Tech	SW Devel	Constantly	OW	?	10	4/60	S4.1–C	16/30	3	ll?	??	n	n	4/4	00.01	0, 33, 1, 0, 28, 0, 15, 0, 14, 0, 0, 0, 0, 0, 0, 0, 3, 0
5	1/21/91–2	High Tech	Net Mgmt	Constantly	other	Sun Net Mgr	13	4/75	S4.1.1–C	64/64	2	ll?	??	n	n	0/4	00.48	0, 32, 3, 0, 35, 1, 14, 0, 2, 2, 1, 1, 0, 0, 0, 0, 4, 0
6	1/21/91–2	High Tech	SW Devel	Much	OW	?	11	4/60	S4.1.1–C	12/30	2	ll?	??	Y	n	8/4	00.71	0, 18, 0, 0, 36, 23, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 0
7	1/23/91–2	High Tech	SW Devel	Half	OW	?	8	4/60	S4.1–C	16/14	2	ll?	??	n	Y	0/4	00.11	0, 66, 0, 0, 17, 1, 10, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0
8	1/23/91–2	High Tech	System Mgmt	Constantly	OW	?	9	sun3	S4.1.1–C	12/63	1	ll?	??	n	n	0/4	00.00	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
12	1/25/91–2	High Tech	SW Devel	Much	Sunview	?	19	4/60	S4.1–C	16/41	2	ll?	??	n	Y	0/4	00.90	0, 27, 0, 0, 45, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 19, 0
13	1/18/91–2	High Tech	M/E Design	Constantly	OW/other	?	12	386i	S4.0.2–C	16/49	1	ll?	??	n	Y	0/4	00.45	0, 48, 1, 0, 22, 6, 5, 0, 5, 2, 1, 0, 0, 0, 0, 0, 6, 0
14	1/25/91–2	Educational	Sci/Bas Resrch	Half	other	?	8	sun4c	S4.0.3c–P	8/14	2	ll?	??	Y	n	8/4	00.35	0, 46, 0, 0, 31, 7, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
15	1/23/91–2	Educational	Sci/Bas Resrch	Much	Sunview	?	6	sun3x	S4.0.3–C	8/14	1	ll?	??	n	Y	8/4	00.15	0, 51, 0, 0, 32, 1, 7, 0, 1, 0, 1, 0, 0, 0, 0, 0, 2, 0

Loads Generated by Client Workstations

Reference#	Sample Date–Survey Version	Type of Organization	Type of Use	NFSops	Frequency of Use	Window System	Primary App	#Applications	System Type	OS	RAM/Swap	#Local Disks	root/swap/usr	Home/Dynlib	Automounter	Route Daemon	nfsds/biods	NFS Operation Mix
19	1/23/91–2	Educational	Sci/Bas Resrch	00.00	Constantly	Sunview	?	7	386i	?–?	?/15	?	???	??	Y	n	0/0	0, 14, 0, 0, 19, 61, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5
20	1/24/91–2	Educational	Sci/Bas Resrch	00.00	Constantly	Sunview	?	11	sun4c	S4.0.3c–P	8/14	1	ll?	??	n	Y	0/4	0, 29, 0, 0, 49, 0, 7, 0, 3, 0, 1, 0, 0, 0, 0, 0, 5, 1
21	1/24/91–2	Educational	Sci/Bas Resrch	03.13	Constantly	Sunview	?	6	386i	S4.0.1–P	8/32	2	ll?	l?	Y	Y	4/4	0, 18, 0, 0, 60, 0, 5, 0, 4, 0, 0, 0, 0, 0, 0, 8, 0
22	1/28/91–2	Medical	SW Devel	01.62	Constantly	none	?	10?	sun4	S4.0.3c–C	32/154	2	ll?	??	n	Y	8/4	0, 25, 0, 0, 41, 10, 3, 0, 8, 2, 0, 0, 0, 0, 0, 5, 0 (appears to be a multi-user server)
23	1/29/91–3	High Tech	Marketing	00.09	Constantly	Sunview	mail	6	386i	?–?	?/15	1	lll	R?	Y	Y	4/4	0, 19, 0, 0, 20, 25, 26, 0, 4, 2, 0, 0, 0, 0, 0, 0, 0, 0
24	11/21/90–X	High Tech	SW Devel	10.47	Much	Sunview	?	?	4/20	S4.1–S	16/?	0	RRR	R?	?	n	0/4	0, 42, 0, 0, 18, 0, 4, 0, 18, 17, 0, 0, 0, 0, 0, 0, 0, 0
25	2/11/90–X	Educational	SW Devel	02.46	Constantly	Sunview	? (SunMark)	?	4/60	S4.0.3–C	8/14	1	lll	R?	?	?	4/4	0, 38, 0, 0, 35, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 22, 0
26	2/11/90–X	Educational	SW Devel	03.37	Constantly	Sunview	? (SunMark)	?	4/60	S4.0.3–C	8/14	1	llR	R?	?	?	4/4	0, 41, 0, 0, 28, 5, 5, 0, 1, 0, 0, 0, 0, 0, 0, 16, 0
27	2/11/90–X	Educational	SW Devel	04.44	Constantly	Sunview	? (SunMark)	?	4/60	S4.0.3–C	8/16	0	RRR	R?	?	?	0/4	0, 38, 0, 0, 24, 8, 5, 0, 7, 0, 0, 0, 0, 0, 0, 13, 0
28	1/18/91–2	High Tech	M/E Design	00.58	Much	OW	?	4	4/65	S4.1.1–P	64/64	2	ll?	??	Y	n	12/4	0, 17, 0, 0, 17, 35, 6, 0, 15, 0, 0, 0, 0, 0, 0, 4, 0
29	2/1/91–3	High Tech	SW Devel	05.16	Constantly	Sunview	none	7	3/80	S4.0.3–S	16/32	1	llR	??	n	n	8/4	0, 33, 0, 0, 50, 4, 6, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0
30	6/12/90–X	High Tech	Tech Support	?	?	?	?	?	3/80	?	8/?	0	RRR	R?	?	?	?/?	0, 20, 0, 0, 22, 26, 6, 0, 21, 0, 0, 0, 0, 0, 0, 0, 0, 0
30b	6/12/90–X	High Tech	Tech Support	?	?	?	?	?	3/80	?	8/?	1	lll	??	?	?	?/?	0, 46, 0, 0, 16, 10, 23, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
31	2/10/91–3	High Tech	SW Devel	05.16	Constantly	Sunview	none	6	3/80	S4.0.3–S	16/32	1	llR	R?	n	n	8/4	0, 33, 0, 0, 50, 4, 6, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0 (appears to be duplicate of reference# 29)
32	2/10/91–3	High Tech	SW Devel	05.16	Constantly	Sunview	none	6	3/80	S4.0.3–S	16/32	1	llR	R?	n	n	8/4	0, 33, 0, 0, 50, 4, 6, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0 (appears to be duplicate of reference# 29)
33	2/1/91–2	High Tech	Tech Support	00.49	Constantly	OW	?	4	4/60	S4.1–C	16/34	3	lll	l?	n	Y	0/4	0, 32, 0, 0, 25, 2, 28, 0, 3, 0, 0, 0, 0, 0, 0, 5, 0

Loads Generated by Client Workstations

Reference#	Sample Date–Survey Version	Type of Organization	Type of Use	NFSops	Frequency of Use	Window System	Primary App	#Applications	System Type	OS	RAM/Swap	#Local Disks	root/swap/usr	Home/Dynlib	Automounter	Route Daemon	nfsds/biods	NFS Operation Mix
34	2/1/91–3	High Tech	SW Devel	00.06	Constantly	Sunview	none	18	4/65	S4.1–S	16/24	2	III	R?	n	Y	0/4	0, 51, 0, 0, 22, 0, 1, 0, 2, 1, 0, 0, 0, 0, 0, 0, 20, 0
35	2/13/91–4	High Tech	/Sci/Bas Resrch	00.09	Much	Unify	none	4	3/2?0	S4.0.3–C	16/29	3	III	II	Y	Y	8/4	0, 51, 0, 0, 9, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 0
36	2/15/91	High Tech	SW Devel	00.00	Much	OW	none	7	4/60	S4.1.1–C	16/64	3	III	Rm	Y	Y	8/4	0, 16, 3, 0, 25, 14, 21, 0, 9, 1, 0, 0, 0, 0, 0, 0, 5, 0
37	2/18/91–4	High Tech	SW Devel	00.08	Much	OW	none	8	4/65	S4.1.1–P	12/23	2	III	RI	Y	Y	0/4	0, 44, 0, 0, 20, 5, 17, 0, 5, 1, 0, 0, 0, 0, 0, 0, 2, 1
38	2/20/91–4	High Tech	Elec Pub	00.15	Once	OW	Interleaf	7	4/65	S4.1.1–P	12/30	3	III	RI	Y	Y	8/4	0, 34, 0, 0, 12, 44, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0
39	3/1/91–4	High Tech	SW Devel	01.16	Constantly	OW	none	7	4/20	S4.1–P	12/20	0	RRR	RR	Y	n	0/4	0, 23, 0, 0, 32, 21, 8, 0, 6, 0, 0, 0, 0, 0, 0, 0, 5, 0
40	3/4/91–4	High Tech	CAD Support	01.94	Much	Sunview	none	11	4/65	S4.1.1–P	8/140	1	IIR	RR	Y	n	0/4	0, 35, 0, 0, 19, 17, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0
41	3/1/91–4	High Tech	Elec Pub	00.00	Once	none	Mailtool	1	3/4?0	S4.1–P	32/24	2	III	II	n	Y	8/4	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
42	2/28/91–4	Educational	Sci/Bas Resrch	04.73	Constantly	OW	none	10	4/65	S4.1.1–C	28/38	2	III	Im	Y	Y	8/4	0, 36, 0, 0, 5, 4, 33, 0, 18, 0, 0, 0, 0, 0, 0, 0, 1, 0
43	3/8/91–2	?	SW Devel	00.95	Constantly	other	?	8	sun4c	?	?/54	?	II?	??	n	Y	8/4	0, 23, 2, 0, 44, 0, 14, 0, 7, 1, 1, 0, 0, 0, 0, 0, 3, 0
48	3/26/91–4	High Tech	Sys Admin	00.22	Constantly	OW	none	15	4/65	S4.1.1–C	16/68	2	III	RI	n	Y	0/4	0, 59, 0, 0, 23, 0, 9, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0
49	4/15/91–5	High Tech	Sci/Bas Resrch	02.49	Constantly	OW	“eum”	7	4/65	S4.1GFX–C	8/44	1	IIR	RR	Y	Y	0/4	0, 23, 0, 0, 34, 8, 13, 0, 2, 0, 0, 0, 0, 0, 0, 0, 17, 0
59	4/30/91–4	Manufacturing	SW Devel	00.68	Half	none	none	5	3/50	?–?	8/24	0	RRR	R?	n	Y	0/4	0, 36, 0, 0, 28, 20, 10, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0
60	9/19/91–3	?	?	00.70	?	other	?	7	4/65?	S4.0.3–P	8/54	1	III	R?	n	n	0/4	0, 26, 0, 0, 37, 24, 4, 0, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0
061	9/26/91–5	High Tech	SW Devel	00.00	Constantly	OW	none	10	4/65	S4.1.1–C (an independent system)	16/72	3	III	II	n	Y	8/4	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Loads Handled By Servers

Reference#	Sample Date–Survey Version	Type of Organization	Type of Use	NFSops	Frequency of Use	Window System	Primary App	#Applications	System Type	OS	RAM/Swap	#Local Disks	root/swap/usr	Home/Dynlib	Automounter	Route Daemon	nfsds/biods	NFS Operation Mix	
9	1/23/91–2	Educational	Sci/Bas Resrch	03.77	Constantly	none	?	10?	sun4	?–?	?/131	?	???	??	Y	n	8/8	0, 27, 0, 0, 33, 3, 13, 0, 13, 3, 2, 0, 0, 0, 0, 0, 1, 0	
10	1/24/91–2	Educational	SW Devel	03.06	Constantly	none	?	14?	490	S4.1–C	32/98	4	ll?	??	Y	n	8/8	0, 38, 0, 0, 30, 13, 6, 0, 3, 0, 0, 0, 0, 0, 0, 0, 4, 0	
11	1/23/91–2	Educational	SW Devel	04.50	Constantly	none	?	11?	490	S4.1–C	32/98	4	ll?	??	Y	n	8/8	0, 37, 0, 0, 30, 13, 6, 0, 3, 0, 0, 0, 0, 0, 0, 0, 4, 0	
16	1/23/91–2	Educational	Sci/Bas Resrch	01.04	Constantly	none	?	?	sun3	S4.0.3–C	8/4	5	ll?	??	Y	n	8/4	0, 32, 0, 0, 20, 7, 14, 0, 16, 1, 1, 0, 0, 0, 0, 0, 2, 3	
17	1/23/91–2	Educational	Sci/Bas Resrch	00.97	Constantly	none	?	?	sun4	S4.0.3–C	32/65	4	ll?	??	Y	n	8/4	0, 16, 1, 0, 23, 1, 26, 0, 23, 1, 1, 0, 0, 0, 0, 0, 2, 0	
18	1/23/91–2	Educational	Sci/Bas Resrch	04.40	Constantly	other	?	15?	sun4	?–?	?/127	5	ll?	??	Y	n	8/4	0, 13, 0, 0, 14, 2, 43, 0, 22, 0, 0, 0, 0, 0, 0, 0, 1, 0	
44	3/11/91–2	High Tech	SW Devel	29.21	Constantly	other	?	10	sun4c	?	?/54	?	ll?	??	n	Y	8/4	0, 5, 0, 0, 40, 1, 43, 0, 3, 0, 0, 0, 0, 0, 0, 0, 5, 0	
45	3/12/91–4	Manufac	SW Devel	06.69	Constantly	other	Emacs	11	4/60	?	?/54	3	lll	ll	n	Y	8/4	0, 5, 0, 0, 40, 1, 43, 0, 3, 0, 0, 0, 0, 0, 0, 0, 5, 0	(10 am; same machine as reference# 44)
46	3/14/91–4	Manufac	SW Devel	08.51	Constantly	OW+	Emacs	11	4/60	?	?/54	3	lll	ll	n	Y	8/4	0, 5, 0, 0, 40, 1, 43, 0, 2, 0, 0, 0, 0, 0, 0, 0, 5, 0	(4 pm; same machine as reference# 44)
47	3/22/91	Manufac	SW Devel	01.37	Constantly	OW+	Emacs	11	4/60	?	?/54	3	lll	ll	n	Y	8/4	0, 4, 0, 0, 40, 1, 44, 0, 2, 0, 0, 0, 0, 0, 0, 0, 4, 0	(11:15 am; same machine as reference# 44)
50	4/29/91–4	Medical	?	09.50	Much	none	none	1	3/2?0	?–?	16/32	2	lll	R?	n	Y	6/6	0, 23, 1, 0, 39, 6, 14, 0, 7, 1, 2, 0, 0, 0, 0, 0, 1, 0	
51	4/29/91–4	Medical	?	02.07	Much	none	none	1	3/2?0	?–?	16/24	5	lll	R?	n	Y	6/6	0, 17, 0, 0, 32, 10, 27, 0, 6, 0, 0, 0, 0, 0, 0, 0, 2, 0	
52	4/29/91–4	Medical	?	06.00	Much	none	none	1	3/2?0	?–?	16/32	2	lll	R?	n	Y	6/6	0, 23, 0, 0, 33, 4, 33, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0	
53	4/29/91–4	Medical	?	02.65	Much	none	none	2	3/2?0	?–?	16/?	3	lll	R?	n	Y	6/6	0, 22, 0, 0, 32, 11, 23, 0, 1, 0, 1, 0, 0, 0, 0, 0, 6, 0	
54	4/29/91–4	Medical	?	04.26	Much	none	none	1	3/2?0	?–?	16/32	2	lll	R?	n	Y	6/6	0, 24, 0, 0, 40, 8, 13, 0, 4, 0, 1, 0, 0, 1, 0, 0, 3, 0	

Reference# | Type of Use | Primary App | OS | root/swap/usr | Route Daemon
Sample Date–Survey Version | Frequency of Use | #Applications | RAM/Swap | Home/Dynlib | nfsds/biods
Type of Organization | Window System | System Type | #Local Disks | Automounter
NFSops | NFS Operation Mix

Loads Handled By Servers

55	4/29/91–4	Medical	?	Much	none	none	2	3/2?0	?–?	16/32	2	lll	R?	n	Y	6/6
			04.62	0, 23, 0, 0, 31, 19, 15, 0, 3, 1, 0, 0, 0, 0, 0, 0, 2, 0												
56	4/29/91–4	Medical	?	Much	none	none	1	3/2?0	?–?	16/32	2	lll	R?	n	Y	6/6
			13.04	0, 16, 0, 0, 40, 20, 11, 0, 4, 0, 2, 0, 0, 1, 0, 0, 3, 0												
57	4/29/91–4	Medical	?	Much	none	none	2	3/2?0	?–?	16/32	4	lll	R?	n	Y	6/6
			08.08	0, 18, 1, 0, 46, 8, 12, 0, 5, 1, 1, 0, 0, 2, 0, 0, 2, 0												
58	4/29/91–4	Medical	?	Much	none	none	11	3/1?0	?–?	8/48	1	lll	R?	n	Y	4/4
			00.00	0, 44, 0, 0, 39, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9				(communications server & NIS master)								

KEY:

Reference#	Index into archive of raw data
Sample Date–Survey Version	mm/dd/yy when sample taken – version number of survey script
Type of Organization	Financial, Retail, High Tech, Manufacturing, Educational, Medical, Other
Type of Use	Electronic Publishing, Manufacturing/Inventory Control, Mechanical/Electrical Design, Financial/Data Analysis, Software Development, Scientific/Basic Research, Image Display/Processing, Other
Frequency of Use	Constantly, Much of the time, Half the time, Once an hour, Once a day
Window System	Sunview, OpenWindows [version], other, none
Primary Application	name of application prog, or none if no one is primary, or ? if unknown
#Applications	approximate number of icons & windows on screen
System Type	Product line & exact product model if possible
OS	SunOS/SysV/AIX/Ultrix Version – Prebuilt(generic), Standard(supplied), Custom, N/A, ?unknown
RAM/Swap	in MegaBytes
#Local Disks	number of local hard disks of any size
root/swap/usr	for each partition: local, mixed, REMOTE, ?unknown
Home/Dynlib	local, mixed, REMOTE, ?unknown, n/a
Automounter	is the automounter daemon running on the machine: no, YES
Route Daemon	is the RIP routing daemon in.routed running on the machine: no, YES
nfsds/biods	number of nfsd processes / number of biod processes
NFSops	average NFS operations per second during heavy activity periods
NFS Operations Mix	percentages: null, getattr, setattr, root, lookup, readlink, read, wrcache, write, create, remove, rename, link, symlink, mkdir, rmdir, readdir, fsstat