

Sizing and Tuning NFS Clients and Servers

3/20/00

[Click here to start](#)

Table of Contents

Author: Rob Lucke

[Sizing and Tuning NFS Clients and Servers](#)

[Seminar Agenda](#)

[NFS Workgroup Environment](#)

[The NFS Client Drives the Network](#)

[Where is the Network Today?](#)

[NFS Design Approach](#)

[Seminar Agenda](#)

[NFS Fileserver Benchmark Data](#)

[NFS Fileserver Benchmark Data \(continued\)](#)

[NFS Fileserver Benchmark Data](#)

[SPEC_SFS_1 operation mixture](#)

[PPT Slide](#)

[PPT Slide](#)

[NFS Fileserver Benchmark Data](#)

[SPEC_SFS_2 Information](#)

[NFS Fileserver Benchmark Data Typical NFS
IOPS requirements**](#)

[NFS Fileserver Benchmark Data The NFS
IOP "Magic" Number, 4KB](#)

[Seminar Agenda](#)

[Measuring Client NFS Activity GPM Global NFS Activity Screen](#)

[Measuring Client NFS Activity GPM Client NFS PV2 & PV3 Operations](#)

[Measuring Client NFS Activity GPM Client NFS PV2 & PV3 Operations](#)

[Measuring Client NFS Activity NFS Activity with the nfsstat command](#)

[Measuring Client NFS Activity NFS Activity with the nfsstat command](#)

[Measuring Client NFS Activity NFS IOPS \(NFS I/Os per second = RPC calls per second\)](#)

[PPT Slide](#)

[PPT Slide](#)

[Seminar Agenda](#)

[NFS Fileserver Sizing Areas of Primary Concern](#)

[NFS Fileserver Sizing Secondary Areas of Concern](#)

[NFS Fileserver Sizing CPU Selection: SPEC_SFS_1 Ratings](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System](#)

[Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling](#)

[NFS Fileserver Sizing High Availability NFS](#)

[Seminar Agenda](#)

[NFS Fileserver Sizing Network Design](#)

[NFS Fileserver Sizing Network Design: Network Technologies](#)

[NFS Fileserver Sizing Network Loading: Ethernet Collision Domains](#)

[NFS Fileserver Sizing Network Loading: 10baseT Ethernet Utilization Curves](#)

[NFS Fileserver Sizing Network Loading: FDDI Token Ring](#)

[NFS Fileserver Sizing Network Loading: FDDI Token Ring](#)

[NFS Fileserver Sizing Network Loading: 100baseTX Collision Domains](#)

[NFS Fileserver Sizing Network Loading: Gigabit Ethernet \(GbE\)](#)

[NFS Fileserver Sizing Network Loading: Is GbE == ATM?](#)

[NFS Fileserver Sizing Network Loading:](#)

[ATM OC3 \(155.52 Mbps\)](#)

[NFS Fileserver Sizing Network Loading:](#)

[ATM OC12 \(622.08 Mbps\)](#)

[NFS Fileserver Sizing Routed Ethernet](#)

[Backbone](#)

[NFS Fileserver Sizing Ethernet Switching:](#)

[“Switch the LAN, route the WAN”](#)

[NFS Fileserver Sizing Layer 2 Switching](#)

[Technology](#)

[NFS Fileserver Sizing Layer 2 Switching](#)

[Technology](#)

[NFS Fileserver Sizing Layer 3 Switching](#)

[Technology](#)

[NFS Fileserver Sizing Network Link](#)

[Aggregation \(Trunking\)](#)

[NFS Fileserver Sizing But what about](#)

[“traditional” routers and ATM?](#)

[NFS Fileserver Sizing Examining the Network](#)

[Interface: lanscan](#)

[NFS Fileserver Sizing Examining the Network](#)

[Interface: landiag](#)

[NFS Fileserver Sizing Examining the](#)

[Network: Looking for Errors](#)

[NFS Fileserver Sizing Examining the](#)

[Network: Specific Notes](#)

[Seminar Agenda](#)

[NFS Fileserver Sizing Memory Subsystem](#)

[Seminar Agenda](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: Stand-Alone Disks](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: Mirroring Stand-Alone Disks](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: HP Smart Enclosures](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: SC-10/FC-10](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: Striping Stand-Alone Disks](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: RAID Modes](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: RAID Modes](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: Model 10 RAID Arrays](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: Model 20 RAID Arrays](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: Model 30/FC RAID Arrays](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: FC-60 RAID Array](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: XP-256](#)

[NFS Fileserver Sizing Mass Storage](#)

[Considerations: AutoRAID](#)

[File Server Sizing Mass Storage](#)

[Considerations: Channel Performance](#)

[File Server Sizing Mass Storage](#)

[Considerations: Channel Performance](#)

[File Server Sizing Mass Storage](#)

[Considerations: Fibre Channel](#)

[File Server Sizing Mass Storage](#)

[Considerations: Fibre Channel](#)

[File Server Sizing Mass Storage](#)

[Considerations: Fibre Channel](#)

[File Server Sizing Mass Storage](#)

[Considerations: Fibre Channel](#)

[Seminar Agenda](#)

[File Server Tuning File System Time Line](#)

[File Server Tuning HFS File System](#)

[File Server Tuning HFS File System](#)

[File Server Tuning HFS File System Structure](#)

[File Server Tuning HFS File System Structure](#)

[File Server Tuning HFS File System Structure](#)

[File Server Tuning Walking Directory Paths](#)

[File Server Tuning JFS File System](#)

[File Server Tuning JFS File System](#)

[File Server Tuning JFS mount options](#)

[File Server Tuning JFS mount options](#)

[File Server Tuning On-Line JFS operations](#)

[File Server Tuning On-Line JFS operations:
Examining Fragmentation](#)

[File Server Tuning On-Line JFS operations:
After Disk/Directory Defragmentation](#)

[NFS Fileserver Tuning NFS Client/Server
Components](#)

[NFS Fileserver Tuning NFS Fileserver Write
Components](#)

[NFS Fileserver Tuning NFS Fileserver Read
Components](#)

[NFS Fileserver Tuning Tuning Network](#)

[Components](#)

[NFS Fileserver Tuning HP-UX 9.X Issues with NFSD](#)

[NFS Fileserver Tuning HP-UX 10.X Issues with NFSD](#)

[NFS Fileserver Tuning How many NFSDs Do I Start?](#)

[NFS Fileserver Tuning Changing the Default Number of NFSDs](#)

[NFS Fileserver Tuning Measuring NFSD Usage](#)

[NFS Fileserver Tuning NFS UDP Socket Depth](#)

[NFS Fileserver Tuning NFS UDP Socket Overflow](#)

[NFS Fileserver Tuning Dedicate Buffer Cache to NFS](#)

[NFS Fileserver Tuning Measuring Buffer Cache Working Set Size](#)

[NFS Fileserver Tuning Measuring Buffer Cache Working Set Size](#)

[NFS Fileserver Tuning Initially Sizing Buffer Cache](#)

[NFS Fileserver Tuning NFS File Server Resource Contention](#)

[NFS Fileserver Tuning Disk Caching](#)

[NFS Fileserver Tuning Asynchronous NFS Mounts](#)

[NFS Fileserver Tuning Asynchronous NFS Mounts](#)

[NFS Fileserver Tuning Tunable Kernel Parameters: HFS inode cache](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: JFS inode cache](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: ninode](#)

[NFS Fileserver Tuning DNLC Cache Hit rate](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: Buffer Cache](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: Fixed Buffer Cache on 9.X](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: Dynamic Buffer Cache](#)

[NFS Fileserver Tuning The “syncer” Process](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: Other Kernel Parameters](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: Other Kernel Parameters](#)

[NFS Fileserver Tuning Tunable Kernel](#)

[Parameters: Monitoring Results](#)

[NFS Fileserver Tuning Controlling Client](#)

[Access: Exporting File Systems](#)

[NFS Fileserver Tuning The server’s NFS](#)

[mount daemon](#)

[Seminar Agenda](#)

[NFS Client Tuning NFS Client Components](#)

[NFS Client Tuning NFS Client 8 KB](#)

[Read/Write Size](#)

[NFS Client Tuning NFS Client 1 KB](#)

[Read/Write Size](#)

[NFS Client Tuning Changing the Default](#)

[Number of BIODs](#)

[NFS Client Tuning Number of Idle BIOD](#)

[Processes](#)

[NFS Client Tuning Tunable Kernel](#)

[Parameters](#)

[NFS Client Tuning Tunable Kernel](#)

[Parameters](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: BIOD buffer size](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Timeout Behavior](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Timeout Behavior Example](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Timeout Behavior Example](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Timeout Behavior Example](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Adjusting timeo and retrans](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Data Integrity](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Symptoms of Timeouts](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Finding Timeouts](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: hard/soft/intr](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Mounts from /etc/fstab](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Attribute Caching](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Attribute Caching](#)

[NFS Client Tuning NFS Client Mount](#)

[Parameters: Attribute Caching](#)

[NFS Client Tuning NFS Client Mount
Parameters](#)

[NFS Client Tuning NFS Client Automounter](#)

[NFS Client Tuning NFS Client Automounter
Master Map](#)

[NFS Client Tuning NFS Client Automounter
Direct Maps](#)

[NFS Client Tuning NFS Client Automounter
Indirect Maps](#)

[NFS Client Tuning Vendor Neutral
Automounter Maps](#)

[NFS Client Tuning Vendor Neutral
Automounter Maps](#)

[NFS Client Tuning “Hybrid” Automounter
Maps](#)

[NFS Client Tuning “Hybrid” Automounter
Maps](#)

[Seminar Agenda](#)

[NFS PV3 Features](#)

[NFS PV3 Features How do I get PV3?](#)

[NFS PV3 Features Additional Mount Options](#)

[NFS PV3 Features Additional Mount Options](#)

[NFS PV3 Features AutoFS](#)

[NFS PV3 Features CacheFS Terms](#)

[NFS PV3 Features CacheFS Operation](#)

[NFS PV3 Features CacheFS Operation](#)

[NFS PV3 Features CacheFS Operation](#)

[NFS PV3 Features CacheFS Operation](#)

[NFS PV3 Features](#)

[NFS PV3 Features NIS+](#)

[NFS PV3 Features NIS+](#)

[NFS PV3 Features NIS+](#)

[NFS PV3 Features](#)

[NFS Design Case Studies](#)

[NFS Design Case Studies Case Study 1:](#)

[“Your NFS server is terrible”](#)

[NFS Design Case Studies Case Study 1:](#)

[“Your NFS server is terrible”](#)

[NFS Design Case Studies Case Study 2: “My
server only handles half of my clients”](#)

[NFS Design Case Studies Case Study 2: “My
server only handles half of my clients”](#)

[NFS Design Case Studies Case Study 3: “My
software build performance is terrible”](#)

[NFS Design Case Studies Case Study 3: “My
software build performance is terrible”](#)

[NFS Design Case Studies Case Study 4:](#)

[“What can your NFS servers do?”](#)

[NFS Design Case Studies Case Study 4:](#)

[“What can your NFS servers do?”](#)



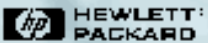
Sizing and Tuning NFS Clients and Servers



*Everything you wanted to know about using
NFS, but were afraid to ask*

Rob Lucke

**Americas Technical Computing
Advanced Technology Center**





Seminar Agenda

Design Issues

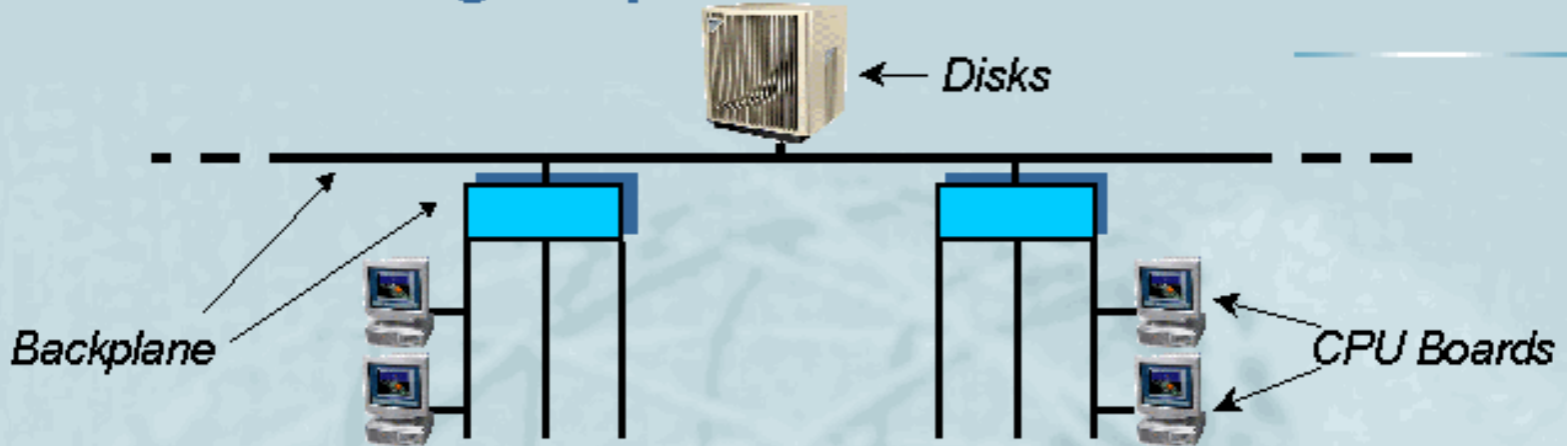
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



2 99/09/20 Rob Lucke



NFS Workgroup Environment

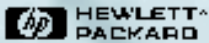
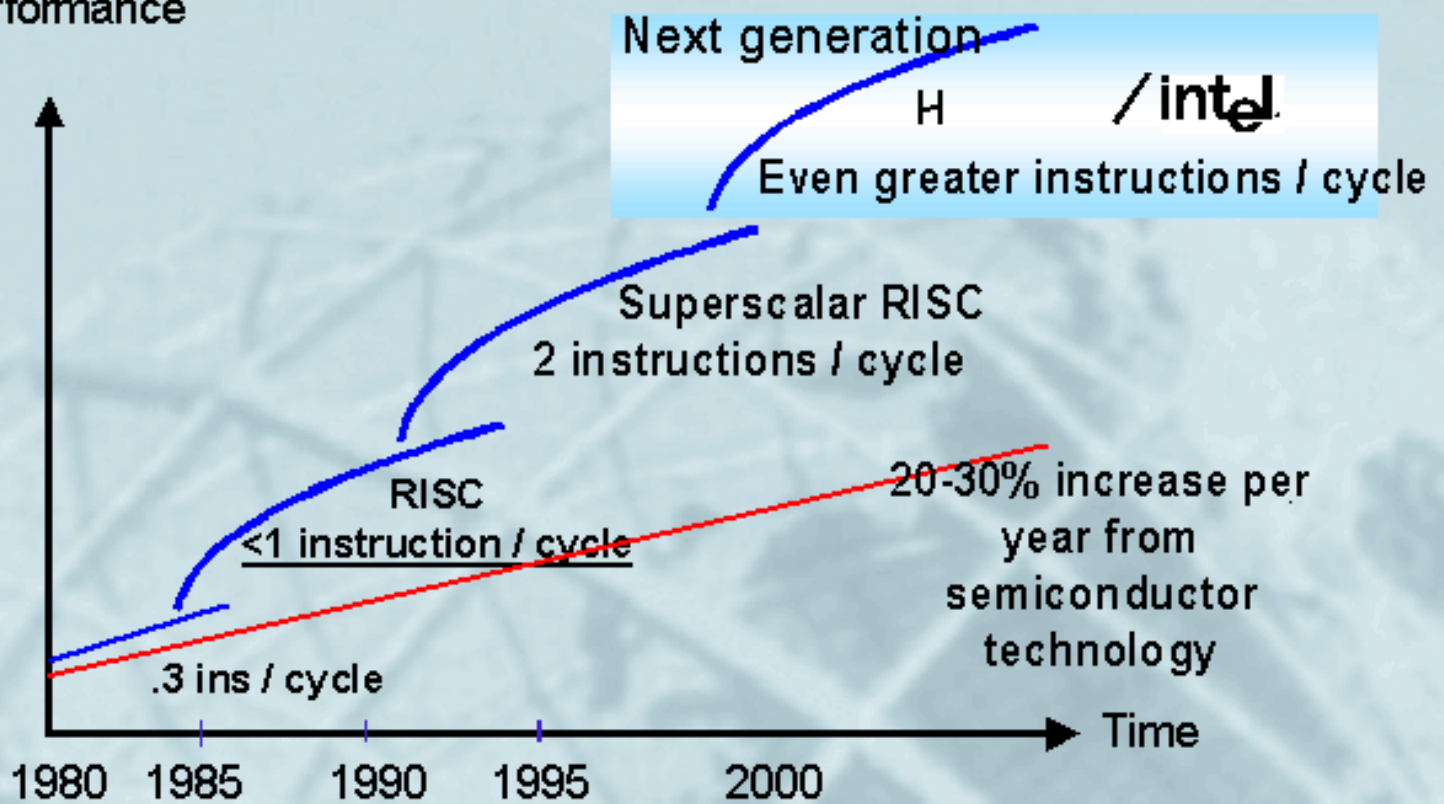


- ⌘ There is more to deploying an NFS fileserver than simply sizing (or plugging in) the server!
- ⌘ Multiple clients using a central fileserver involves designing a virtual computer, which must be balanced.
 - Fileserver throughput and response time
 - LAN bandwidth and latency
 - Client throughput



The NFS Client Drives the Network

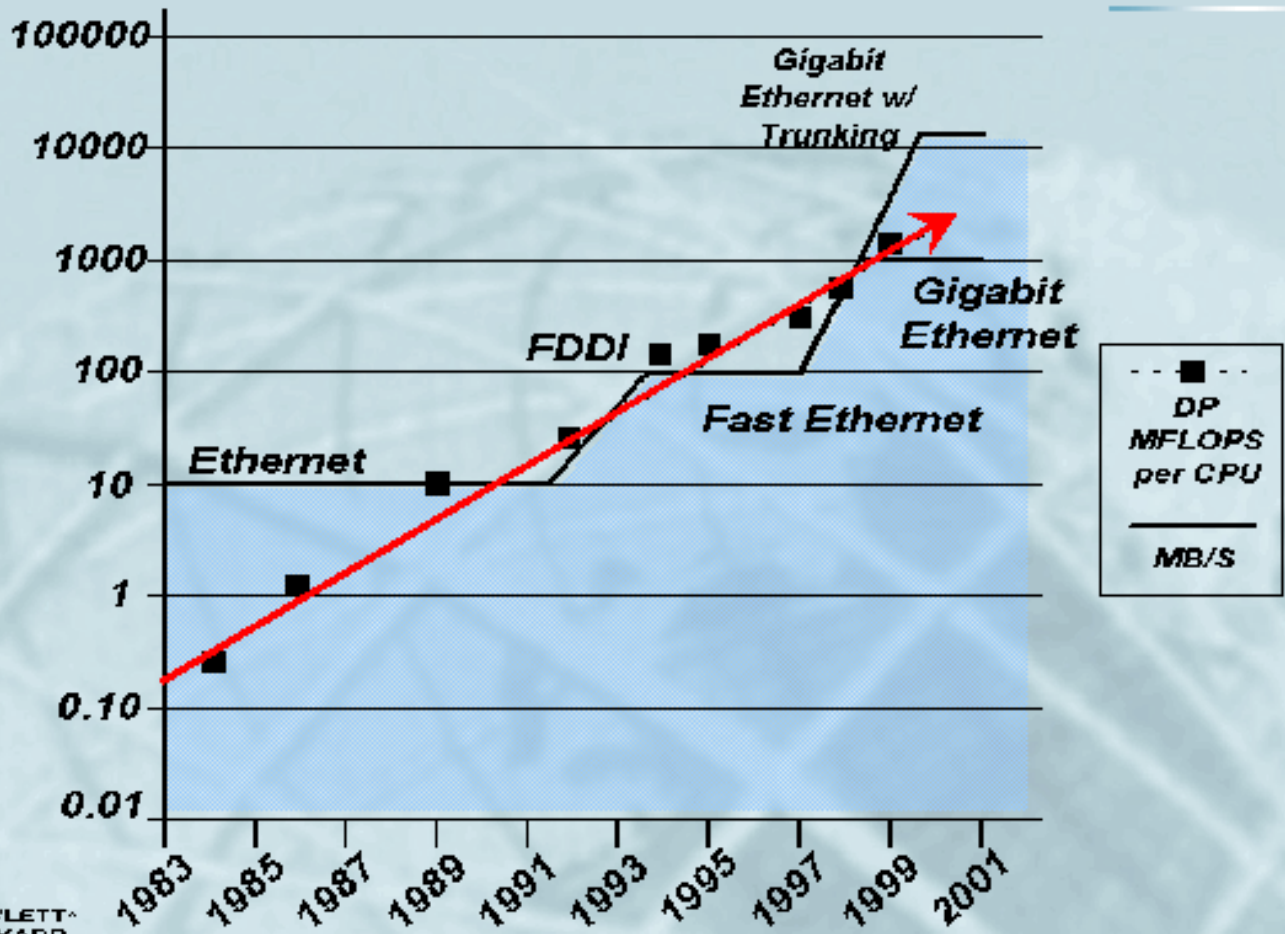
Performance



4 99/09/20 Rob Lucke



Where is the Network Today?



HP HEWLETT-PACKARD

5 99/09/20 Rob Lucke



NFS Design Approach



- ⌘ Determine client NFS IOP needs
- ⌘ Size the NFS server resource(s)
- ⌘ Size the network infrastructure
- ⌘ Tune the clients and the server
- ⌘ Measure performance



6 99/09/20 Rob Lucke



Seminar Agenda

- ⌘ Design Issues
- ⌘ **NFS Fileserver Benchmark Data**
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



7 99/09/20 Rob Lucke



NFS Fileserver Benchmark Data

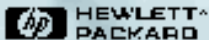
NFS server performance is specified by the **SPEC** (System Performance Evaluation Committee) benchmark(s).

The specific benchmark is **System FileServer**.

- Nhfsstone
- LADDIS is a re-implementation of Nhfsstone
- SPEC_SFS_1 / SPECsfs93
 - ◆ NFS PV2, UDP, specific operation mix
- SPEC_SFS_2 / SPEC SFS97
 - ◆ NFS PV2/PV3, TCP or UDP, different operation mix



Details may be found at <http://WWW.SPEC.ORG>



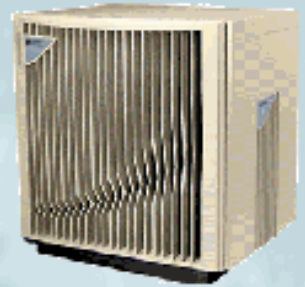
8 99/09/20 Rob Lucke



NFS Fileserver Benchmark Data

(continued)

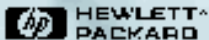
⌘ The **SPEC SFS** benchmarks measure fileserver performance in terms of NFS I/O operations per second (IOPS) and response time.



⌘ The server is subjected to larger and larger numbers of requests until the fileserver cannot keep up, causing throughput to decrease or response times to become unacceptable. (> 50 milliseconds)



⌘ Reporting **SPEC** numbers entails also reporting hardware and software configuration.



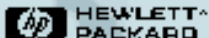
9 99/09/20 Rob Lucke



NFS Fileserver Benchmark Data

SPEC_SFS_1 operation mixture

<i>Operation Name</i>	<i>SFS Mix</i>	<i>Operation Description</i>
Null	0%	No operation
Getattr	13%	Get file attributes
Setattr	1%	Set file attributes
Root	0%	Find the root of a file system
Lookup	34%	Get a file handle
Readlink	8%	Get the value of a symbolic link
Read	22%	Read a file
Wrcache	0%	Not used
Write	15%	Write to a file
Create	2%	Create a new file
Remove	1%	Delete a file
Rename	0%	Rename a file
Link	0%	Create a new hard link to a file
Symlink	0%	Create a new symbolic link
Mkdir	0%	Create a new directory
Rmdir	0%	Remove a directory
Readdir	3%	Read a directory entry
Statfs	1%	Get file system statistics



10 99/09/20 Rob Lucke

Slide 10 of 188



SPEC SFS 1 Summary (Slide 1 of 2)

097.LADDIS

Hewlett-Packard Company

HP Series 9000/V2250 Enterprise Server

Server Hardware

Model No. 9000J
 CPU 2x0 V86C30C
 # of CPUs 16
 Primary Cache 256K-256K/16K
 Secondary Cache None
 Other Cache None
 Memory 15685
 Factory None
 User Configurable (by) None

CPU Subsystem & Firmware
 # of Configurations 5
 Num Jibs 0/2
 Num Disks 0/2
 Num Side Modules 183

Network Subsystem
 (Port 22: dynamic (on switch))
 Num Net Cards 10
 Num Networks 10
 Num Servers 4-101

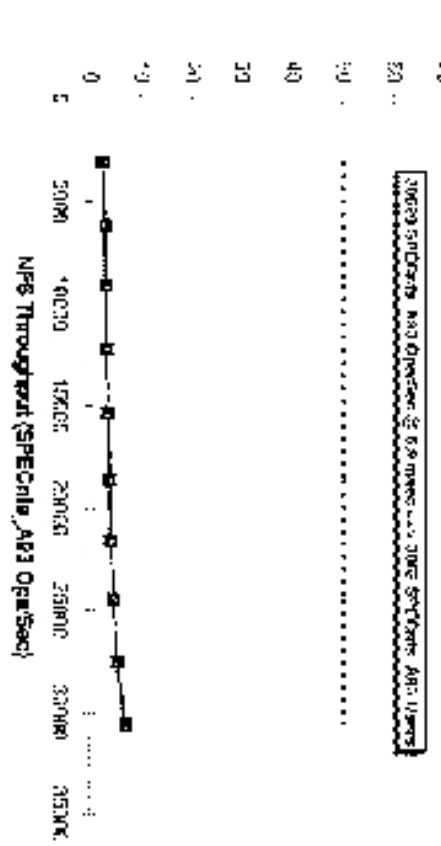
OS Vendor HP
 Other Software None
 File System UFS

Server Tuning
 (Admin Name: TJA (TJA))
 Admin User Name TJA
 Admin User Password 273
 Admin User Date 09/19/99
 Admin User 09/19/99

Number of Load Generators 80
 Transactions Per Load Generator 6
 Bits per Second 2
 Disk I/Os per Second 2

System Config	Hardware Availability	System Availability	Uptime
Specimen 1	100%	100%	100%
Specimen 2	100%	100%	100%
Specimen 3	100%	100%	100%
Specimen 4	100%	100%	100%
Specimen 5	100%	100%	100%
Specimen 6	100%	100%	100%
Specimen 7	100%	100%	100%
Specimen 8	100%	100%	100%
Specimen 9	100%	100%	100%
Specimen 10	100%	100%	100%
Specimen 11	100%	100%	100%
Specimen 12	100%	100%	100%
Specimen 13	100%	100%	100%
Specimen 14	100%	100%	100%
Specimen 15	100%	100%	100%
Specimen 16	100%	100%	100%
Specimen 17	100%	100%	100%
Specimen 18	100%	100%	100%
Specimen 19	100%	100%	100%
Specimen 20	100%	100%	100%

Average NFS Response Time (msec)



Report Summary of Tuning Parameters (Continued on Slide 2 of 2)
 Scenario: SFS-1X-Server
 Scenario: SFS-1X-Server

File: 097.LADDIS
 Run Date: 09/19/99
 Run Time: 10:00:00
 Run User: root



SPEC SFS 1 Summary (Side 2 of 2)

097.LADDIS

**Hewlett-Packard Company
HP Series 9000/V2250 Enterprise Server**

Default Server Description

Disk Subsystem & Filesystems

FCI 104: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 105: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 106: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 107: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 108: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 109: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 110: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 111: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 112: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 113: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 114: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 115: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 116: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 117: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 118: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 119: Adapter for I/OA (FCI) : 16 4000 Controller
 FC1 120: Adapter for I/OA (FCI) : 16 4000 Controller

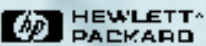
Default Local Generator List Description

Model	Quantity	Description
45 2000-05	1	16 4000 Controller
45 2100	1	16 4000 Controller

Method Configuration

Local Gen. #	Local Gen. Type	Network	Target File Systems	Mount
01	01	Y	FCI 104: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 104: Adapter for I/OA (FCI) : 16 4000 Controller
02	02	Y	FCI 105: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 105: Adapter for I/OA (FCI) : 16 4000 Controller
03	03	Y	FCI 106: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 106: Adapter for I/OA (FCI) : 16 4000 Controller
04	04	Y	FCI 107: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 107: Adapter for I/OA (FCI) : 16 4000 Controller
05	05	Y	FCI 108: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 108: Adapter for I/OA (FCI) : 16 4000 Controller
06	06	Y	FCI 109: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 109: Adapter for I/OA (FCI) : 16 4000 Controller
07	07	Y	FCI 110: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 110: Adapter for I/OA (FCI) : 16 4000 Controller
08	08	Y	FCI 111: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 111: Adapter for I/OA (FCI) : 16 4000 Controller
09	09	Y	FCI 112: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 112: Adapter for I/OA (FCI) : 16 4000 Controller
10	10	Y	FCI 113: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 113: Adapter for I/OA (FCI) : 16 4000 Controller
11	11	Y	FCI 114: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 114: Adapter for I/OA (FCI) : 16 4000 Controller
12	12	Y	FCI 115: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 115: Adapter for I/OA (FCI) : 16 4000 Controller
13	13	Y	FCI 116: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 116: Adapter for I/OA (FCI) : 16 4000 Controller
14	14	Y	FCI 117: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 117: Adapter for I/OA (FCI) : 16 4000 Controller
15	15	Y	FCI 118: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 118: Adapter for I/OA (FCI) : 16 4000 Controller
16	16	Y	FCI 119: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 119: Adapter for I/OA (FCI) : 16 4000 Controller
17	17	Y	FCI 120: Adapter for I/OA (FCI) : 16 4000 Controller	FCI 120: Adapter for I/OA (FCI) : 16 4000 Controller

Method Summary of Tuning Parameters: (Expand to view details)



12 99/09/20 Rob Lucke



NFS Fileserver Benchmark Data

SPEC_SFS_2 Information

⊗ **SPEC_SFS_2 differs from SPEC_SFS1 in several ways:**

- Benchmark workload is based on a survey of over 1000 servers in different environments and is more realistic
- Supports NFS protocol version 2 (PV2) and protocol version 3 (PV3) and separate reporting
 - ◆ SPECsfs97.v2/SPECsfs97.v3
- Supports both UDP and TCP transport mechanisms

⊗ **We do not yet know how this benchmark affects some of the things we will discuss. Be careful!**









13 99/09/20 Rob Lucke

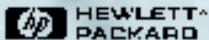


NFS Fileserver Benchmark Data

*Typical NFS IOPS requirements***

 PC/Office Automation 	1-3 NFS IOPS per client per second
Light Unix Workstation (interactive) 	3-8 NFS IOPS per client per second
Intense Unix Workstation (CASE)	10-20 NFS IOPS per client per second 
Extremely Heavy Unix Workstation (Simulation)	25-40 NFS IOPS per client per second
 Pure/Atria Clearcase (Configuration Mgmt.) 	125 NFS IOPS per client per second (!)

***Typical numbers only. There is no substitute for actual measurements of client NFS IOPS.*



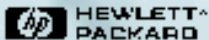
14 99/09/20 Rob Lucke



NFS Fileserver Benchmark Data

The NFS IOP "Magic" Number, 4KB

- ⌘ **SPEC_SFS_1** defines 1000 NFS IOPS as 1000 NFS RPC calls of the previously noted mixture.
- ⌘ NFS read and writes are 8192 (8KB). All other operations are approximately 192 bytes.
- ⌘ With the defined mixture, the average NFS I/O in the **SPEC_SFS_1** benchmark is approximately 3.2KB.
- ⌘ The 3.2KB is rounded to 4KB/NFS IOP to account for things like packet headers, variable length data, etc.



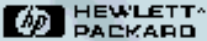
15 99/09/20 Rob Lucke

Slide 15 of 188



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements**
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



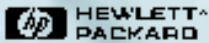
16 99/09/20 Rob Lucke



Measuring Client NFS Activity

GPM Global NFS Activity Screen

GPM NFS - NFS Global Activity				
File Reports				Help
System: hpupora		Last Update: 15:01:37		Int: 10 sec
Activity	Server (Inbound)		Client (Outbound)	
	Current	Cum	Current	Cum
Read Rate	1.4	1.5	0.0	0.0
Write Rate	0.0	0.3	0.0	0.0
Read Byte Rate	0.0	0.0	0.0	0.0
Write Byte Rate	0.0	0.0	0.0	0.0
NFS IOs	14	2650	0	0
NFS Calls	0	576	0	576
Bad Calls	0	0	0	0
Service Time	0.00	3.17	0.00	0.00
Network Time			0.00	0.00
Read/Write Qlen			0.00	0.00
Idle bids			0	



17 99/09/20 Rob Lucke

Source: GPM->Reports->Network Info->NFS Global Activity

Slide 17 of 188



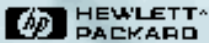
Measuring Client NFS Activity

GPM Client NFS PV2 & PV3 Operations

Client: hqiguna Last Update: 15:02:17 Int: 11 sec

Operations: All 40 Selected
Number of Systems: 1

Operation Name	Svr. Op	Svr. Op Cum	Clnt. Op	Clnt. Op Cum
v2_mall	0	0	0	0
v2_getattr	0	0	0	0
v2_setattr	0	0	0	0
v2_root	0	0	0	0
v2_lookup	0	0	0	0
v2_readlink	0	0	0	0
v2_read	0	0	0	0
v2_wrtleacu	0	0	0	0
v2_write	0	0	0	0
v2_create	0	0	0	0
v2_remove	0	0	0	0
v2_rename	0	0	0	0
v2_link	0	0	0	0
v2_symlink	0	0	0	0
v2_mkdir	0	0	0	0
v2_rmdir	0	0	0	0
v2_readdir	0	0	0	0
v2_statfs	0	0	0	0



18 99/09/20 Rob Lucke

Source: GPM->Reports->Network Info->NFS By Operation

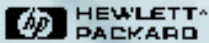


Measuring Client NFS Activity

GPM Client NFS PV2 & PV3 Operations



Operation Name	Svr Op	Svr Op Cum	Clnt Op	Clnt Op Cum
v3_null	0	0	0	0
v0_getattr	15	1661	0	0
v0_setattr	1	21	0	0
v3_lookup	1	251	0	0
v0_access	0	0	0	0
v0_readlink	0	0	0	0
v3_read	2	367	0	0
v3_write	1	783	0	0
v0_create	0	7	0	0
v3_mkdir	0	0	0	0
v3_symlink	0	0	0	0
v0_mknod	0	0	0	0
v3_remove	0	1	0	0
v3_rmdir	0	0	0	0
v0_rename	0	4	0	0
v3_link	0	0	0	0
v3_readdir	0	18	0	0
v0_readdirp	0	0	0	0
v3_lstat	0	88	0	0
v3_statfs	0	88	0	0
v0_pathconf	0	0	0	0
v3_commil	0	37	0	0



19 99/09/20 Rob Lucke

Source GPM>Reports>Network Info>NFS By Operation



Measuring Client NFS Activity

NFS Activity with the nfsstat command

```

root@hp:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       10G   2.1G   7.9G  21% /
/dev/sda5       10G   2.1G   7.9G  21% /home

root@hp:~# nfsstat -c
Client root:
calls      badcalls  retrans   badretd   timeout   werr     reworded
133615    439      140       0         23.2      0        0

Client nfs:
calls      badcalls  noxact    noisleep
133615    7        1524120   0

Version 2 : (0 calls)
null      getattr  setattr  root      lockup    readlink  read
0 0%      0 0%      0 0%      0 0%      0 0%      0 0%      0 0%
write    write    create    remove    rename    link      symlink
0 0%      0 0%      0 0%      0 0%      0 0%      0 0%      0 0%
mkdir    mkdir    readdir  statfs
0 0%      0 0%      0 0%      0 0%

Version 3 : (133615 calls)
null      getattr  setattr  linkup    statfs    readlink  read
0 0%      475714 36% 7665 0%  7253 1%  0 0%      2634 0%  26314 21%
write    create    mkdir    symlink    mkdir    remove    mkdir
73046 5%  1053 0%  12 0%    26 0%    0 0%      1049 0%  7 0%
rename  link      readdir  readdir   factor    getattr  pathconf
19 0%    6 0%    1267 0%  0 0%      147973 11% 147481 11%  0 0%
commit
33219 1%

```





Measuring Client NFS Activity

NFS Activity with the nfsstat command

```


root@hupiter: /root/home
# nfsstat -s
Server rpc
calls      badcalls  nullrecv  badlen    xdrcall   nfedran
1712       0         /         0         0         1712

Server nls:
calls      badcalls
1708       0

Version 3 : (1708 calls)
null      getattr  setattr  lockup    access    readlink  read
2 0%     / 0%     0 0%     0 0%     0 0%     0 0%     166 / 97%
wrtache  write    create   remove    rename    link      symlink
0 0%     0 0%     0 0%     0 0%     0 0%     0 0%     0 0%
mkdir    rmdir   readdir  statfs
0 0%     0 0%     0 0%     14 0%

Version 3 : (0 calls)
null      getattr  setattr  lockup    access    readlink  read
0 0%     0 0%     0 0%     0 0%     0 0%     0 0%
write    create   mkdir    symlink    mknod     remove    rmdir
0 0%     0 0%     0 0%     0 0%     0 0%     0 0%
rename   link     readdir  readdir   isstat    isinfo    pathconf
0 0%     0 0%     0 0%     0 0%     0 0%     0 0%
commit
0 0%
  
```



 HEWLETT-
PACKARD “nfsstat -s” gives accumulated server NFS RPCs and mix
21 99/09/20 Rob Lucke



Measuring Client NFS Activity

NFS IOPS (NFS I/Os per second = RPC calls per second)

⌘ Rebooting or executing “nfsstat -z” as root will zero out the NFS counters used by “nfsstat”.

⌘ Method 1: Monday morning zero the counters, then Friday evening total the counters with the “nfsstat -c”. Divide to obtain the average.

⌘ Method 2: Monday morning zero the counters, then at set time intervals measure the client activity with “nfsstat -c”.

⌘ Method 3: Determine a standard mix of user activities, like “load drawing”, and measure IOPS for each operation.





```
#!/sbin/sh
###
## This script will log the number of NFS RPC calls
## made during the interval tracked by the value in
## ${1}. The default interval is set to 10 seconds
## if no value is passed in ${1}. The lowest possible
## interval is one second, which should be small enough.
###
    INTERVAL=10
    NFSSTAT=/usr/bin/nfsstat
    LOG_OPTIONS="-rc"
    RESET_OPTIONS="-z"
    LOGFILE=/tmp/NFS_IOPS.log
    HOSTNAME=$(hostname)
    HOSTSTAMP="${HOSTNAME} ${INTERVAL} Seconds"

if [[ -n ${1} ]]; then
    INTERVAL=${1}
fi

${NFSSTAT} ${RESET_OPTIONS} > /dev/null

###
## Save the old log file, just in case we really want
## the data that is already there.
###
23 99/09/20 Rob Lucke
```



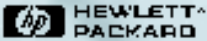
```

if [[ -f ${LOGFILE} ]]; then
    mv ${LOGFILE} ${LOGFILE}.OLD
fi

exec 9>${LOGFILE}
print -u9 "${HOSTSTAMP}"
print -u9 "          IOPS      retrans      timeout"

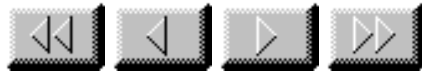
while true
do
    TIMESTAMP=$(date '+%b %d %H:%M:%S' )
    #
    #      Print out the values for "calls", "retrans", and "timeout" from
    #      the client RPC information
    #
    INFO=$( ${NFSSTAT} ${LOG_OPTIONS} | \
        awk '{if (NR==4)
{printf("%s\t\t%s\t\t%s", $1,$3,$5)}}')
    ${NFSSTAT} ${RESET_OPTIONS} > /dev/null
    print -u9 "${TIMESTAMP} = ${INFO}"
    sleep ${INTERVAL}
done
exit 0
#####
#####

```





Slide 25 of 188



NFS Fileserver Sizing

Areas of Primary Concern

☿ CPU subsystem

- SPEC_SFS_1 rating
- CPU and I/O bus capacities
- CPU growth path



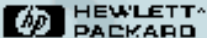
☿ LAN subsystem

- Numbers and types of LAN connections supported
- I/O Bus capacity
- Path to other LAN technologies



☿ Disk subsystem

- I/O Bus and channel capacities
- JBOD versus RAID
- LVM, HFS, and JFS (VxFS)



26 99/09/20 Rob Lucke



NFS Fileserver Sizing

Secondary Areas of Concern

High Availability

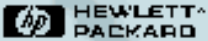
- CPU failover
- Disk failover
- LAN failover
- Power failover



Performance Tools



Service and Diagnostics



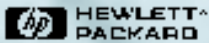
27 99/09/20 Rob Lucke



NFS Fileserver Sizing

CPU Selection: SPEC_SFS_1 Ratings

HP System	SPEC SFS 1		Data Flow	Network I/Fs		
	NFS IOP Rating	Response Time (ms)	NFS MB/S (IOPS*4KB)	10 Mb/S	100 Mb/S	1000 Mb/S
D370-1	2390		5.45	4	1	1
K460-1	3700		14.45	12	2	1
D370-2	3945	23.80	14.84	12	2	1
V2250-2	5192	8.60	20.28	17	2	1
K460-2	6400		25.00	20	2	1
L2000-2	9133	3.60	37.41	30	4	1
K460-4	9572	19.90	37.39	30	3	1
V2250-4	9789	6.60	38.24	31	4	1
K570-6	10400		40.63	33	4	1
N4360-2	10724	3.09	43.93	35	4	1
J5000 (2)	11500		47.10	38	4	1
V2250-8	16442	9.70	64.23	52	6	1
V2250-12	24425	9.40	95.41	77	8	1
V2250-16	30620	6.90	125.42	100	10	1
V2500-32	50000		204.80	164	17	2



28 99/09/20 Rob Lucke

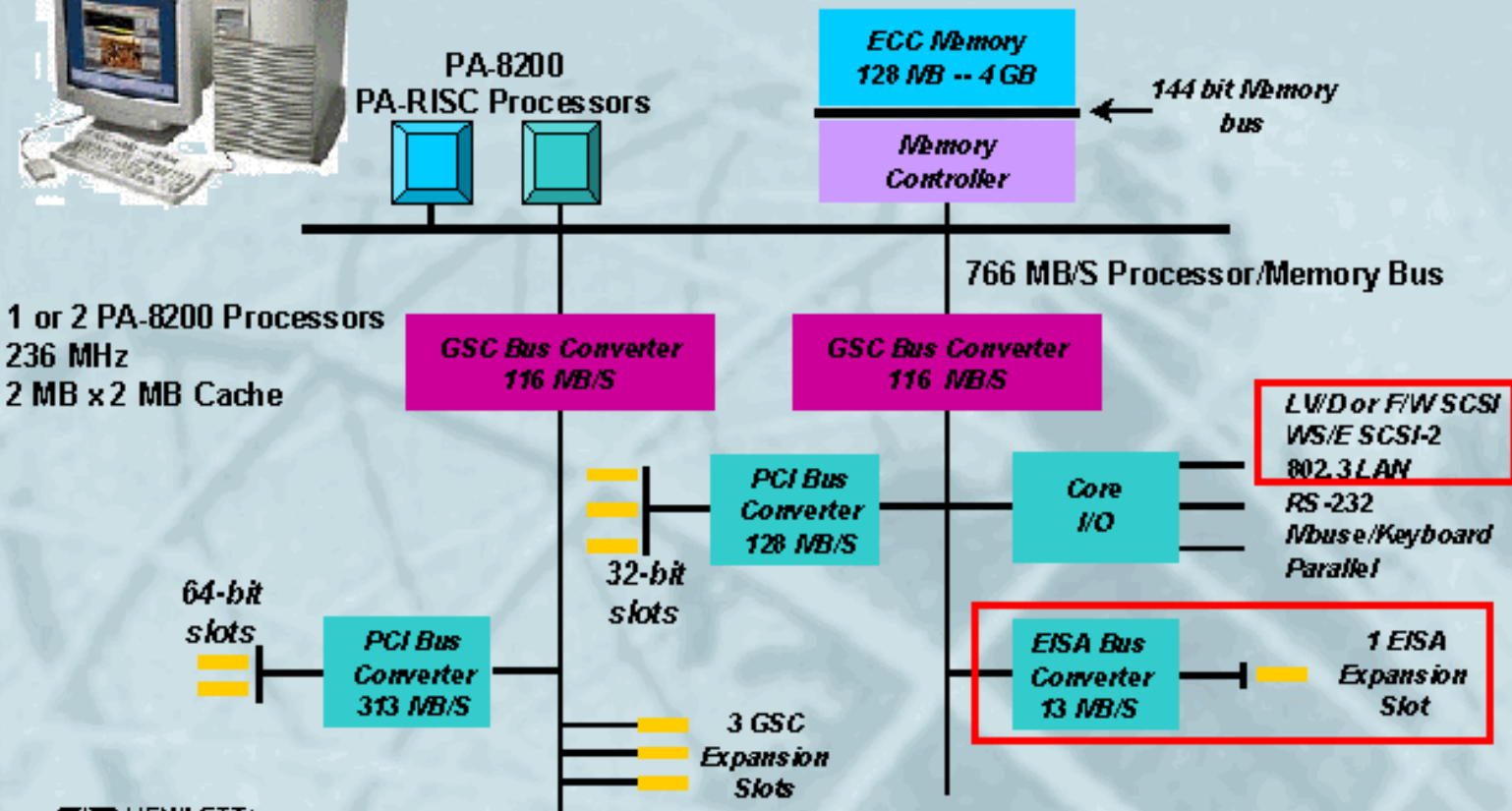
Numbers in **blue** are estimated.
Reported SPEC data as of 10/99.



NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

J2240 System Architecture



HP HEWLETT-PACKARD
 29 99/09/20 Rob Lucke



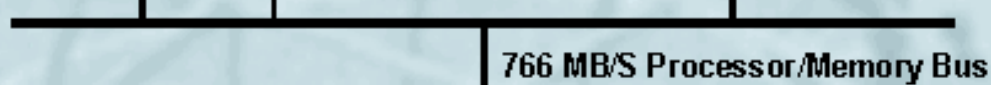
NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

D370/80 System Architecture

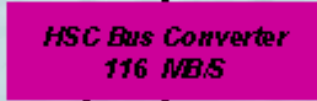


PA-8000
PA-RISC Processors



1 or 2 PA-8000 Processors

160 MHz	180 MHz
512x512	1024x1024 KB Cache
D370	D380



D370 Turbo Slot

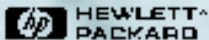
5 HSC
Expansion
Slots



S/E SCSI-2
802.3 LAN
RS-232
Nbus/Keyboard
Parallel



7 EISA
Expansion
Slots



30 99/09/20 Rob Lucke

Source: HP 9000 Enterprise Servers Configuration Guide 5/98



NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

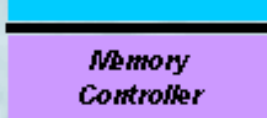
D370/80 System Architecture



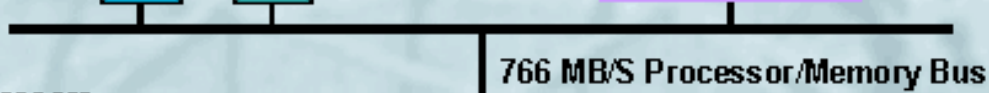
PA-8000
PA-RISC Processors



ECC Memory
64 MB -- 3 GB



144 bit Memory bus



1 or 2 PA-8000 Processors

160 MHz 180 MHz

512x512 1024x1024 KB Cache

D370 D380

HSC Bus Converter
116 MB/S

D370 Turbo Slot

5 HSC
Expansion
Slots

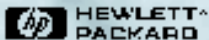
Core
I/O

S/E SCSI-2
802.3 LAN

RS-232
Nbus/Keyboard
Parallel

EISA Bus
Converter
13 MB/S

7 EISA
Expansion
Slots



31 99/09/20 Rob Lucke

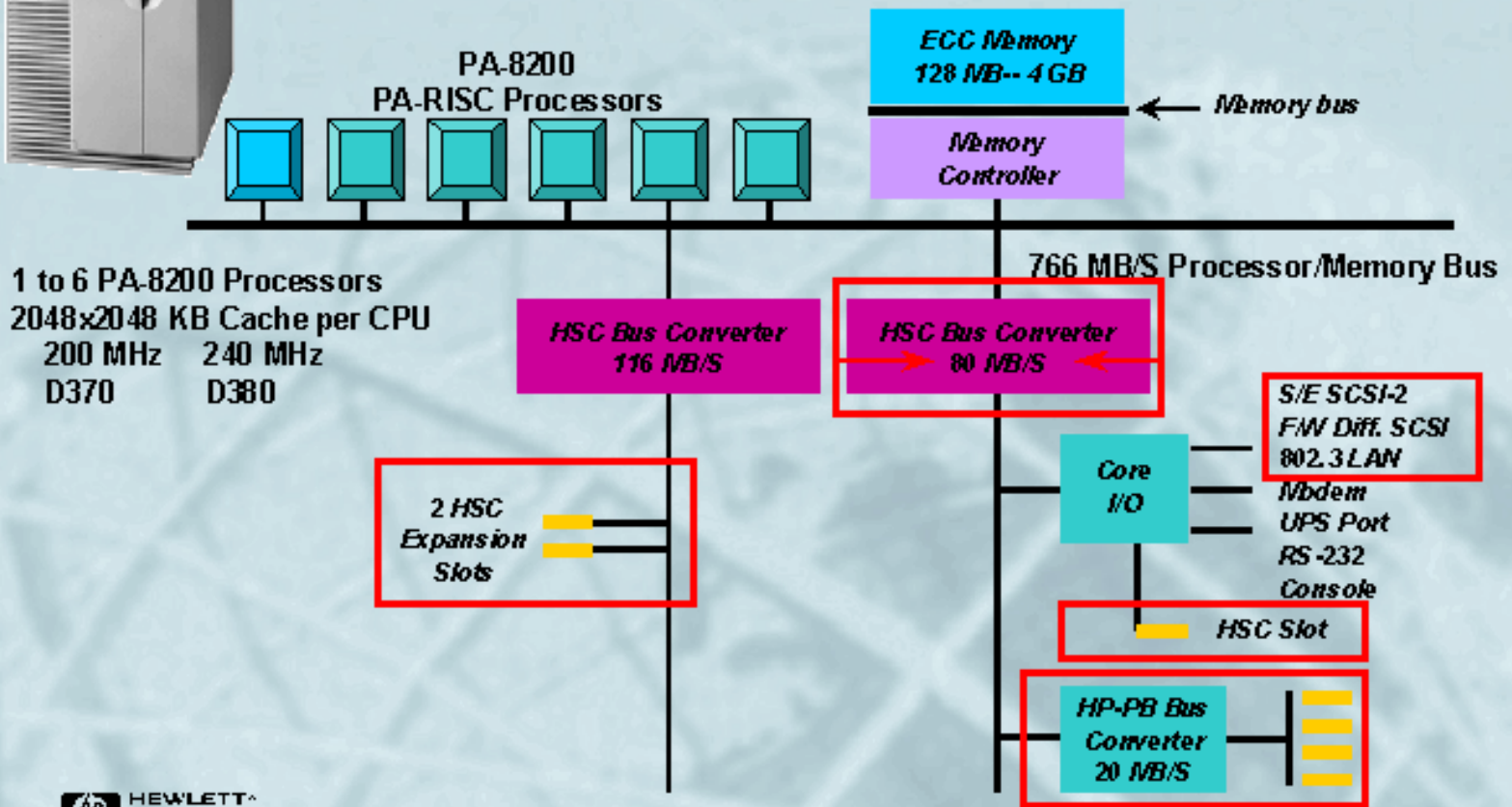
Source: HP 9000 Enterprise Servers Configuration Guide 5/98



NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

K370/80 System Architecture



32 99/09/20 Rob Lucke

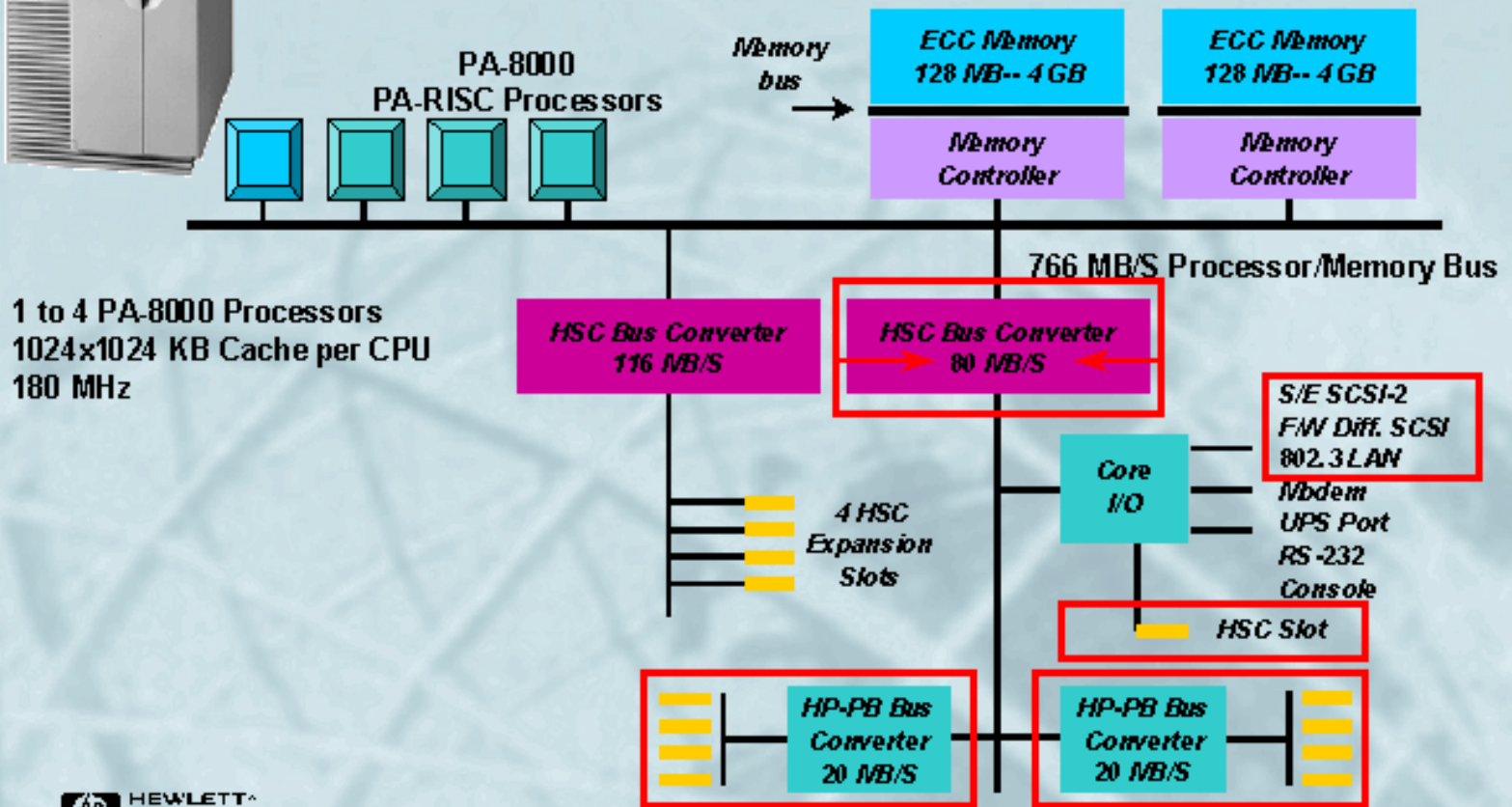
Source: HP 9000 Enterprise Servers Configuration Guide 5/98



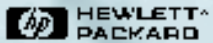
NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

K460/EG/XP System Architecture



1 to 4 PA-8000 Processors
1024x1024 KB Cache per CPU
180 MHz



33 99/09/20 Rob Lucke

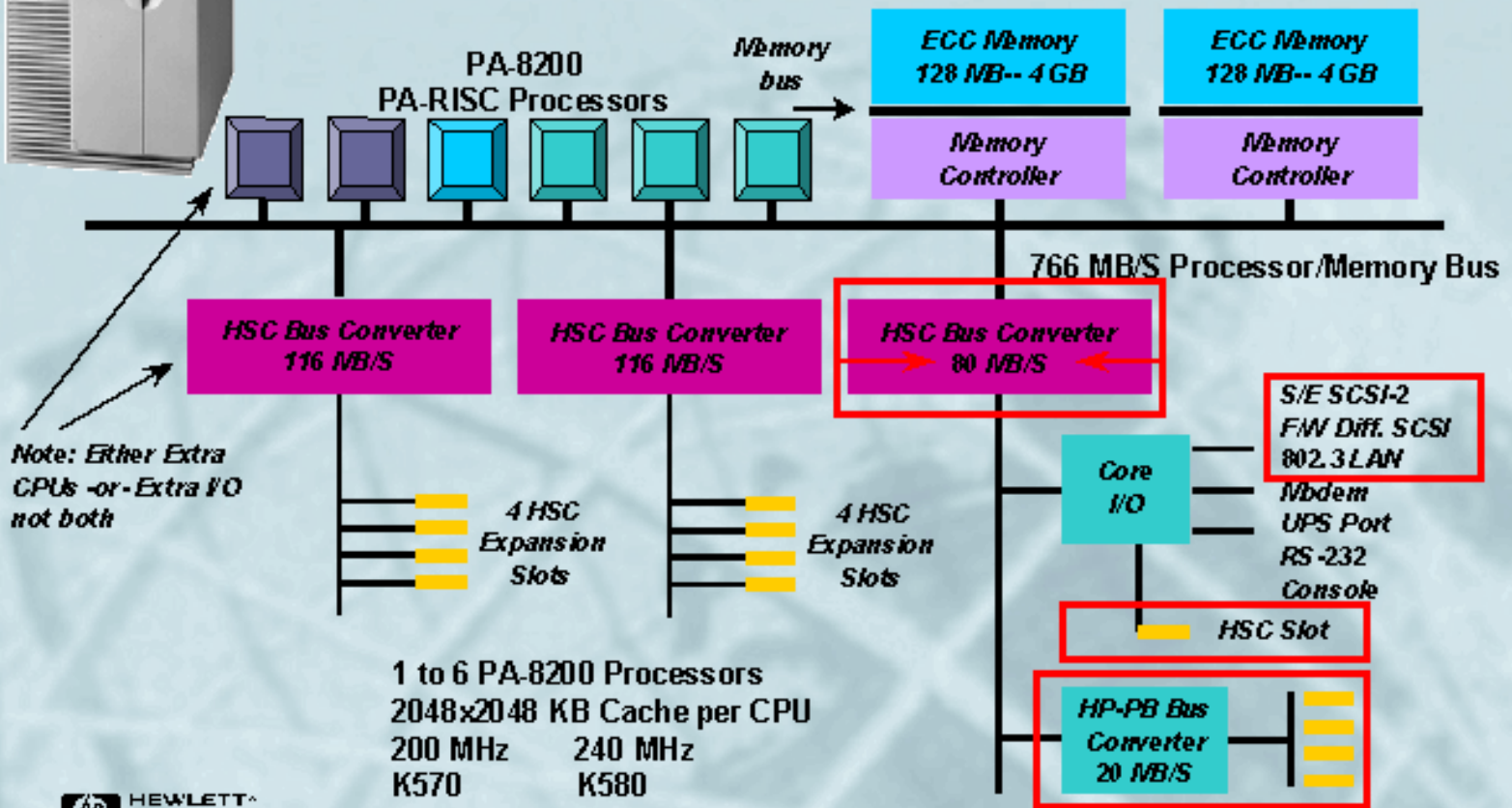
Source: HP 9000 Enterprise Servers Configuration Guide 5/98



NFS Fileserver Sizing

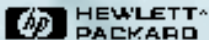
CPU Selection: System Bus Capabilities/Scaling

K570/80 System Architecture



Note: Either Extra CPUs -or- Extra I/O not both

1 to 6 PA-8200 Processors
 2048x2048 KB Cache per CPU
 200 MHz 240 MHz
 K570 K580



34 99/09/20 Rob Lucke

Source: *HP 9000 Enterprise Servers Configuration Guide 5/98*



NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

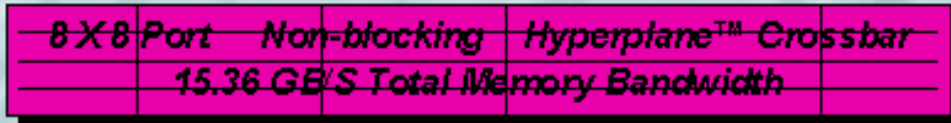
V2200/2250 System Architecture



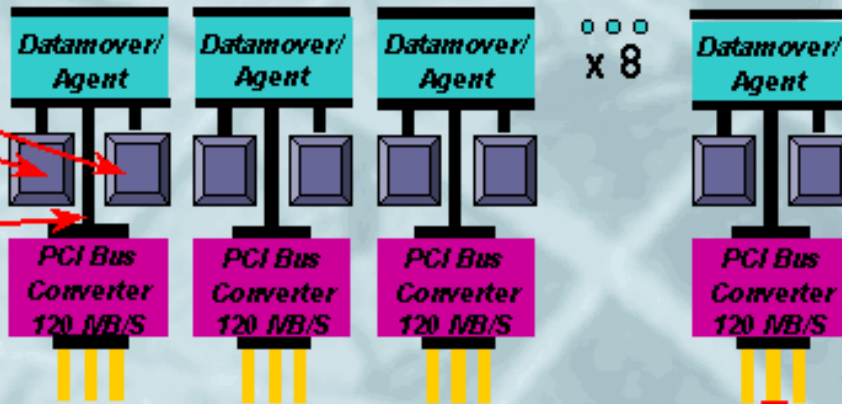
1 to 16 PA-8200 CPUs
200 MHz V2200
240 MHz V2250



980 MB/S
Each direction
per port

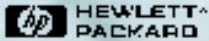


2 PA-8200
CPUs and Cache
240 MB/S
I/O Bus



980 MB/S
Each direction
per port

Up to 2 GB/S
Aggregate I/O
Bandwidth
(240 MB/S x 8)



35 99/09/20 Rob Lucke



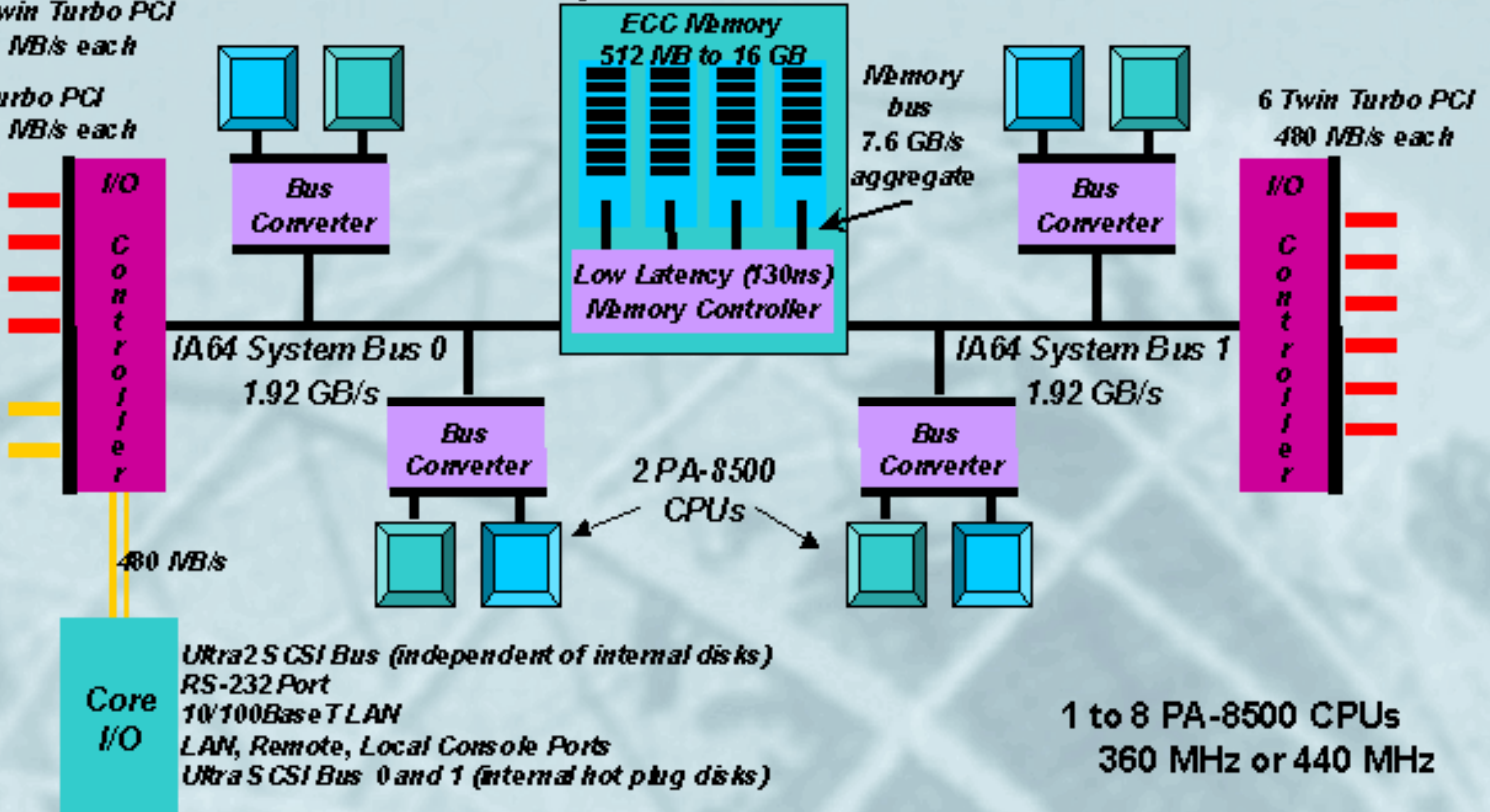
NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

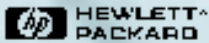
N4000 System Architecture

4 Twin Turbo PCI
480 MB/s each

2 Turbo PCI
240 MB/s each



1 to 8 PA-8500 CPUs
360 MHz or 440 MHz



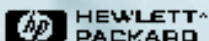
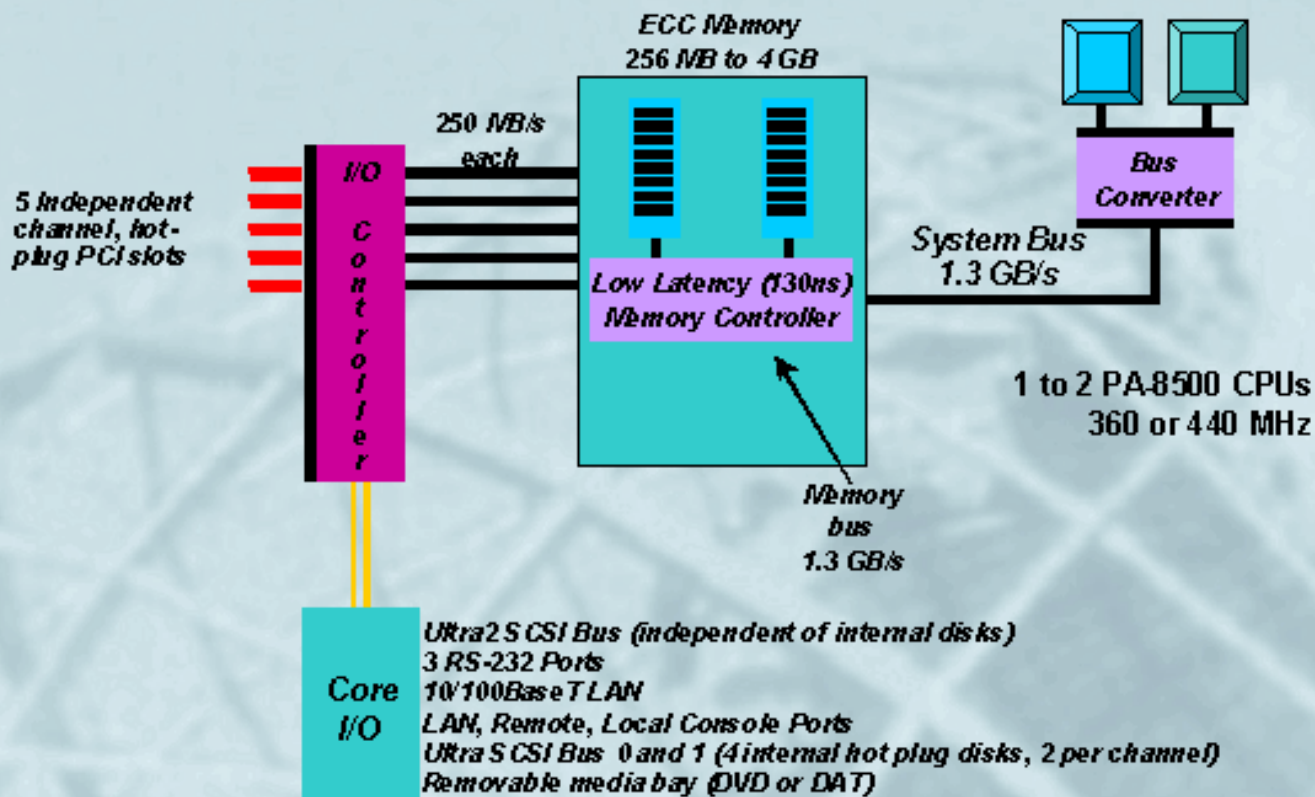
36 99/09/20 Rob Lucke



NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

L1000 System Architecture



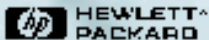
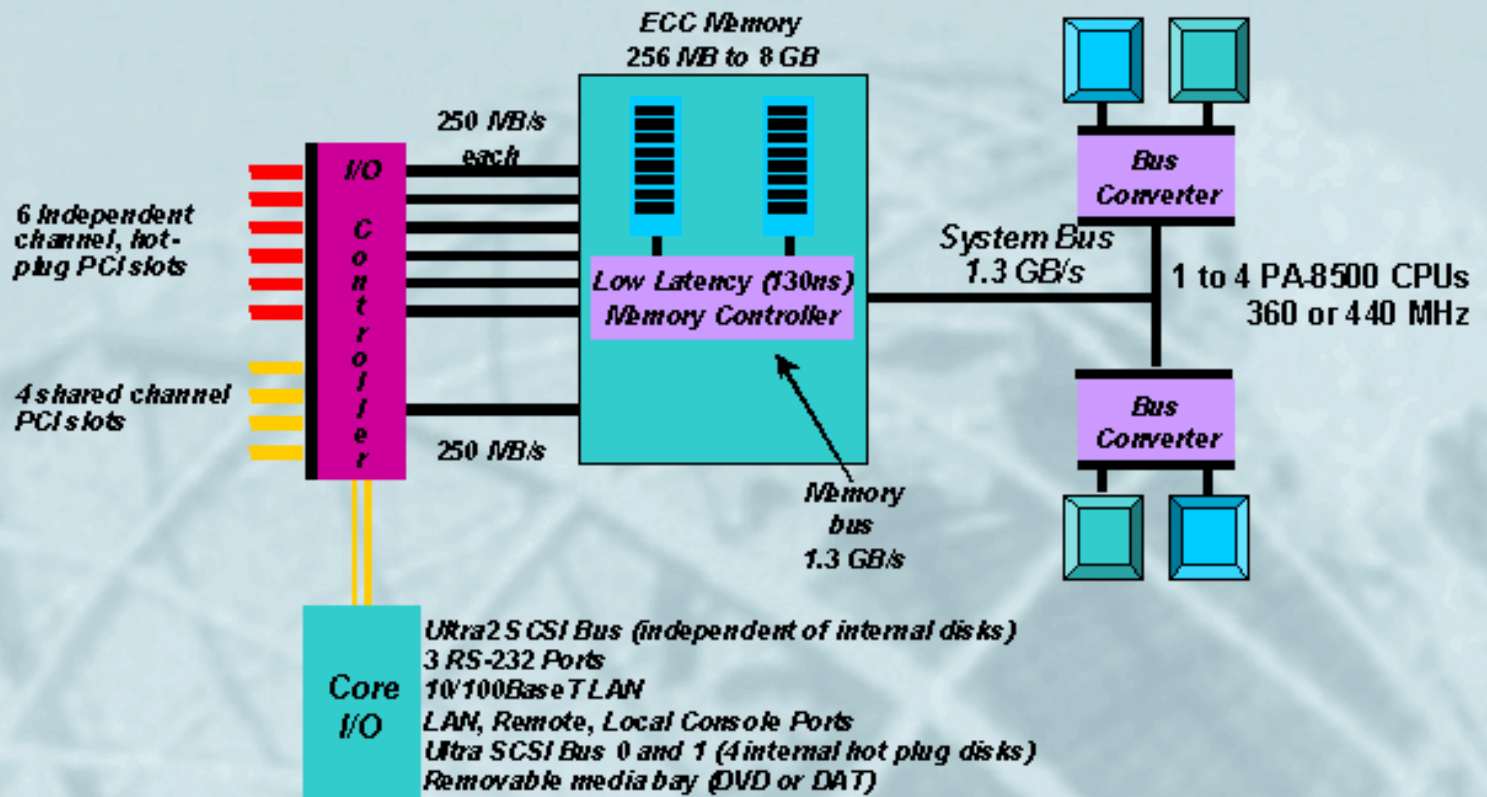
37 99/09/20 Rob Lucke



NFS Fileserver Sizing

CPU Selection: System Bus Capabilities/Scaling

L2000 System Architecture



38 99/09/20 Rob Lucke



NFS Fileserver Sizing

High Availability NFS

Increasing
Availability

- High Availability Features:**
- Dynamic page deallocation
 - JFS/LVM
 - Hot-Pluggable Disks
 - RAID Arrays
 - Mirrored Disks
 - Multiple I/O paths
 - Integrated UPS
 - MC/ServiceGuard



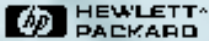
*Highly reliable systems
highly reliable hardware
high quality OS
predictive support
up time > 99.8%*

*Data protection
disk array*

*Data availability
mirrored arrays*

*System availability
mirrored arrays
redundant systems*

**System and Network Management Tools
24x7 Support Services
Disaster Recovery Services**



39 99/09/20 Rob Lucke



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



40 99/09/20 Rob Lucke

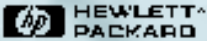


NFS Fileserver Sizing

Network Design

"There is no network from the user's point of view. There are only services. The goal of all network design and management is to preserve this misunderstanding."

INTEROP fall 1991
Scott Bradner
Harvard University



41 99/09/20 Rob Lucke

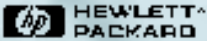
Slide 41 of 188



NFS Fileserver Sizing

Network Design: Network Technologies

	<i>Ethernet</i>	<i>Fast Ethernet</i>	<i>Gigabit Ethernet</i>	<i>ATM</i>
<i>Data Rate</i>	10 Mbps	100 Mbps	1000 Mbps	155/622/2488 Mbps OC3/OC12/OC48
<i>Cat. 5 UTP</i>	100 m	100 m	100 m	
<i>Shielded Copper</i>	500 m	100 m	25 m	
<i>Multi-mode Fiber</i>	2 Km	(HD) 412 m (FD) 2 Km	260-550 m	2 KM
<i>Single Mode Fiber</i>	25 Km	20 Km	3 Km	Up to 75 Km (unlimited w/Carrier)



42 99/09/20 Rob Lucke

Slide 42 of 188

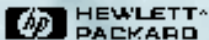


NFS Fileserver Sizing

Network Loading: Ethernet Collision Domains

⌘ Ethernet Segment Loading Rules for 10baseT

- 10baseT Ethernet = 10 Mb/S; 10 Mb/S = **1.25 MB/S**
(Maximum theoretical)
- 10baseT Ethernet = 1.25 MB/4KB per IOP = **300 NFS IOPS**
(Maximum theoretical Ethernet saturation)
- Studies show that interactive X-Windows clients suffer when Ethernet usage exceeds 25%
- Therefore: Don't plan to load Ethernet more than **25-30%**
(100 NFS IOPS)
- Headroom should be left for interactive traffic beyond the NFS traffic (e.g. telnet, X-Windows, ftp, etc.)
- Maximum NFS IOPS per Ethernet LAN = 100 IOPS
(This will still annoy people with poor interactive response)



43 99/09/20 Rob Lucke

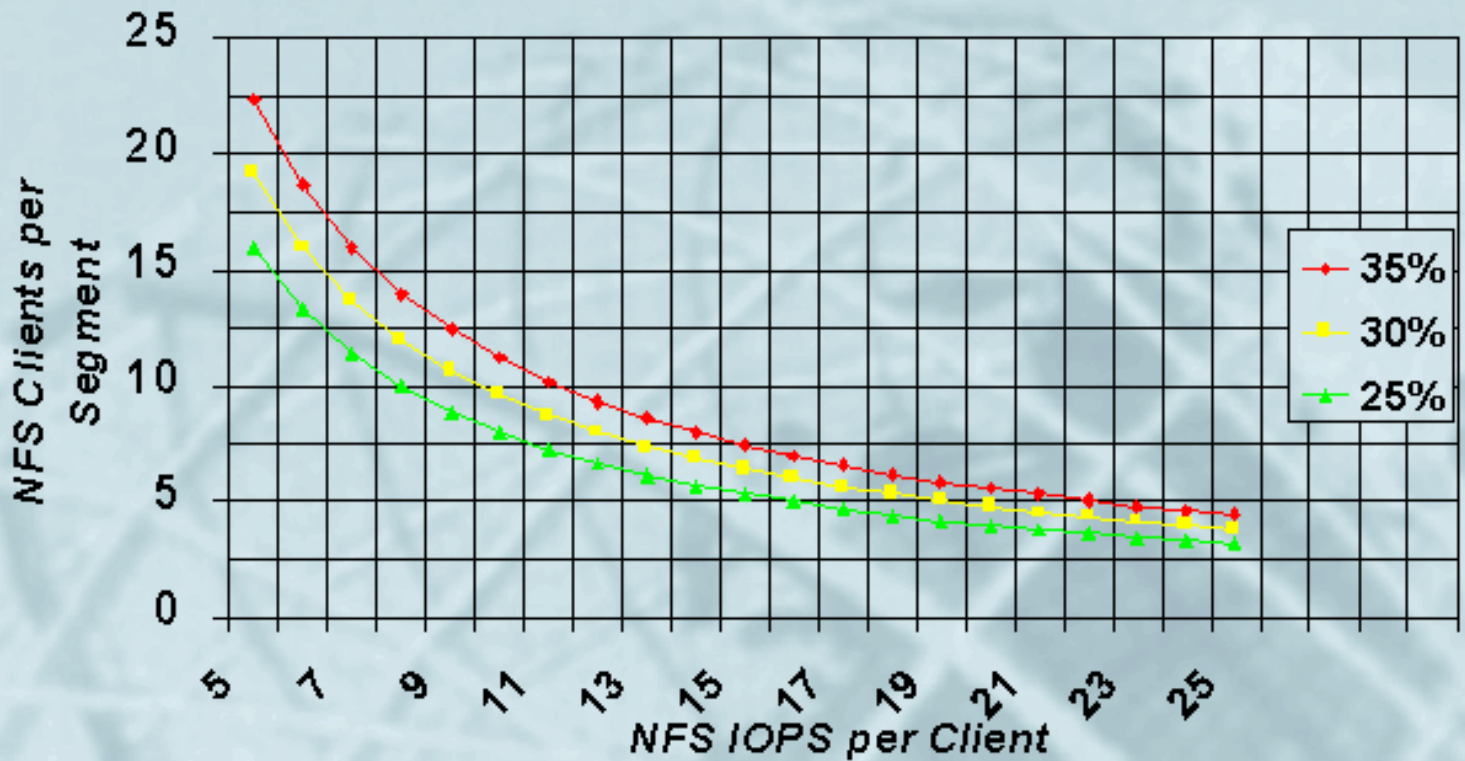
Slide 43 of 188



NFS Fileserver Sizing

Network Loading: 10baseT Ethernet Utilization Curves

NFS Clients per Shared 10baseT Segment



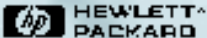


NFS Fileserver Sizing

Network Loading: FDDI Token Ring

⊗ FDDI Segment Loading Rules

- FDDI = 100 Mb/S; 100 Mb/S = **12.5 MB/S**
(Maximum theoretical)
- FDDI = 12.5 MB / 4 KB per NFS IOP = **3200 NFS IOPS**
(Maximum theoretical FDDI saturation)
- Studies show FDDI performance breaks down at 80% utilization
- Therefore, don't plan to load FDDI at more than **80%**
(2500 NFS IOPS)
- Leave 50% headroom for growth and interactive traffic
(e.g. telnet, X-windows, ftp, etc.)
- Maximum Operating NFS IOPS per FDDI = 1200 IOPS
(about 12 10baseT ethernets)



45 99/09/20 Rob Lucke

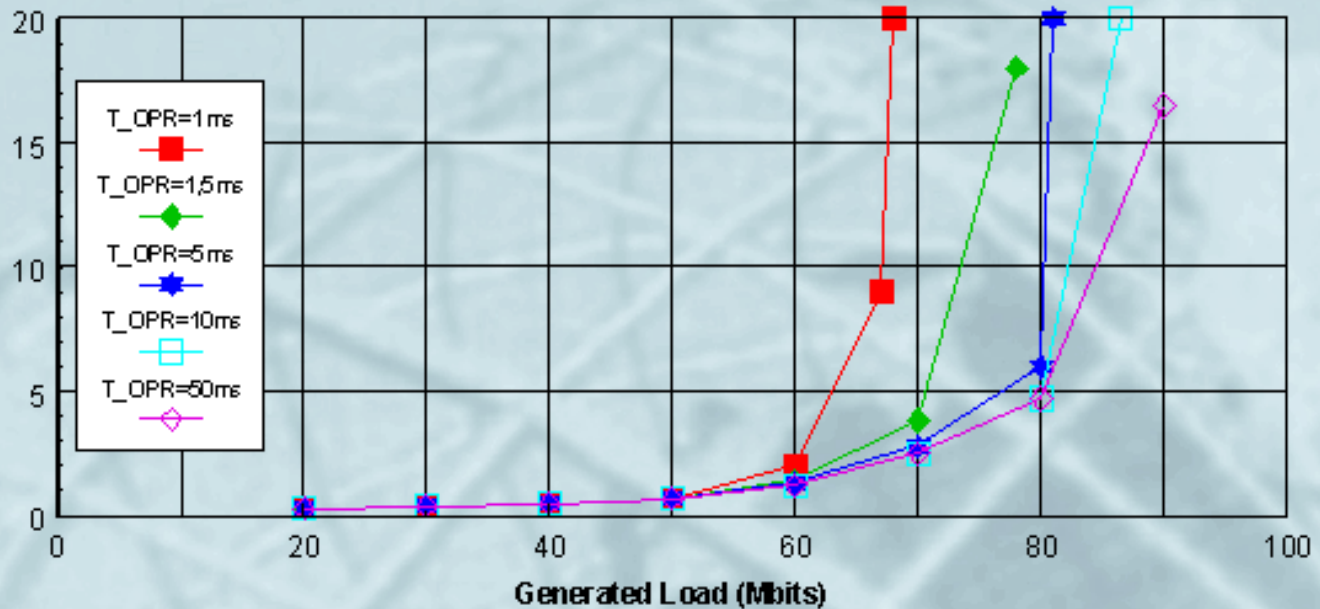
Slide 45 of 188



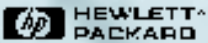
NFS Fileserver Sizing

Network Loading: FDDI Token Ring

Average Packet Time (ms)



16th Conference on Local Computer Networks
 IEEE Performance Evaluation of FDDI - What About Packet Loss
 Peter Martini, Ralf Wittenberg



46 99/09/20 Rob Lucke

Slide 46 of 188

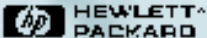


NFS Fileserver Sizing

Network Loading: 100baseTX Collision Domains

Fast Ethernet Collision Domain Loading Rules

- 100baseTX = 100 Mb/S; 100 Mb/S = **12.5 MB/S**
(Maximum theoretical)
- 100baseTX = 12.5 MB/ 4 KB per NFS IOP = **3200 NFS IOPS** (Maximum theoretical 100baseTX saturation)
- Is 100baseTX = 10X the performance of 10baseT?
(experience so far shows between 5-7X the performance)
- 100baseTX SHARED = **up to 1000 NFS IOPS**
- Notice that FDDI is still better at moving data than 100baseTX, due to no collisions and larger MTU size (4500 bytes versus 1500 bytes)



47 99/09/20 Rob Lucke

Slide 47 of 188



NFS Fileserver Sizing

Network Loading: Gigabit Ethernet (GbE)

⌘ Gigabit Ethernet (GbE) IEEE 802.3Z

<u>Transceiver</u>	<u>Transmission media</u>	<u>Distance</u>
> 1000Base-SX <i>Short wavelength</i>	62.5 μm Multi-mode Fiber	260 m
	50 μm Multi-mode Fiber	550 m
> 1000Base-LX <i>Long wavelength</i>	62.5 μm Multi-mode Fiber	440 m
	50 μm Multi-mode Fiber	550 m
	Single-mode Fiber	3 Km
> 1000Base-CX <i>4 conductor coaxial cable</i>	Twinax or Quad	25 m
> 1000Base-T <i>Unshielded Twisted Pair</i>	Category 5 UTP	100 m

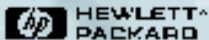
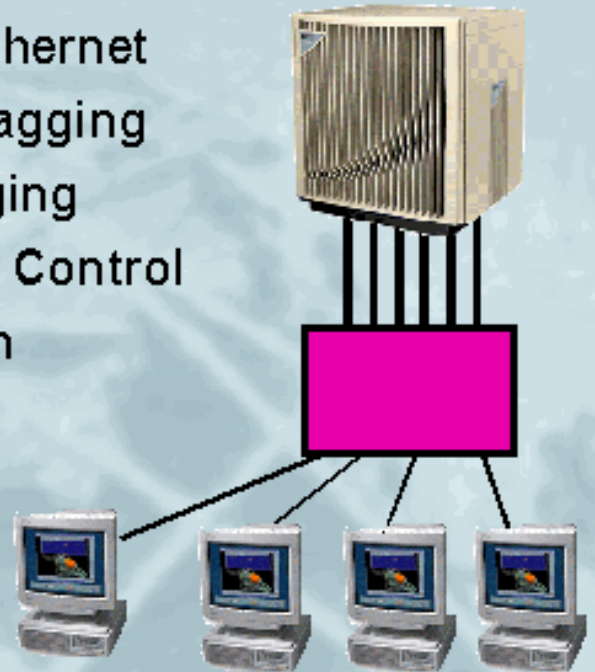


NFS Fileserver Sizing

Network Loading: Is GbE == ATM?

- ⊗ IEEE 802.3Z Gigabit Ethernet
- ⊗ IEEE 802.1P Traffic Priority Tagging
- ⊗ IEEE 802.1Q Virtual LAN Tagging
- ⊗ IEEE 802.3X Full Duplex Flow Control
- ⊗ IEEE 802.3AD Link Aggregation

- ⊗ Switches Readily Available
- ⊗ No LANE needed!
- ⊗ Original Ethernet MTU size retained (~1500 bytes)



49 99/09/20 Rob Lucke

Slide 49 of 188

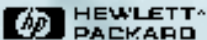


NFS Fileserver Sizing

Network Loading: ATM OC3 (155.52 Mbps)

⚗ ATM OC3 Loading Rules

- ATM OC3 = 155.52 Mb/S; 155.52 Mb/S = **19.44 MB/S**
(Maximum theoretical)
- 19.44 MB / 52 bytes per ATM Cell = 392,006 Cells/S
- 392,006 Cells/S x 48 data bytes per Cell = 17.94 MB/S
(Maximum theoretical ATM saturation)
- 17.94 MB / 4 KB per NFS IOP = **4594 NFS IOPS**
- ATM has no collisions
- The MTU size for ATM may be 9000 bytes (!)



50 99/09/20 Rob Lucke

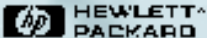


NFS Fileserver Sizing

Network Loading: ATM OC12 (622.08 Mbps)

⚗ ATM OC12 Loading Rules

- **ATM OC12 = 622 Mb/S; 622 Mb/S = 81.53 MB/S**
(Maximum theoretical)
- **81.53 MB / 52 bytes per ATM Cell = 1,567,822 Cells/S**
- **1,567,822 Cells/S x 48 data bytes per Cell =**
75.26 MB/S (Maximum theoretical ATM saturation)
- **75.26 MB / 4 KB per NFS IOP = 18,373 NFS IOPS**

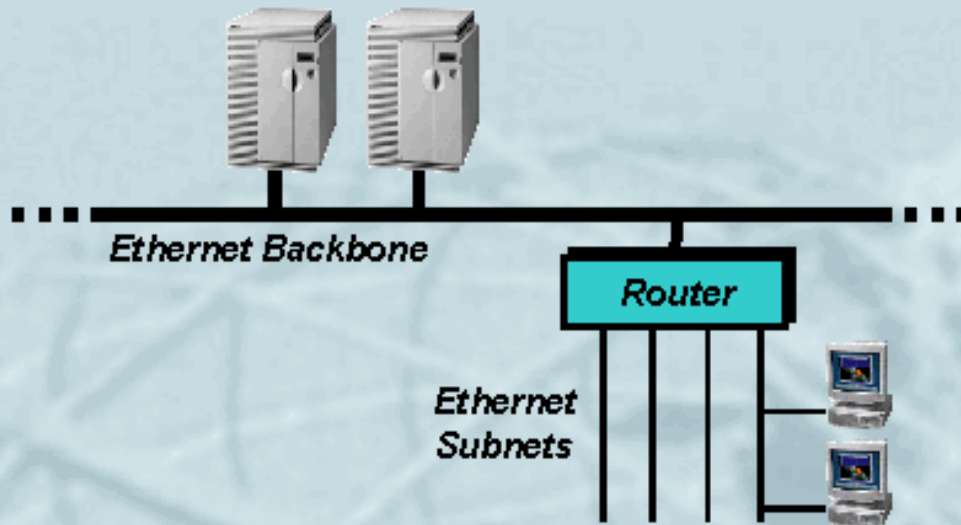


51 99/09/20 Rob Lucke



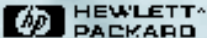
NFS Fileserver Sizing

Routed Ethernet Backbone



⚡ Routed Ethernet Backbones are Dead (1980s Architecture)

- Routers are inappropriate for the I/O rates of network OS
- Routing is not necessary; need more segments not subnets
- Router latency and bandwidth becomes a system bottleneck
- Increasing the number of segments yearly is expensive using routers



52 99/09/20 Rob Lucke



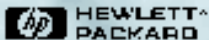
NFS Fileserver Sizing

Ethernet Switching: "Switch the LAN, route the WAN"



⚡ Ethernet Switching Replaces Multiple Network Interfaces in Servers

- Ethernet switch is 3X-10X router performance at 30% of the cost
- Ethernet switching extends useful life of Ethernet investment
- Single network investment (switch) is available to all servers
- Need a high-speed link ("large pipe") between servers and switches
- Switches handle FDDI, 10baseT, 100baseT, and other network media
- Reduction in system administration due to moves, route configuration

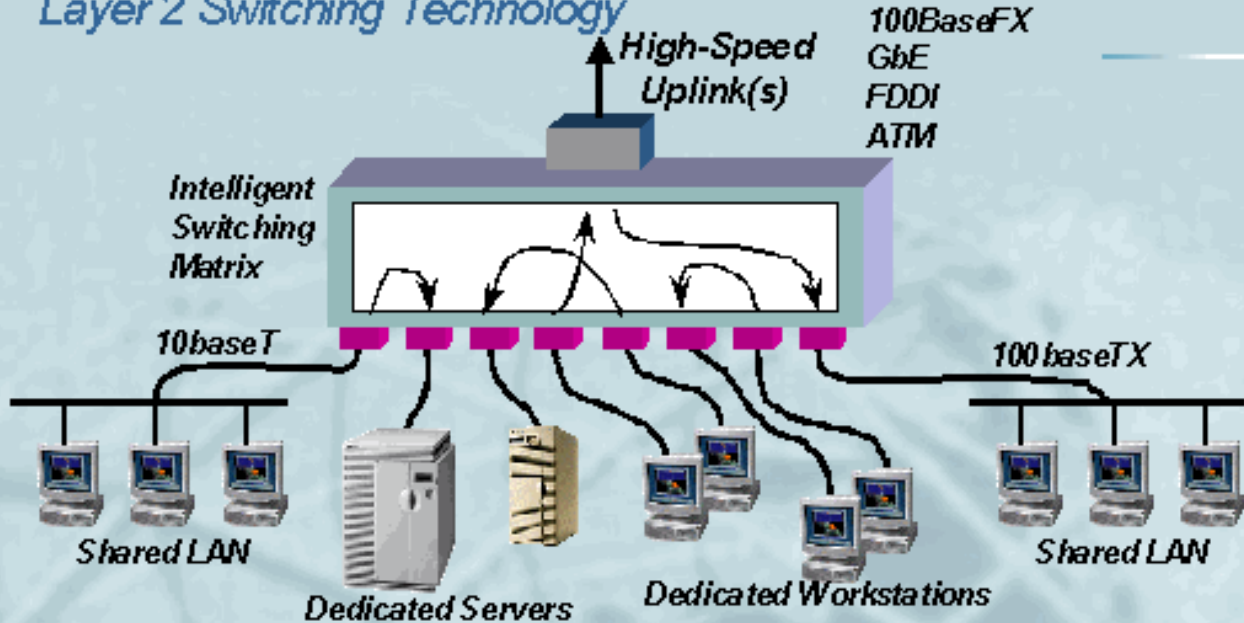


53 99/09/20 Rob Lucke



NFS Fileserver Sizing

Layer 2 Switching Technology



Switching Technology is the way to go for LANs

- > Smart learning bridges that make independent hardware connections between ports at the MAC layer
- > High throughput and low latency compared to routers
- > Create Smaller Collision Domains
- > Simple, easy to understand, "Plug-and-Play"
- > Protocol Transparent
- > Low price per port



54 99/09/20 Rob Lucke



NFS Fileserver Sizing

Layer 2 Switching Technology

- ⌘ Switches may be used to reduce the size of collision domains
- ⌘ Switches will propagate (flood) broadcasts to all ports
- ⌘ Virtual LANs (VLANs) can determine which ports get which broadcasts
- ⌘ Warning: Look for "oversubscribed" backplanes in switches!

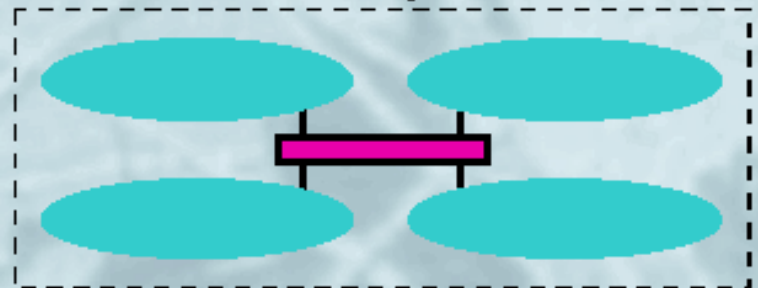


55 99/09/20 Rob Lucke

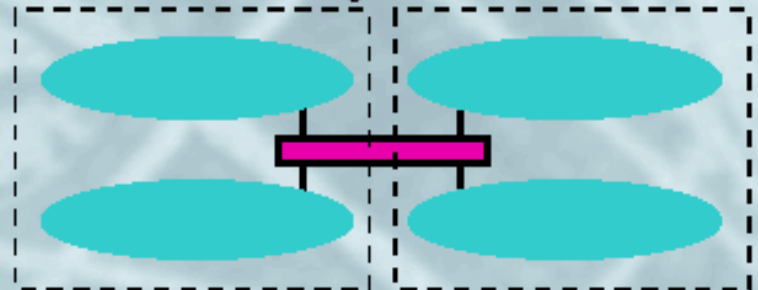
Before Installing a Switch

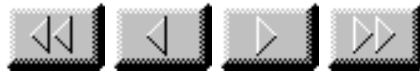


After Installing a Switch



Switching with VLANs



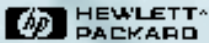
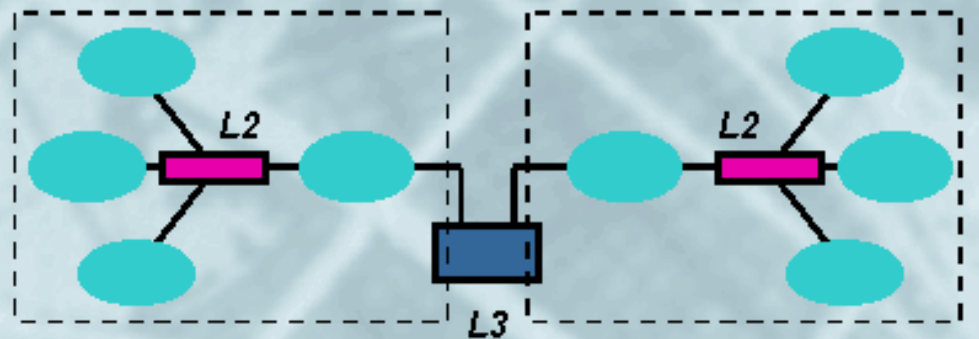
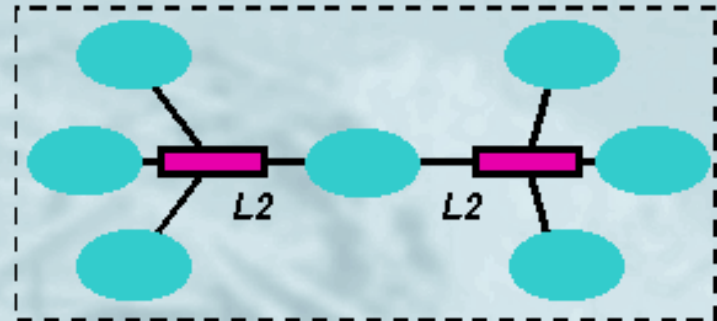


NFS Fileserver Sizing

Layer 3 Switching Technology

Layer 3 Switch Characteristics

- Optimized for LAN routing
- Wire-speed (Mpps) forwarding
- Forwards inter-subnet traffic based on Network layer addresses (layer 3)
- Forwards intra-subnet traffic based on MAC addresses (layer 2)
- Contains the flooding of broadcast traffic



56 99/09/20 Rob Lucke

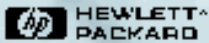
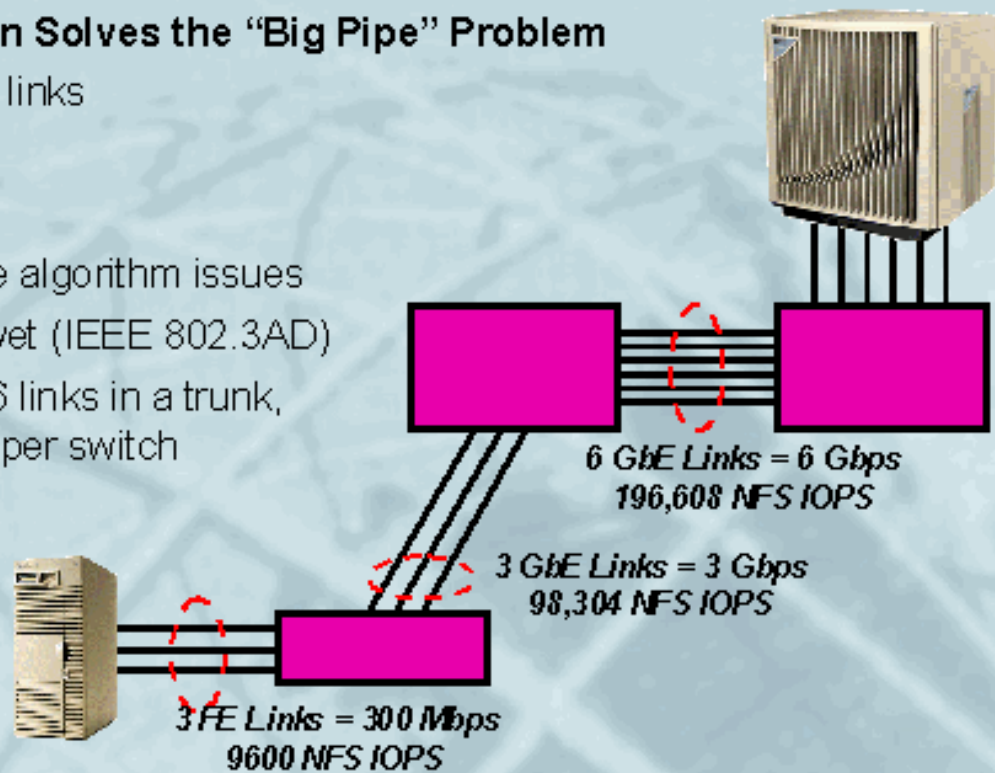


NFS Fileserver Sizing

Network Link Aggregation (Trunking)

⚡ Link Aggregation Solves the “Big Pipe” Problem

- Parallel active links
- Load sharing
- Resiliency
- Spanning Tree algorithm issues
- No standard, yet (IEEE 802.3AD)
- 3Com allows 6 links in a trunk, up to 4 trunks per switch



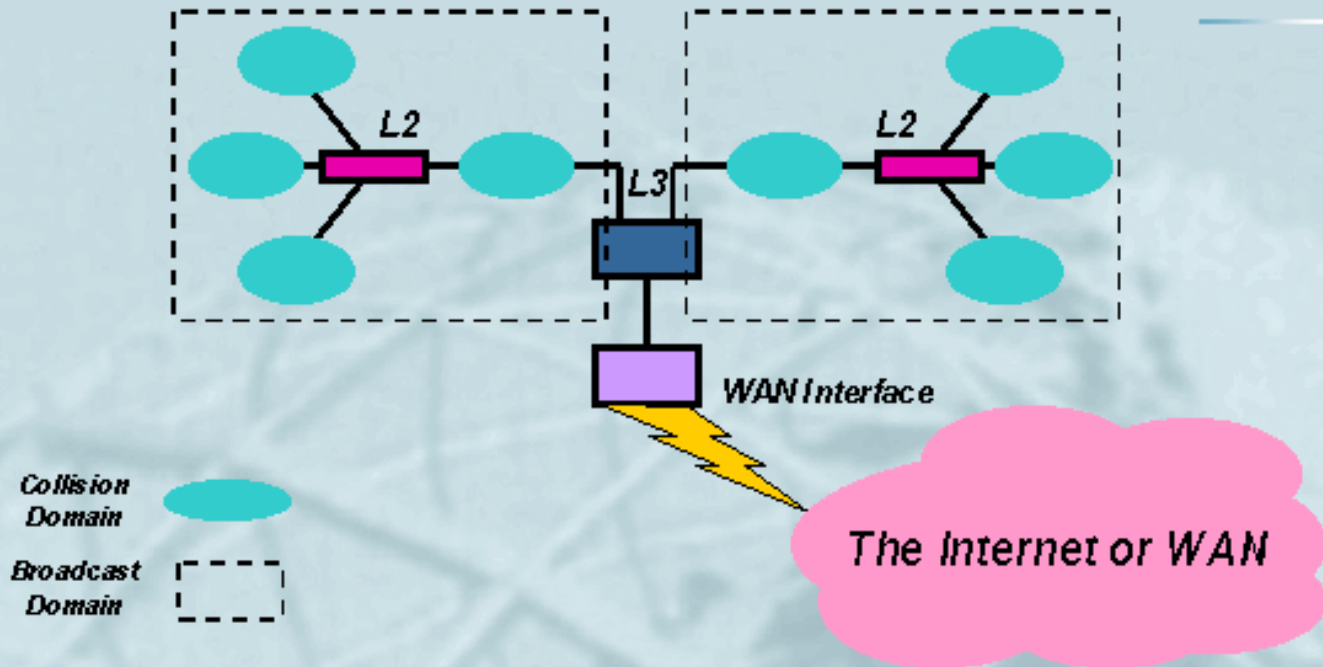
57 99/09/20 Rob Lucke

Slide 57 of 188



NFS Fileserver Sizing

But what about "traditional" routers and ATM?



- ⌘ Traditional routers are still needed for security and access to the WAN
- ⌘ ATM is complicated, expensive, and requires LANE, but is useful for interfaces at the edge of the LAN or where one technology is needed for LAN, WAN, voice, data, and video



NFS Fileserver Sizing

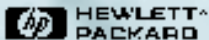
Examining the Network Interface: lanscan

```

root@hpupcra: /root_home
# lanscan
Hardware Station      Crd Hardware Net-Interface  NM  MAC           HP DLFI M...
Path      Address          In# State  NameUnit State ID  Type           Support Num
E/1E/6    Cx0E00C9CB987A  0  UP    lan0      DOWN  4  ETHER          Yes     52
E/2C/5/   Cx0360E044DC2B  1  UP    lan1      DOWN  5  ETHER          Yes    176
E/2C/5/8  Cx0360E0588C8   2  UP    btlan02   UP     6  ETHER          Yes     75
  
```

MAC (Media Access Control) Address or LLA (Link Level Address)

Use the lanscan command to see all network interfaces in the server. The Network Management ID (NMI) can be used with landiag to examine lower-level statistics, including the MTU size, the connection rate, framing errors, etc.



59 99/09/20 Rob Lucke



NFS Fileserver Sizing

Examining the Network Interface: landiag

Results of "landiag>I" then "nmid 6" then "d"

Statistics
for
server's
100baseT
link

```

--(0)hpvsa/mnt_krns
Network Management ID      - 6
Description                 - btlan92 Hewlett Packard LAN adapter RevC
Type (value)                - ethernet-standard16
MTU Size                    - 1500
Speed                       - 100000000
Station Address             - 0260b05011d0
Administrative Status (value) - up(1)
Operation Status (value)   - up(1)
Last Change                 - 0
Inbound Octets              - 37759000
Inbound Unicast Packets    - 85294
Inbound Non-Unicast Packets - 255722
Inbound Errors              - 6319
Inbound Discards            - 0
Inbound Unicast Packets    - 7407
Outbound Octets             - 60345914
Outbound Unicast Packets   - 102923
Outbound Non-Unicast Packets - 4150
Outbound Errors             - 0
Outbound Discards          - 0
Outbound Queue Length      - 0
Specific                    - 0

LAN Interface test mode. LAN Interface Net Mgmt ID = 6

clear      - Clear statistics registers
display   - Display LAN Interface status and statistics registers
end       - End LAN Interface Administration, return to Test Selection
menu      - Display this menu
nmid      - Network Management ID of the LAN Interface
quit      - Terminate the Administration, return to shell
test      - Test LAN Interface to execute its selftest

Enter command:

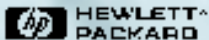
```

Packets can be:

Unicast *specific host*

Multicast *multiple hosts*

Broadcast *everybody on the LAN*



60 99/09/20 Rob Lucke

Slide 60 of 188

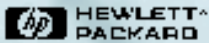
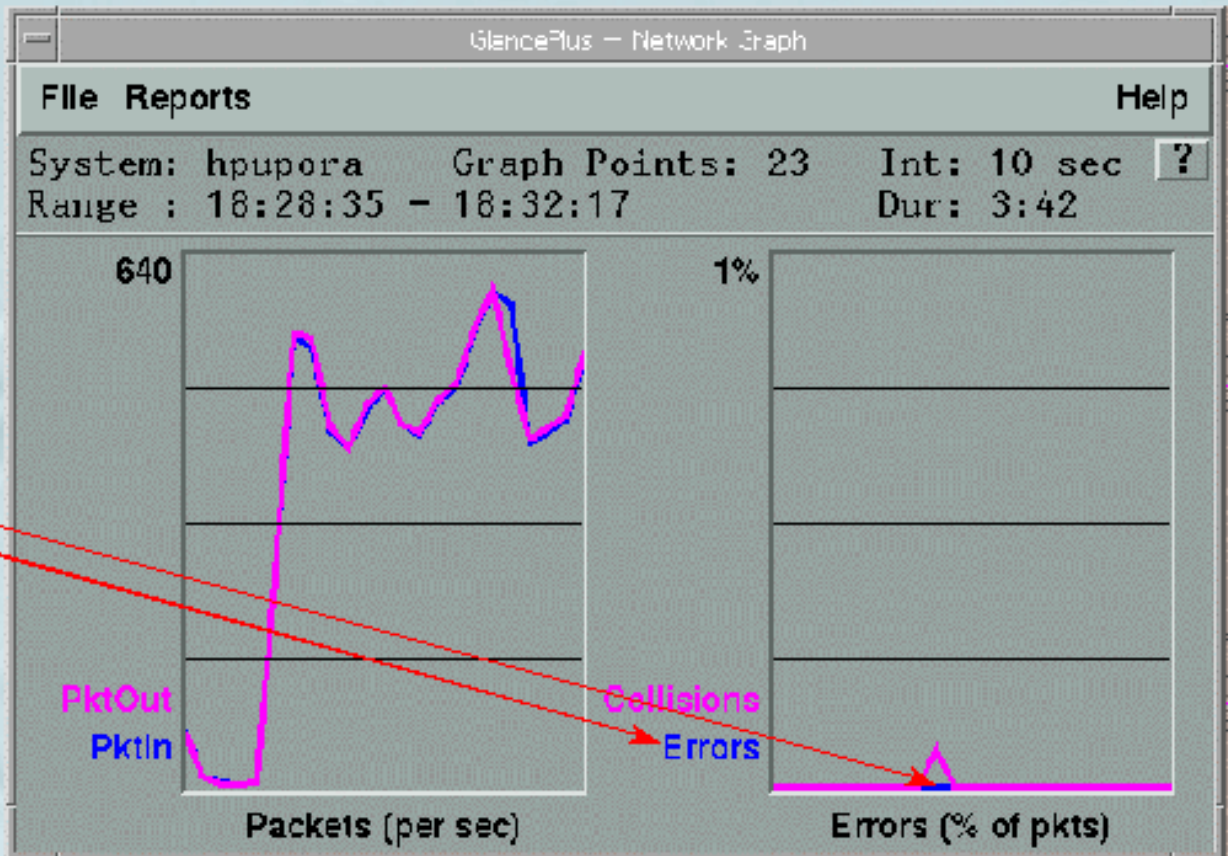


NFS Fileserver Sizing

Examining the Network: Looking for Errors

Look for low error and collision rates. There are low collision rates here because the link is a dedicated switch port. But should full-duplex 100baseT EVER have collisions? NO!

If the error rate is greater than zero, start looking at network interfaces, cabling, hubs, switches, etc. and check that you are really running full-duplex.



61 99/09/20 Rob Lucke

Source: GPM>Reports->Network Info-> Network Graph

Slide 61 of 188



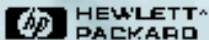
NFS Fileserver Sizing

Examining the Network: Specific Notes

⌘ Cabletron equipment appears to have issues with properly fragmenting an FDDI packet that is heading for an ethernet segment. If you are getting unexplained errors or retries, try lowering the FDDI MTU size to 1500 bytes to see if they disappear



⌘ Cisco switches appear to drop UDP traffic under heavy loads or burst conditions. If you are seeing unexplained NFS retries, then try enabling some backup buffering in the switch. For software revisions 5.3 and higher, there is an undocumented command: “**set option flowcontrol enable**” that will allow additional buffering for UDP traffic.

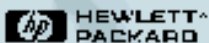


62 99/09/20 Rob Lucke



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - **Memory**
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



HEWLETT
PACKARD

63 99/09/20 Rob Lucke



NFS Fileserver Sizing

Memory Subsystem

System Memory is needed for 3 primary reasons:



➤ Fileserver kernel

- ◆ Assume 32-50 MB for kernel, data structures, network memory, root login, and GlancePlus

➤ Fileserver Applications

- ◆ Add memory for additional administration and performance tools
- ◆ Add memory for additional server processes like NIS, DNS, SDU, X11 Font Server, etc.
- ◆ *Remember that any large, CPU or file-intensive applications may damage NFS fileserver performance*

➤ Fileserver I/O Buffer Cache

- ◆ Start with ~128 KB for each NFS IOP (128 MB per 1000 NFS IOPs)
- ◆ Measure and tune from this starting point



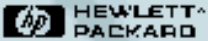
64 99/09/20 Rob Lucke

Slide 64 of 188



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - **Disk**
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3





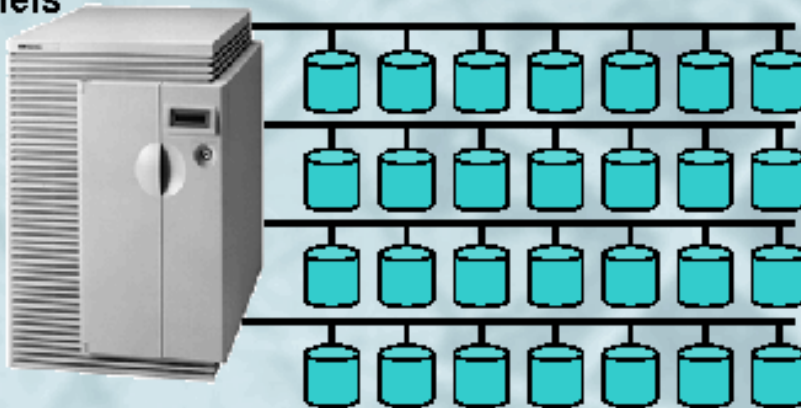



NFS Fileserver Sizing

Mass Storage Considerations: Stand-Alone Disks

⌘ Also known as JBOD (J**u**st a **B**unch **O**f **D**isks)

- Provide good performance but no redundancy
- Performance from lots of independently moving heads
- Use MirrorDisk/UX and LVM to gain redundancy
- Stripe stand-alone disks with LVM to gain performance
- Can quickly run out of expansion ability because of limited I/O channels



 HEWLETT
PACKARD

67 99/09/20 Rob Lucke

Slide 67 of 188




NFS Fileserver Sizing

Mass Storage Considerations: Mirroring Stand-Alone Disks

⌘ Mirroring Stand-Alone Disks

- 100% storage overhead (Must duplicate space for mirror)
- Performance benefit: Reads will come from least busy mirror
- Cannot mirror and stripe at the same time (more on this later)
- MirrorDisk/UX now available for 700 and 800 systems
- Writes are duplicated to each mirror group (2X the I/O)
- Supports up to 2 mirrors
- Mirror groups may be "split" and "joined" and will synchronize data



 HEWLETT
PACKARD

68 99/09/20 Rob Lucke

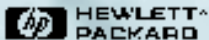
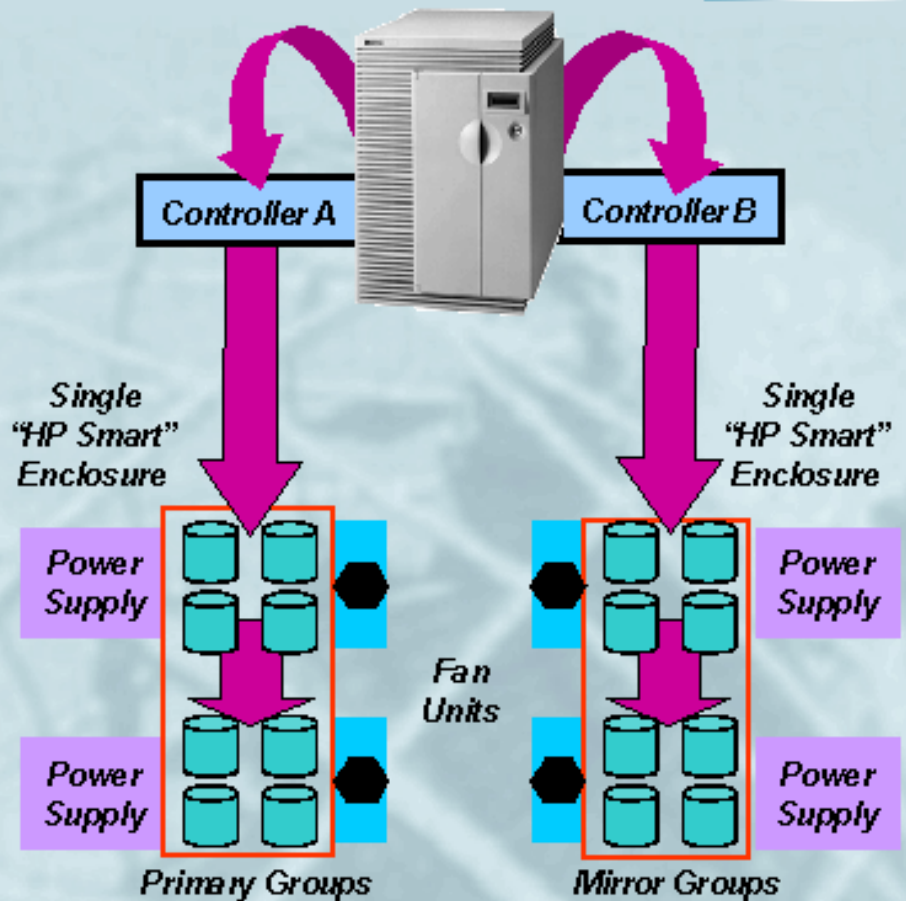
Slide 68 of 188



NFS Fileserver Sizing

Mass Storage Considerations: HP Smart Enclosures

- Controllers are GSC F/W SCSI or Fibre Channel I/Fs
- No single point of failure
- F/W or S/E devices
- Can be used with FC SCSI Mux
- Enclosures can be "split in half"
- Hot-plug disks, fans, power supplies

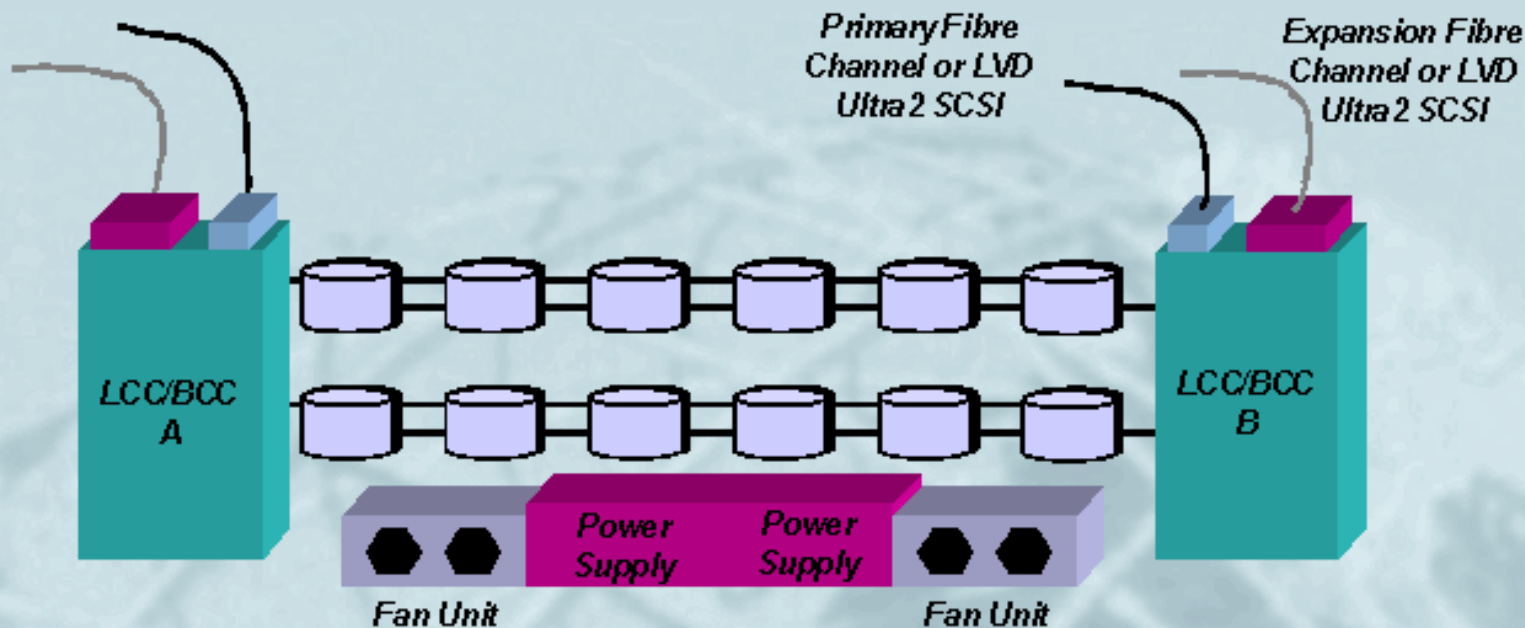


69 99/09/20 Rob Lucke



NFS Fileserver Sizing

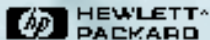
Mass Storage Considerations: SC-10/FC-10



*3.5 EIA Units, 10 Disks per enclosure [11 enclosures per 2m rack]
2m Rack = 110 disks = 0.99 TB (9.1 GB disk) or 1.98 TB (18 GB disk)*

*Loop Control Card (LCC) Bus Control Card (BCC)
Fibre Channel 100 MB/s (FC-10) [Dual FC to each disk]
LVD Ultra2 SCSI 40 MB/s (SC-10)*

*Hot Plug Disks, Power Supplies, Fans, LCC/BCC
Real-time hardware monitoring and event detection*



70 99/09/20 Rob Lucke

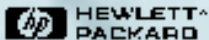
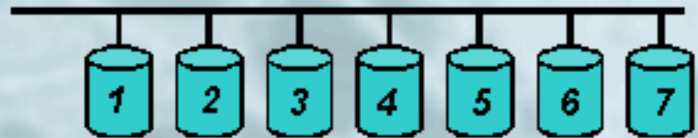


NFS Fileserver Sizing

Mass Storage Considerations: Striping Stand-Alone Disks

⌘ Striping Stand-Alone Disks with LVM

- Avoids exceeding the throughput of a single disk by spreading the I/O over multiple devices
- Stripes may be made across channels, as well as across devices
- Supported LVM Striping:
 - ◆ “lvcreate -D y -s g -i <# pvols> -l <stripe size>”
 - ◆ Stripe size can be 4, 8, 16, 32, or 64 KB
 - ◆ Cannot be mirrored
- “Extent striping”
 - ◆ “lvcreate <no size parameter>”
 - ◆ “lvextend -L <size + 1 extent> ... <lv_name> <physical device>”
 - ◆ Can be mirrored
 - ◆ Stripe is limited to Volume Group extent size (minimum 1 MB)
 - ◆ Long set up time (each extend takes about 1 second)
 - ◆ There are tools to automatically generate a script to perform the operations



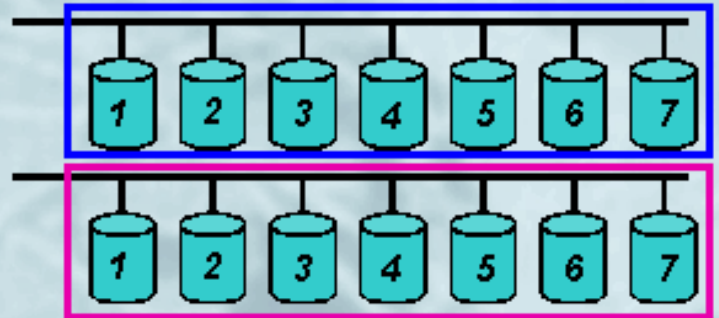


NFS Fileserver Sizing

Mass Storage Considerations: RAID Modes

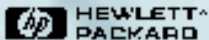
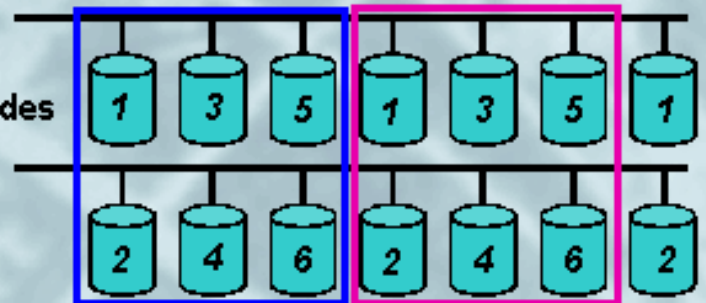
RAID 1 **\$\$**

- Disk mirroring
- Drives operate asynchronously
- Provides redundancy with **100% storage overhead**



RAID 0/1 **\$\$**

- Disk striping and mirroring
- Provides best performance of RAID modes
- Provides redundancy with **100% storage overhead**



72 99/09/20 Rob Lucke



NFS Fileserver Sizing

Mass Storage Considerations: RAID Modes

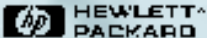
~~RAID 3~~

- > Disk striping, dedicated parity drive
- > Combines 5 drives to act as a large virtual drive
- > RAID 3 provides poor performance for any multi-user random access activity
- > 25% storage overhead



~~RAID 5~~

- > Disk striping, parity data distributed for performance
- > Independent data paths to drives
- > RAID 5 provides redundancy, but poor write performance for multi-user random-access activity
- > RAID 5 provides good performance for read-mostly file systems like (/usr, /opt, /home)
- > RAID 5 supports 3-16 drives in a group, 5 is typical

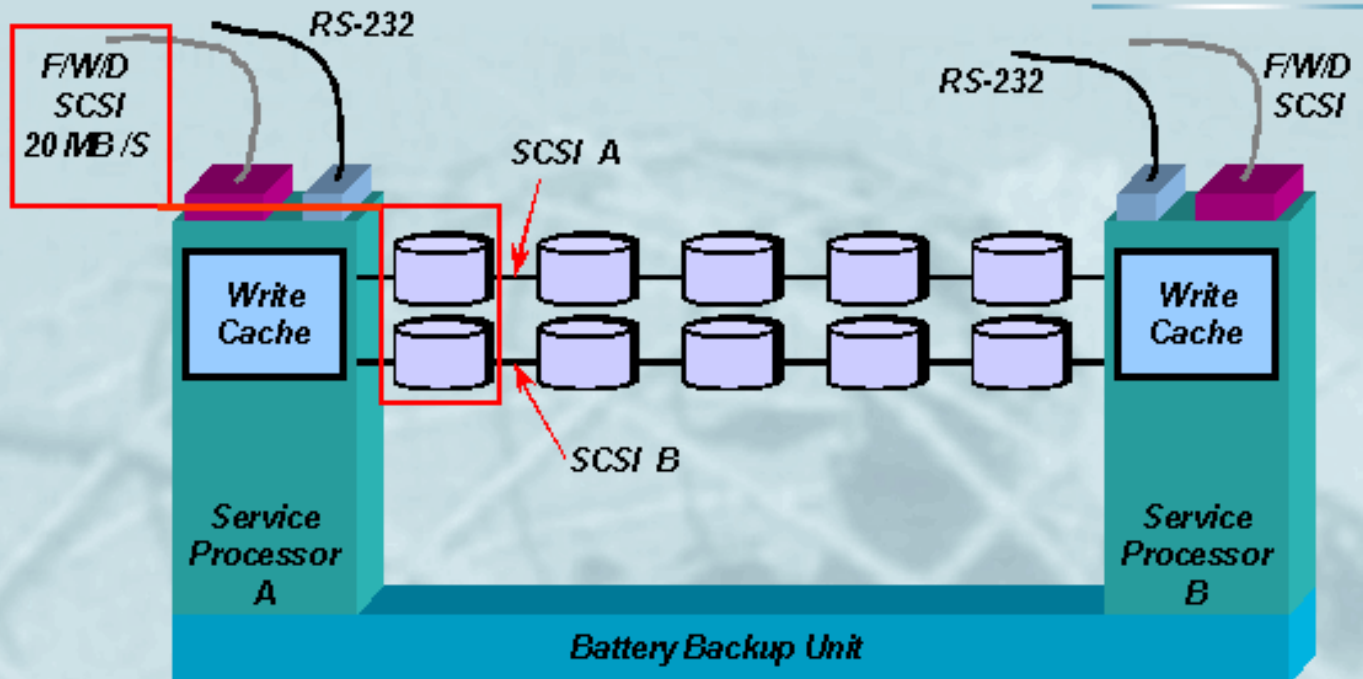


73 99/09/20 Rob Lucke

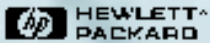


NFS Fileserver Sizing

Mass Storage Considerations: Model 10 RAID Arrays



RAID modes 1, 1/0, and 5
10 Disk slots
36.9/79.4 GB capacity with 4.2/8.8 GB Disks
Up to 16 MB of write cache per controller

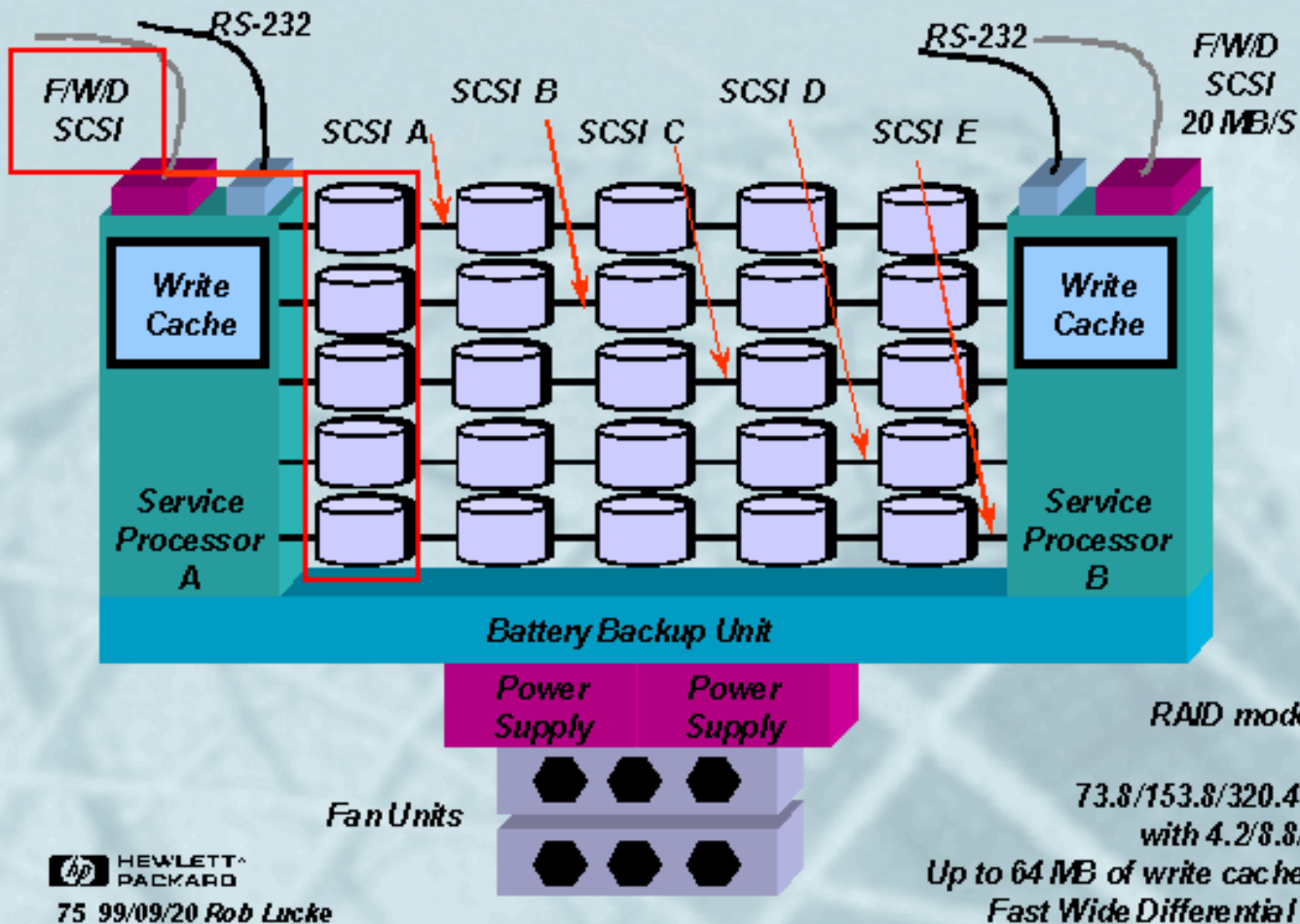


74 99/09/20 Rob Lucke



NFS Fileserver Sizing

Mass Storage Considerations: Model 20 RAID Arrays

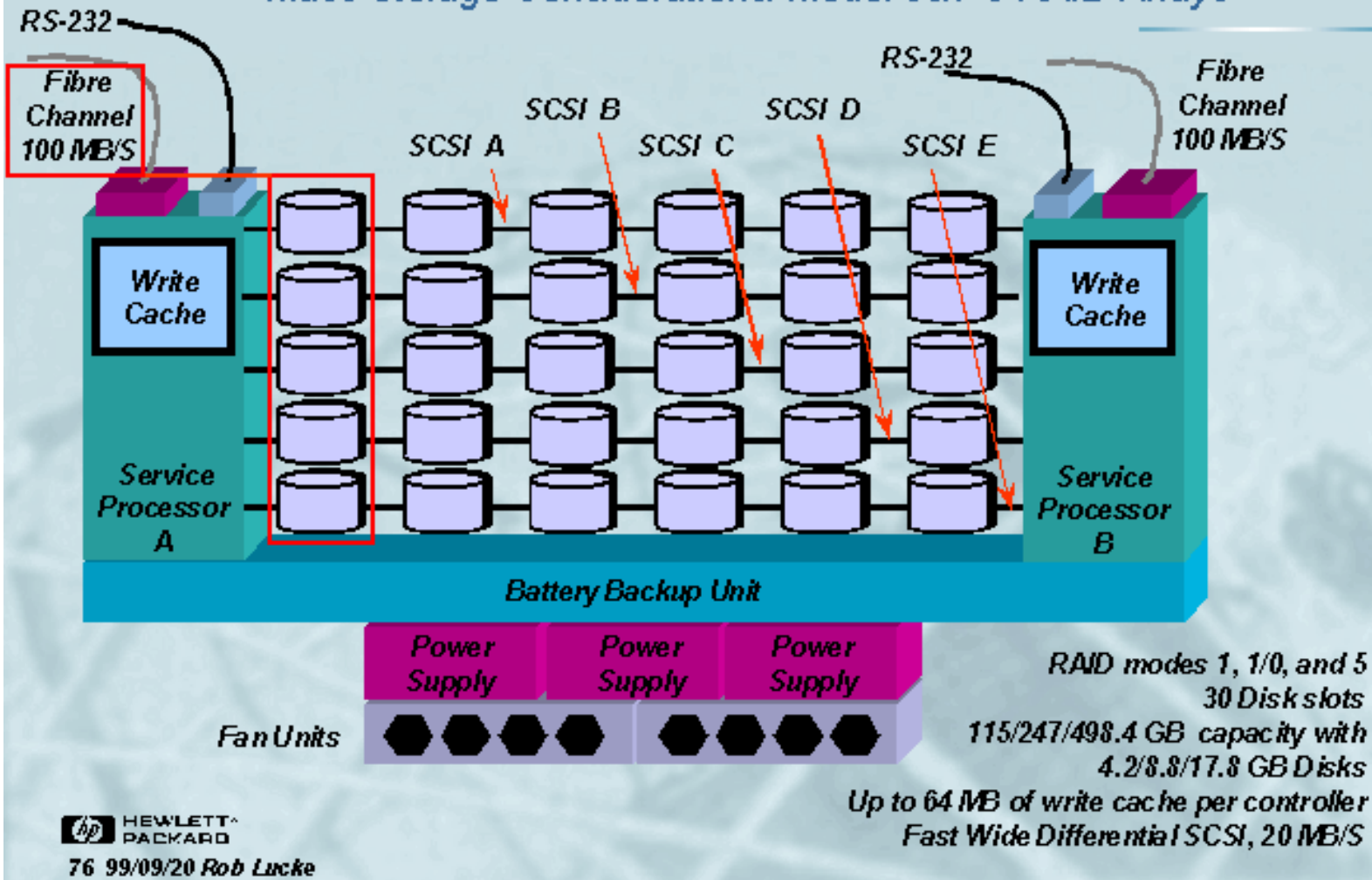


Slide 75 of 188



NFS Fileserver Sizing

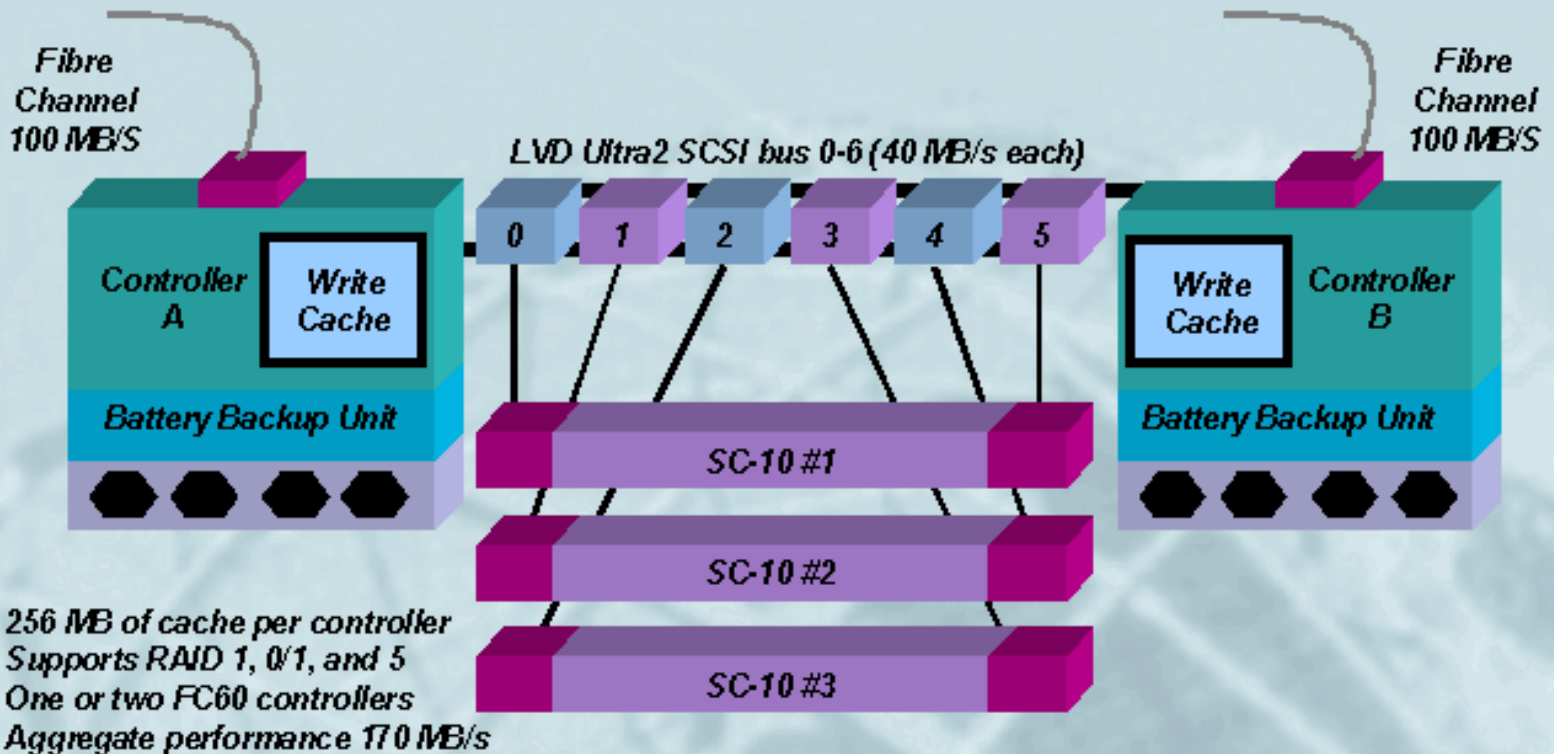
Mass Storage Considerations: Model 30/FC RAID Arrays



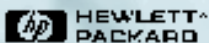


NFS Fileserver Sizing

Mass Storage Considerations: FC-60 RAID Array



Realtime hardware monitoring and event detection



77 99/09/20 Rob Lucke

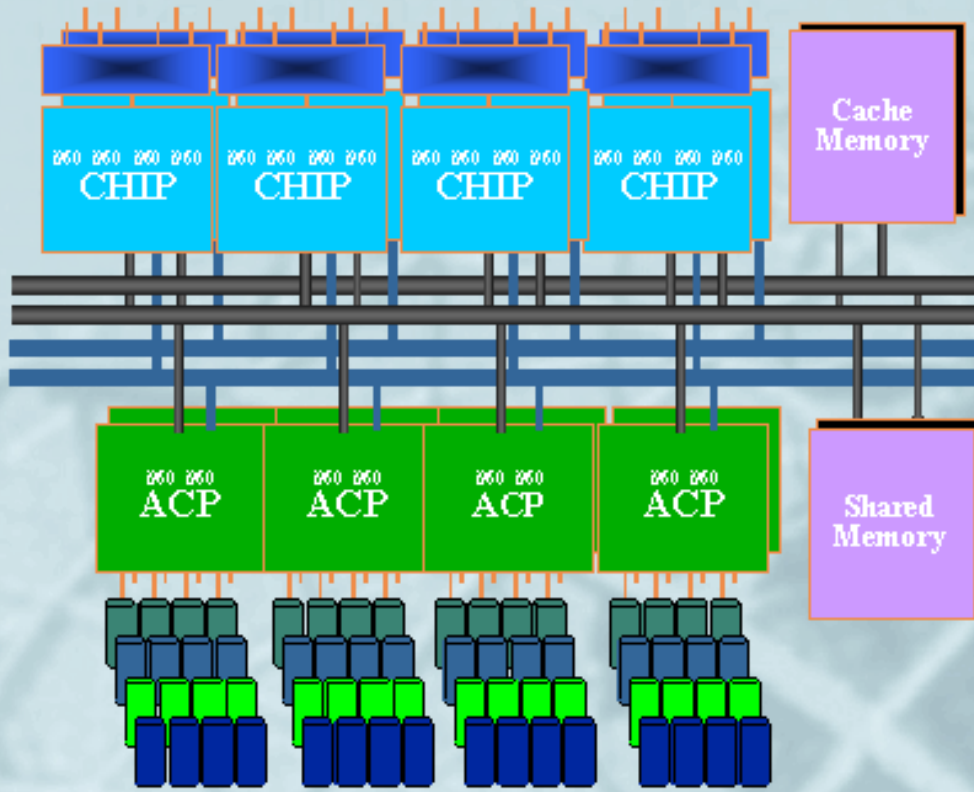
1.6m Rack = 1 FC60 + 8 SC10 = 1.31 TB (18 GB disks)
 2 FC60 + 7 SC10 = 655.20 GB (9.1 GB disks)
 2.0m Rack = 1 FC60 + 10 SC10 = 1.638 TB (18 GB disks)
 2 FC60 + 9 SC10 = 819 GB (9.1 GB disks)

Slide 77 of 188



NFS Fileserver Sizing


Mass Storage Considerations: XP-256



Two 240MB/sec Data Buses
Two 130MB/sec Control Buses

17 GB to 9 TB of Storage
Up to 32 SCSI channels
Up to 64 FC channels
256 MB to 16 GB of cache

High speed 12000 RPM super-high reliability disks
15GB & 36.9GB

 HEWLETT
PACKARD
78 99/09/20 Rob Lucke



NFS Fileserver Sizing

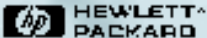
Mass Storage Considerations: AutoRAID

⊗ AutoRAID: Performance without the RAID trade offs

- Dynamic data migration between RAID 0/1 and RAID 5 based on access patterns
- Frequently accessed data kept in RAID 0/1, infrequently accessed data moved to RAID 5
- RAID mode cannot be specified for LUNs
- Data location (RAID 0/1 or RAID 5) cannot be determined at any given time



- | | |
|---------------------------|--------------------------------|
| ➤ <u>Model 12</u> | <u>Model 12H</u> |
| 24 MB memory per ctrlr | 96 MB memory per ctrlr |
| 19.5/39.5 GB capacity | 37.8/82.3/166.4 GB capacity |
| Supports 2.1/4.3 GB disks | Supports 4.3/9.1/18.2 GB disks |



79 99/09/20 Rob Lucke

Slide 79 of 188

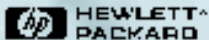


File Server Sizing

Mass Storage Considerations: Channel Performance

	Maximum Rate	Transfer Rate**	Bus Width	Maximum Distance	Maximum Devices
Single-Ended SCSI (EISA)	15 MB/S	12-13.5 MB/S	16-bits		Don't Even Think about it
Single-Ended SCSI (HP-PB)	5 MB/S	3 MB/S	8 bits	6 m	7
Fast-Wide SCSI (HP-PB)	20 MB/S	7-10 MBS	16-bits	25 m	15
Fast-Wide SCSI (HSC)	20 MB/S	12-15 MB/S	16-bits	25 m	15
Ultra-SCSI	40 MB/S	32 MB/S	16-bits		15
HD Fibre Channel	100 MB/S	Varies	N/A	2 Km	126 ports per loop
FD Fibre Channel	200 MB/S	Varies	N/A	2 Km	126 ports per loop

****** In reality, the maximum transfer rate on the SCSI bus for single device operations may be limited by the maximum transfer rate of the SCSI device. For example, the 7200 RPM disks can transfer 5.9 - 9.0 MB/S. Note that higher priority devices (larger SCSI ID) may also starve lower priority devices.



80 99/09/20 Rob Lucke



File Server Sizing

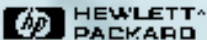
Mass Storage Considerations: Channel Performance

	Maximum Data Rate	F/W I/F per Channel	F/W Disks per Bus	Ultra I/F per Channel	Ultra Disks per Bus
HP-PB	32 MB/s	2	14	N/A	N/A
HSC	120 MB/s	6 ¹	28	3	28
2X PCI	240 MB/s	12	21	6	21
Turbo (4X) PCI	480 MB/s ³	24	168	8	168
HD Fibre Channel	100 MB/s	5	35	3	21
FD Fibre Channel	200 MB/s	10	70	N/A	N/A

¹ There are only 4 expansion slots per HSC bus converter

² The HSC bus is limited to 112 MB/S sustained throughput

³ We must wait for bus expansion to fully use one of these channels



81 99/09/20 Rob Lucke

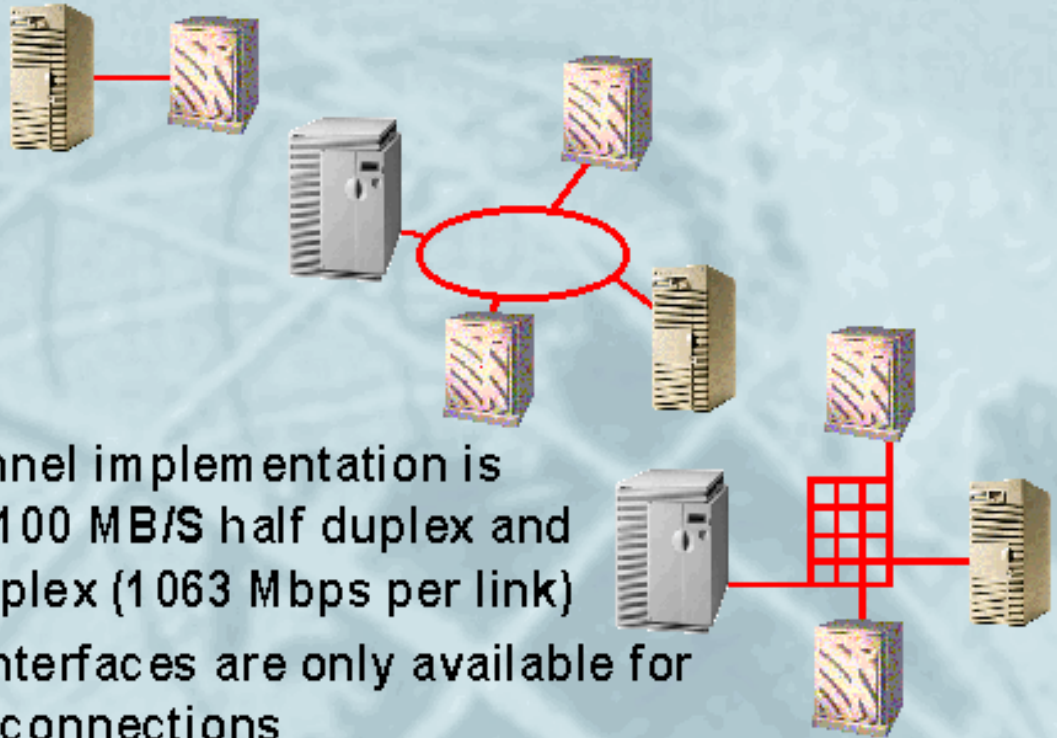


File Server Sizing

Mass Storage Considerations: Fibre Channel

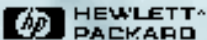
⌘ Fibre Channel comes in three topologies:

- Point-to-point
- Arbitrated loop
- Switched



⌘ HP's Fibre Channel implementation is arbitrated loop, 100 MB/S half duplex and 200 MB/S full duplex (1 063 Mbps per link)

⌘ Fibre Channel interfaces are only available for PCI or HSC bus connections



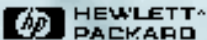
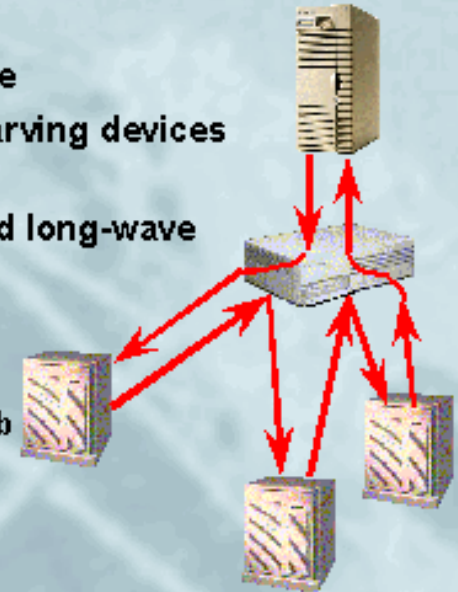
82 99/09/20 Rob Lucke



File Server Sizing

Mass Storage Considerations: Fibre Channel

- _ HP supports two arbitrated loop topologies: direct connect and hub connect
- _ Maximum of two cascaded hubs per F/C interface
- _ Fibre Channel uses "fair" arbitration to avoid starving devices (as in the SCSI 6-0,15-8 priority scheme)
- _ Hub connect is implemented with short-wave and long-wave Fibre-Channel hubs
 - > Short-wave
 - ◆ Supports up to 500 m between devices
 - ◆ Nine short-wave and one long wave ports per hub
 - > Long-wave
 - ◆ Supports up to 2 Km between devices
 - ◆ 10 short-wave ports per hub
 - > Sum of lengths of all links < 5 Km
- _ Fibre Channel to SCSI MUX has up to four FW SCSI ports supporting up to 15 devices per port, and up to two short-wave F/C adapters



83 99/09/20 Rob Lucke

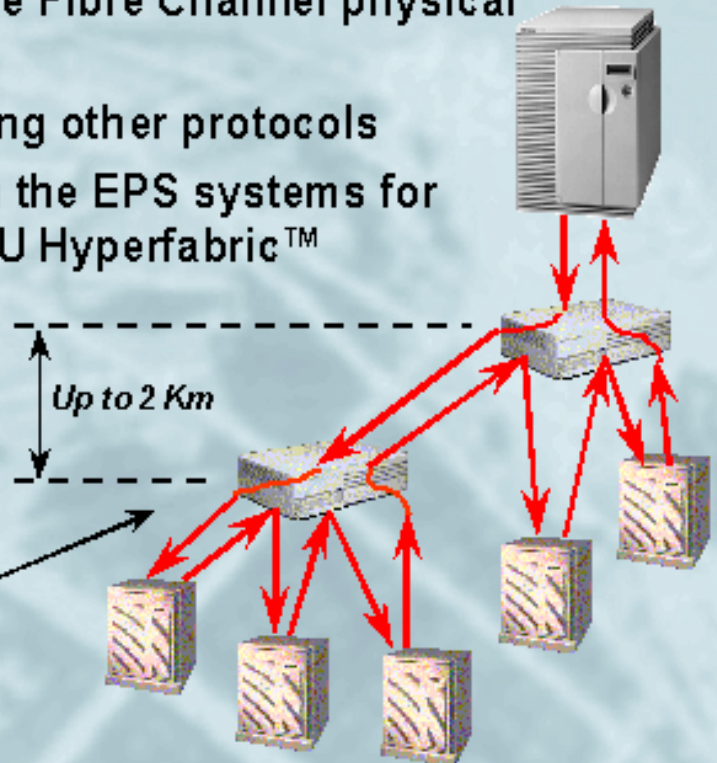


File Server Sizing

Mass Storage Considerations: Fibre Channel

- ⌘ Note that Gigabit Ethernet uses the Fibre Channel physical layer
- ⌘ Fibre Channel is capable of carrying other protocols
- ⌘ Switched Fibre Channel is used in the EPS systems for high-speed (266 Mbps) CPU to CPU Hyperfabric™ connections

Up to nine disk arrays may be connected to a hub. Each disk array may have 8 separate LUNs. Each LUN is a separate file system.

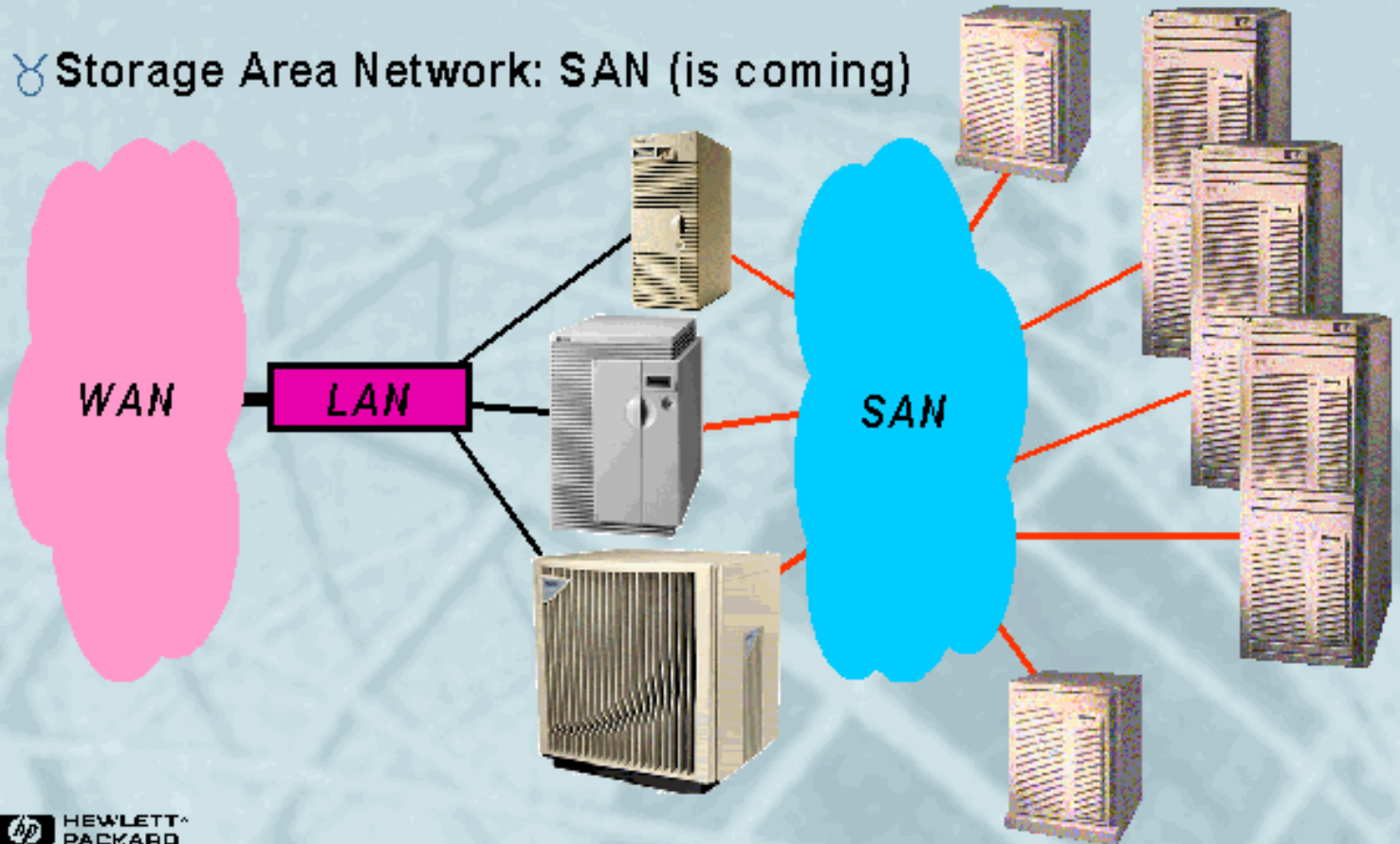




File Server Sizing

Mass Storage Considerations: Fibre Channel

♂ Storage Area Network: SAN (is coming)



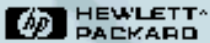
 HEWLETT
PACKARD
85 99/09/20 Rob Lucke

Slide 85 of 188



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning**
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



86 99/09/20 Rob Lucke



File Server Tuning

File System Time Line

HP-UX 10.00 HP-UX 10.01 HP-UX 10.10 HP-UX 10.10 HP-UX 10.20 HP-UX 11.00

HFS

JFS Version 2

Large File System
Support (128 GB)

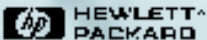
Large File Support
Large UID/GID Support

JFS Version 3

JFS Version 3.3

JFS Version 2 file systems are upwards compatible, but JFS Version 3 may not be mounted on previous HP-UX versions.

The default type for creation is JFS Version 3. JFS Version 2 file systems may be upgraded to JFS Version 3 with the "vxupgrade" command.



87 99/09/20 Rob Lucke

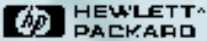
Slide 87 of 188



File Server Tuning

HFS File System

- ⌘ HFS (also known as UFS) has been around a long time
- ⌘ The file system structures are designed for multiple platter and multiple head disk mechanisms
- ⌘ System failures can require hours of file system recovery via "fsck", since the entire structure must be checked
- ⌘ The file system may only be defragmented by copying from one volume to another with "dcopy"



88 99/09/20 Rob Lucke

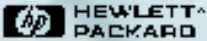
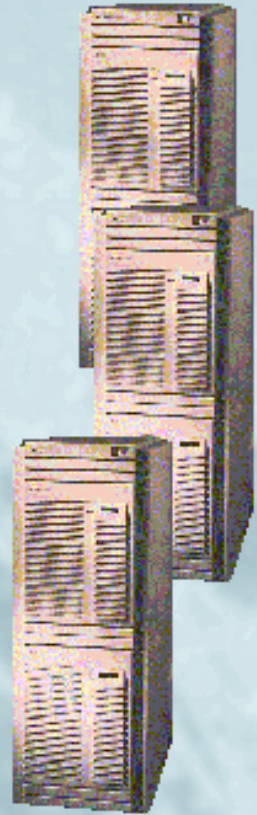
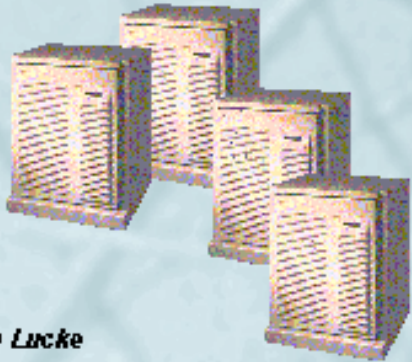
Slide 88 of 188



File Server Tuning

HFS File System

- ⌘ There are several “issues” with HFS and NFS
- ⌘ The **ENTIRE** buffer cache must be searched on file deletes or closes (unlike JFS, no “dirty” block list is maintained)
- ⌘ Performance degrades for both clients and servers that use HFS, based on buffer-cache size
- ⌘ **DO NOT EXPORT HFS FILE SYSTEMS VIA NFS**
- ⌘ **MINIMIZE THE USE OF HFS ON CLIENT SYSTEMS**



HEWLETT
PACKARD

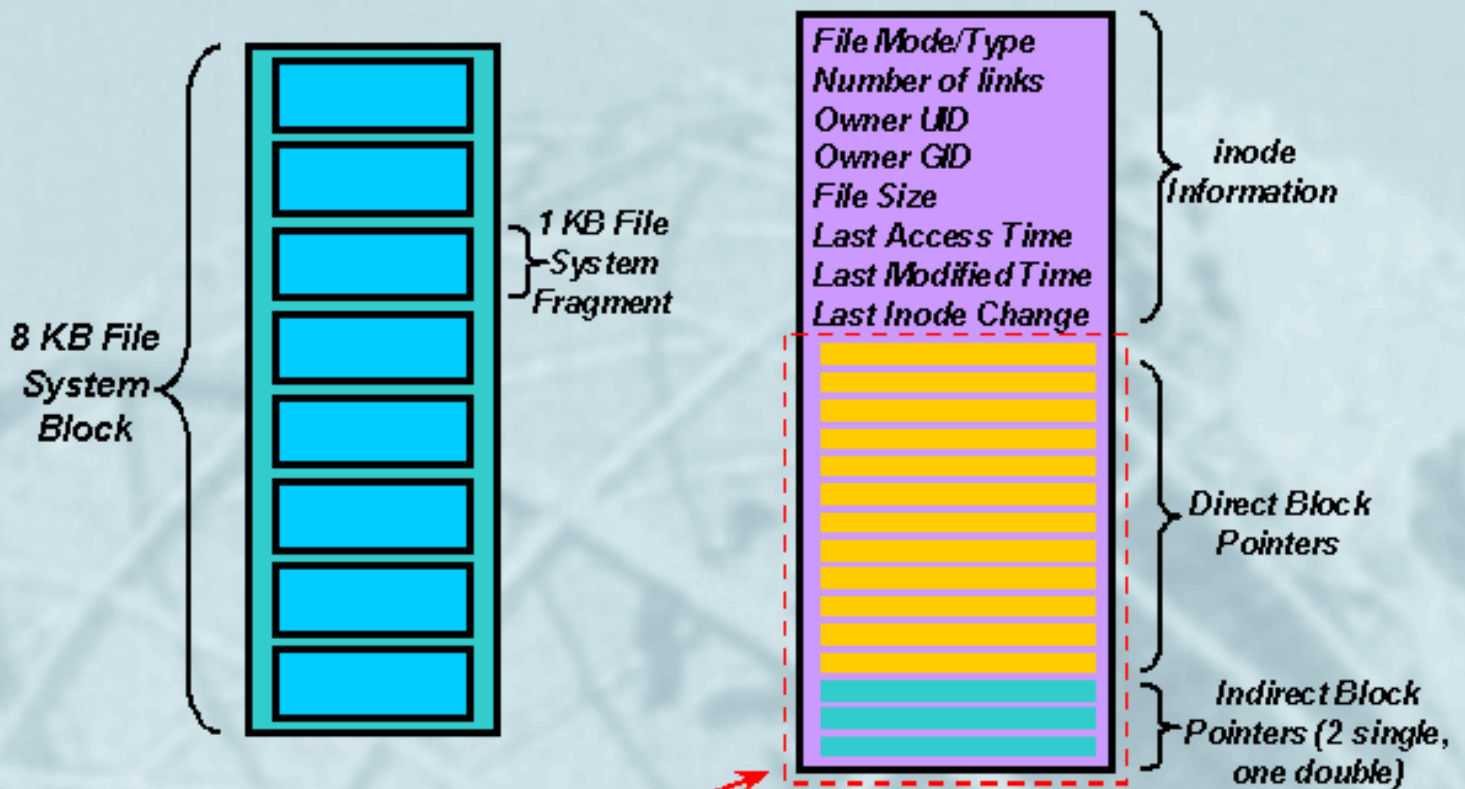
89 99/09/20 Rob Lucke

Slide 89 of 188



File Server Tuning

HFS File System Structure

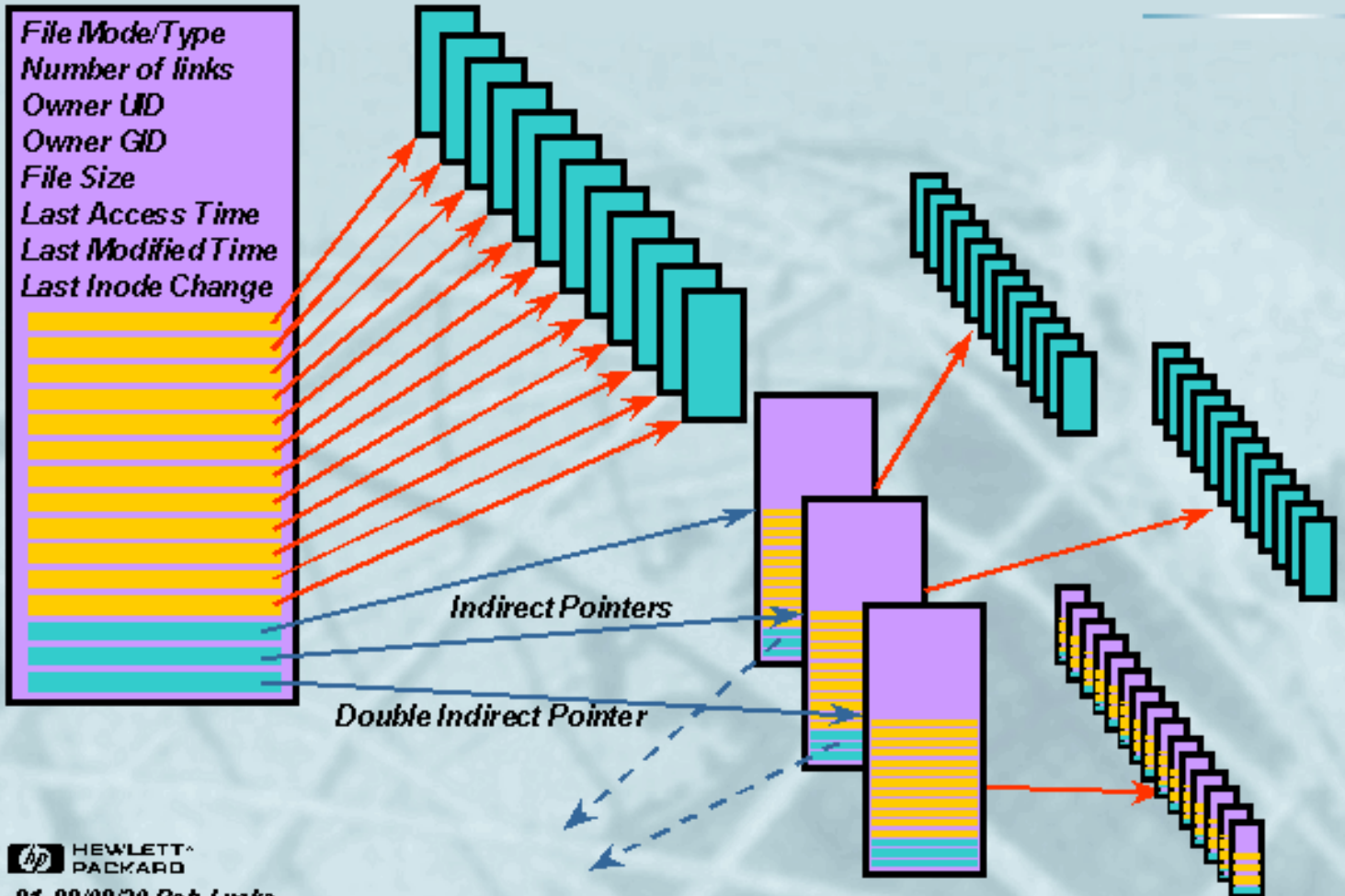


This space (60 bytes) may be used for symbolic link information if the kernel parameter "create_fastlink" is set to 1



File Server Tuning

HFS File System Structure

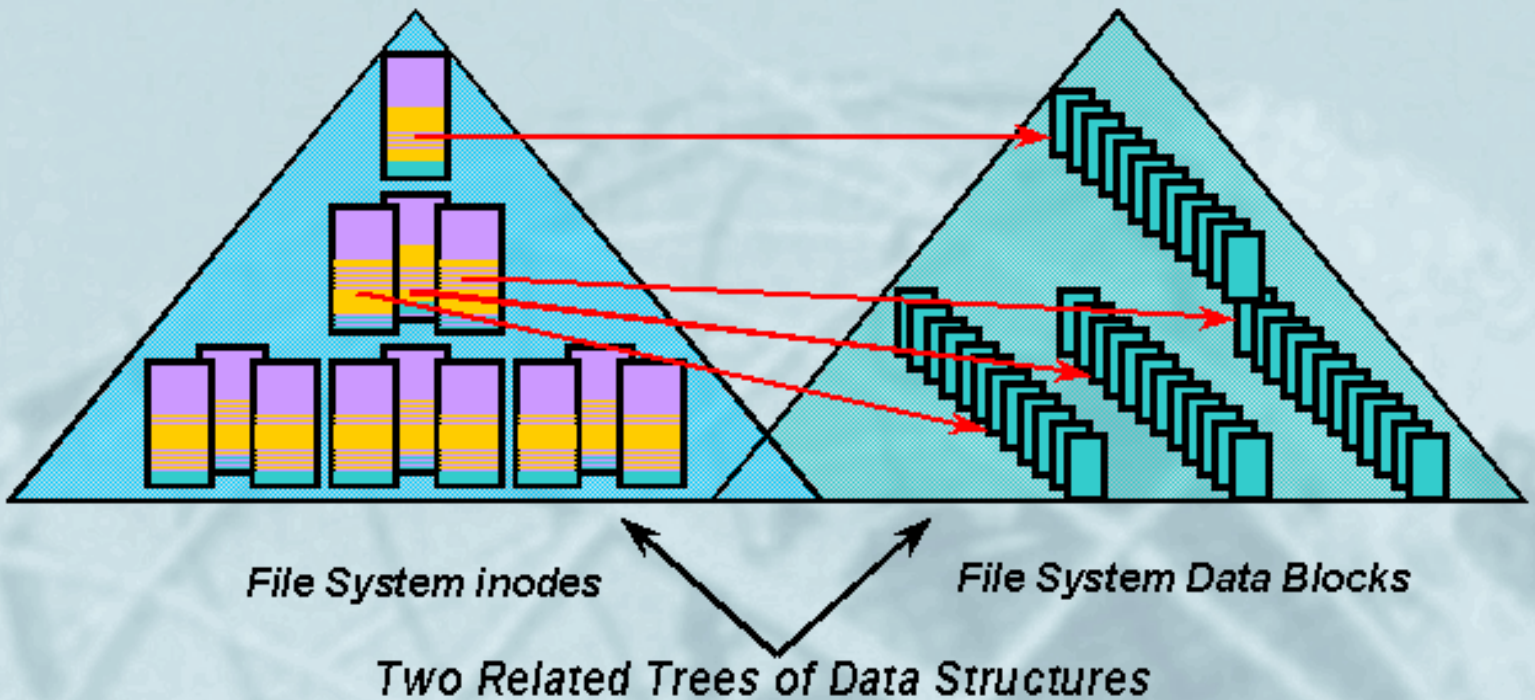


91 99/09/20 Rob Lucke

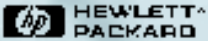


File Server Tuning

HFS File System Structure



*This will be important later, when we talk about
buffer cache and inode cache*



92 99/09/20 Rob Lucke



File Server Tuning

Walking Directory Paths

`/local_home/robl/Tools/SystemAdmin/PatchTools/LoadFilesets/README`

Each level in the path involves inode and file I/O:

Read directory file inode from disk

Read directory data from disk

...

Read file inode from disk

Read file data from disk

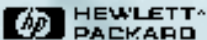
Close file: Update data and inode on disk

Each symbolic (soft) link in the path can contribute two disk I/Os:

Read soft link inode

Read soft link data

Inserting NFS into this scenario means that each operation becomes one or more RPC calls (NFS IOPS) to the remote server. Network latency and NFS server performance become the major performance components



93 99/09/20 Rob Lucke




File Server Tuning

JFS File System

- ⌘ **J**ournale**d F**ile **S**ystem (VxFS) keeps a log of structural changes being made to the file system
- ⌘ The internal structure is different from HFS
- ⌘ When system failures occur, the log is replayed to fix the file system, the repair time is dramatically shorter than running the “fsck” command for HFS
- ⌘ Logging the “meta” data impacts performance
- ⌘ **JFS** may be defragmented in place if the “Online JFS” product is purchased



 HEWLETT
PACKARD

94 99/09/20 Rob Lucke

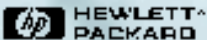
Slide 94 of 188



File Server Tuning

JFS File System

- ⌘ Veritas® recommends that the log size be increased for synchronous or NFS workloads to improve performance. This will slow down recovery time.
 - “mkfs -F vxfs logsize=n” (see man page for mkfs_vxfs)
- ⌘ The number of inodes is dynamically adjusted as files are created, unlike HFS
- ⌘ Mount options and application calls may specifically alter the performance characteristics of the file system (see man page for vxfsio)



HEWLETT
PACKARD

95 99/09/20 Rob Lucke

Slide 95 of 188



File Server Tuning

JFS mount options

log (default)

- JFS guarantees that all structural changes to the file system have been logged on the disk when the system call returns.

delaylog

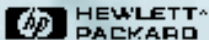
- Some system calls return before the log is written. This logging delay improves performance, but some changes are not guaranteed until a short time after the system call returns. This mode approximates the traditional UNIX file system behavior

tmplog

- Logging is almost always delayed. Only recommended for temporary file systems

~~nolog~~

- Logging is disabled. Fast recovery does not work with this mode, potential for data loss is greater than with UFS/HFS



HEWLETT
PACKARD

96 99/09/20 Rob Lucke



File Server Tuning

JFS mount options

⌘ nodatainlog

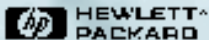
- Normally small synchronous data is put into the log (inode and write), requiring only one disk access to update. Disables that behavior: 50% slower than standard behavior.

⌘ blkclear

- Used in increased security environments to ensure that uninitialized storage never appears in files.

⌘ mincache=[closesync, direct, dsync,unbuffered,tmpcache]

- closesync synchronize file data on file close (15% slower)
- direct, dsync, unbuffered Alter the system's buffering behavior (see man page)
- tmpcache Trades less integrity for better performance (Temp file systems)





File Server Tuning

On-Line JFS operations

⌘ Examining JFS file attributes

➤ “gettext -F vxfs -D <file>”

⌘ Reserving space for a JFS file (On-Line JFS only)

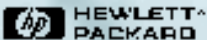
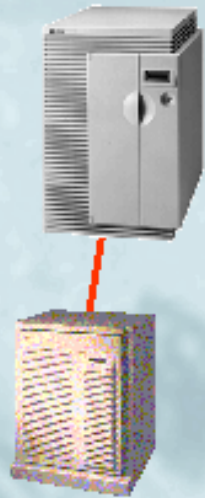
➤ “setext -F vxfs -e <extent size> -r <blks> -f contig -f trim”

⌘ Checking dir/ext fragmentation (On-Line JFS only)

➤ “fsadm -F vxfs -E -D <mount point>”

⌘ Defragmenting files, sorting/compacting dirs

➤ “fsadm -F vxfs -e -d <mount point>”



98 99/09/20 Rob Lucke



File Server Tuning

On-Line JFS operations: Examining Fragmentation

```

root@hpupora: /root_home
# fsadm -F vxfs -D -E /export/fs1

Directory Fragmentation Report
      Dirs      Total
      Searched  Blocks
-----
Total          5948      3217

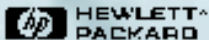
Extent Fragmentation Report
  Total      Average      Average      Total
  Files     File Blks   # Extents  Free Blks
-----
  147:             60           1    2131025

Blocks used for inodes: 620
% free blocks in extents smaller than 64 blks: 0.04
% free blocks in extents smaller than 128 blks: 0.04
% blks allocated to extents 64 blks or larger: 99.28
Free Extents By Size
  1:      633      2:      976      4:      1216      8:      1760
 16:     2373     32:     2021     64:     1445     128:     1390
 256:     329     512:     971     1024:     648     2048:     260
4096:     147     8192:     48     16384:     8     32768:     5
  
```

Directories that may be reduced in size

1 KB Blocks to reclaim

Directories that are or could be contained only in the inode data space



99 99/09/20 Rob Lucke

Source: `fsadm_vxfs(1M)` man page



File Server Tuning

On-Line JFS operations: After Disk/Directory Defragmentation

```

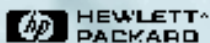
root@hpcupora: /root hcmc
# fsadm -F vxfs -D -3 /export/fs1

Directory Fragmentation Report
      Dirs      Total
      Searched  Blocks
total          5943    3417

      Imaged  Imaged  Dirs to  Blocks to
      Dirs   to Add  Reduce  Reduce
      3338   33      46      292

Extent Fragmentation Report
      Total      Average      Average      Total
      Files     File Blks   # Extents   Free Blks
      61475      65          1           4135633
blocks used for indirects: 96
% Free blocks in extents smaller than 64 blks: 0.02
% Free blocks in extents smaller than 8 blks: 0.06
% blks allocated to extents 64 blks or larger: 81.98

Free Extents By Size
      1:      344      2:      313      4:      379      8:      1861
      16:     2125     32:     1173     64:     1615     128:     1520
      256:    1293     512:    1023     1024:    677     2048:    322
      4096:   165      8192:    35     16384:    5     32768:    0
  
```



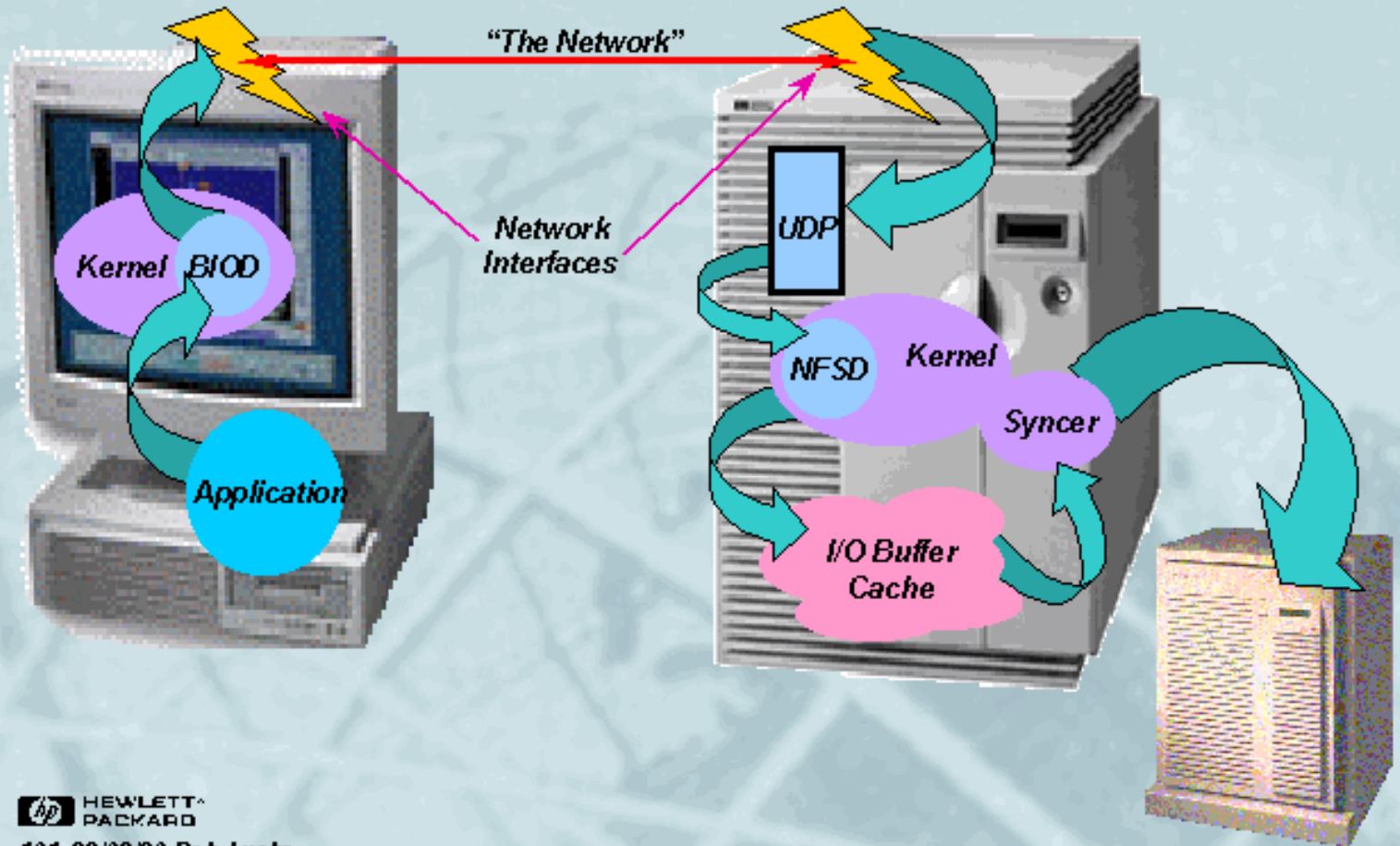
100 99/09/20 Rob Lucke

Slide 100 of 188



NFS Fileserver Tuning

NFS Client/Server Components

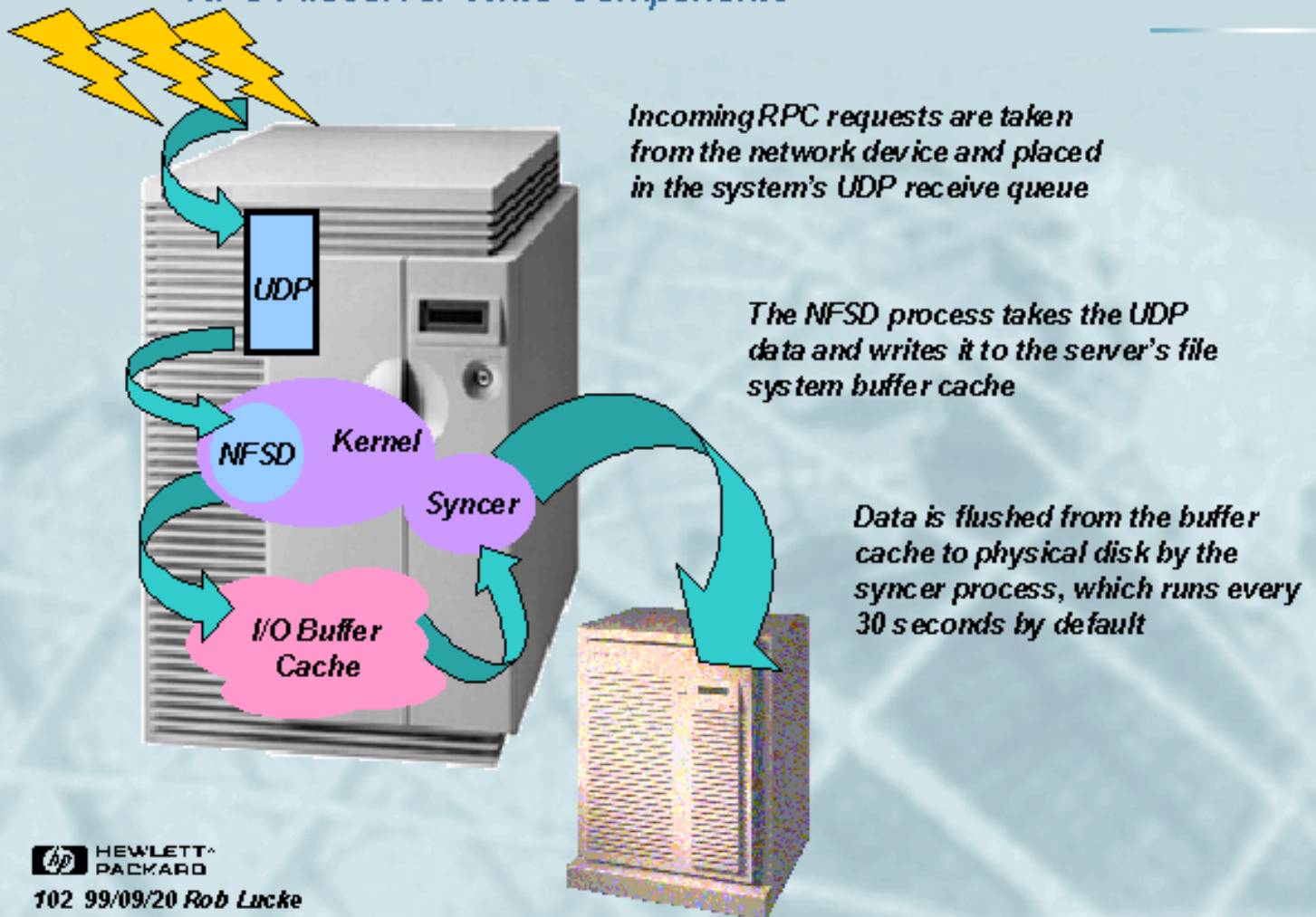


hp HEWLETT-
PACKARD
101 99/09/20 Rob Lucke



NFS Fileserver Tuning

NFS Fileserver Write Components

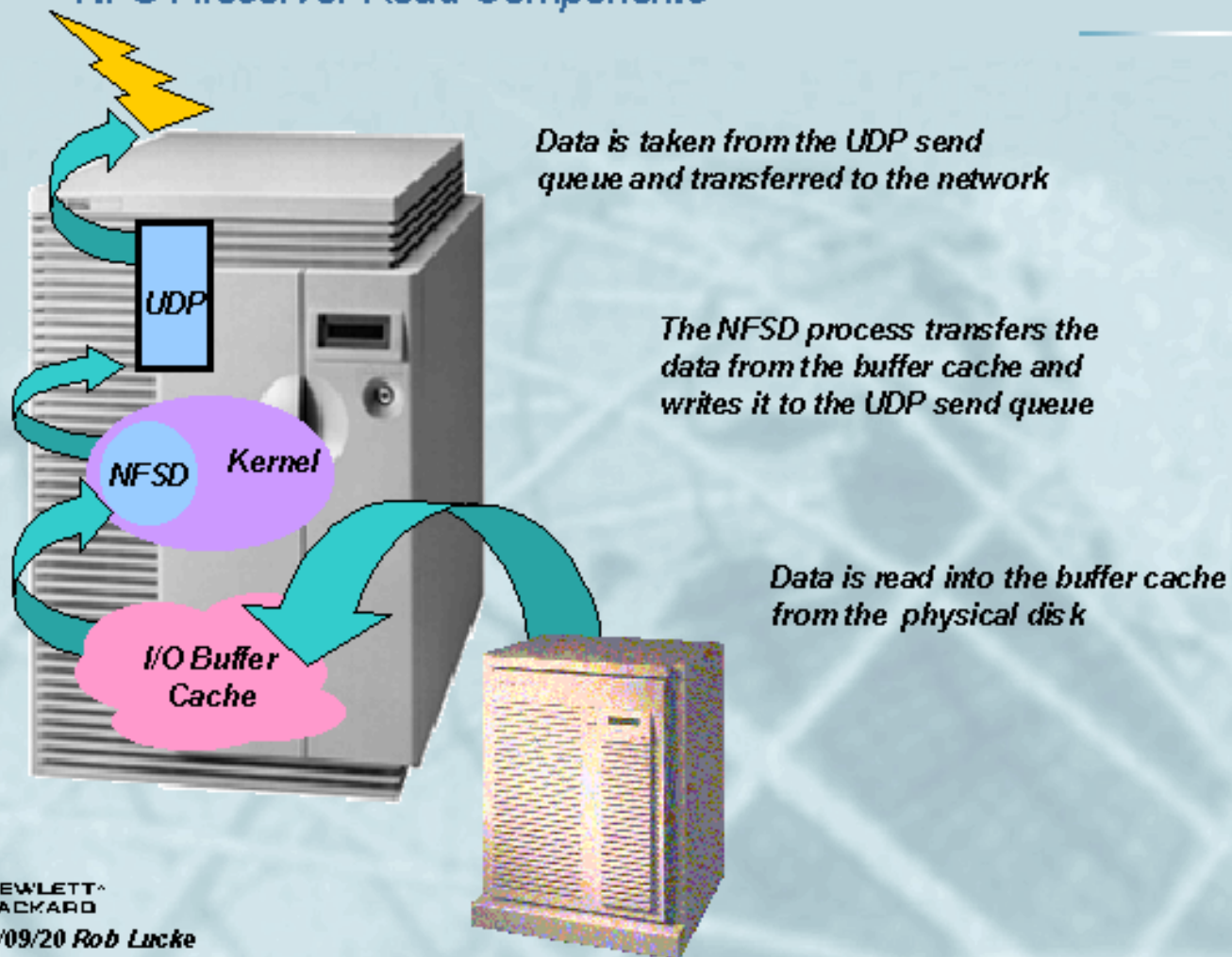


Slide 102 of 188



NFS Fileserver Tuning

NFS Fileserver Read Components





NFS Fileserver Tuning

Tuning Network Components

⌘ Fileserver LAN parameters are largely untouchable (except MTU size)



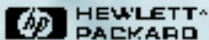
⌘ NFS RPCs arrive in the NFS UDP socket, number 2049

⌘ The UDP socket is emptied by NFSDs

⌘ The socket buffer is a FIFO, with 256 KB of buffer

⌘ Server needs enough NFSDs to keep the UDP socket “drained” (NFSDs can be a bottleneck)

⌘ Too few NFSDs will not keep the socket emptied



HEWLETT
PACKARD

104 99/09/20 Rob Lucke



NFS Fileserver Tuning

HP-UX 9.X Issues with NFSD

⌘ Starting an excessive number of NFSDs may have a negative impact on server performance



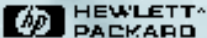
⌘ If 32 NFSDs are available, and a single NFS RPC comes into the server, then all 32 NFSDs will be place on the server's run queue



⌘ The HP-UX scheduler will context switch the first NFSD, which will service the incoming RPC, and the other 31 NFSDs will successively run, find no RPC to service, and then go back to sleep



⌘ The cost of context switching the NFSDs may limit performance over about 30 instances of NFSD



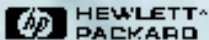
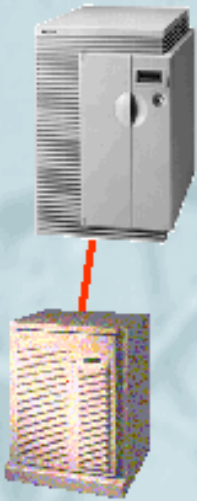
105 99/09/20 Rob Lucke



NFS Fileserver Tuning

HP-UX 10.X Issues with NFSD

- ⌘ There appears to be no cost in starting too many NFSDs on HP-UX 10.X systems (other than the small amount of process space and the proc table entries)
- ⌘ Regardless of how many NFSD processes are available, only a single NFSD is awakened for each RPC that arrives at the server
- ⌘ This allows a site to start the maximum number of NFSD processes that should be needed simultaneously, knowing that only the actual number required will be awakened
- ⌘ The V-Class NFS IOP number was obtained with 270 NFSD processes!



106 99/09/20 Rob Lucke

Slide 106 of 188



NFS Fileserver Tuning

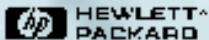
How many NFSDs Do I Start?

⌘ A common “rule of thumb” is that a server should have 2 NFSDs per disk (or logical disk). One NFSD to receive a request and move the disk head, and a second queuing up the next RPC that will move the disk head (double buffering)

⌘ In practice, there is no correlation between NFSDs and disks:

- If the server has adequate buffer cache, the file data being requested may well be in RAM and not require disk access
- If the server had an adequately sized RAM resident inode table, the inode data being requested may well be in RAM and not require disk access

⌘ It turns out that this is still a good first order approximation for a starting point



HEWLETT
PACKARD

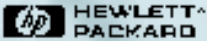
107 99/09/20 Rob Lucke



NFS Fileserver Tuning

Changing the Default Number of NFSDs

- ⌘ The variable controlling the number of NFSD processes is “NUM_NFSD”, located in the “/etc/rc.config.d/nfsconf” file
- ⌘ You can test and set the value from inside of a script with the “/usr/sbin/ch_rc” command
- ⌘ “ch_rc” treats all of the data definitions in the /etc/rc.config.d directory as a database, values may be added, tested, and changed:
 - > # ch_rc -l -p NUM_NFSD
4
 - > # ch_rc -a -p NUM_NFSD=32
- ⌘ This is a very useful command if you are configuring systems from inside of a script (as with Ignite-UX)!



108 99/09/20 Rob Lucke



Slide 108 of 188



NFS Fileserver Tuning

Measuring NFSD Usage

- ⌘ The HP-UX 9.X goal is to have just enough NFSDs to service the incoming RPCs, this is handled automatically on 10.X
- ⌘ The number of NFSDs needed can actually be measured on an HP server using the command “nfsstat -s”

◆ \$nfsstat -s

Server rpc:

9.X

```
calls      badcalls  nullrecv  badlen    xdrcall   nfsdru
2335351                                5333280
```

◆ \$nfsstat -s

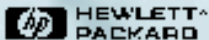
Server rpc:

10.X

```
calls      badcalls  nullrecv  badlen    xdrcall   nfsdru
2335351                                2335351
```

One to one ratio on 10.X is ideal behavior

5.3 MNFSDs started to service 2.3 MRPC calls, the number of NFSDs could be cut in half. Remember that this is an average, and more NFSDs might be needed during peaks.



109 99/09/20 Rob Lucke

Slide 109 of 188



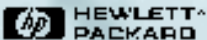
NFS Fileserver Tuning

NFS UDP Socket Depth

- ⌘ Client NFS RPC requests are received into the UDP socket on port 2049
- ⌘ The UDP socket has a 256 KB FIFO that is emptied by the NFSD processes
- ⌘ If the UDP socket continually has data in it, it is a sign that there are not enough NFSDs to keep it serviced
- ⌘ There will be times when there will be data in transit, so the average queue length is probably of more interest
- ⌘ The number of bytes in the UDP receive buffer can be displayed using the command:

```
$netstat -an | grep -e Proto -e 2049
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address (state)
udp	0	0	*.2049	*.*



110 99/09/20 Rob Lucke



NFS Fileserver Tuning

NFS UDP Socket Overflow

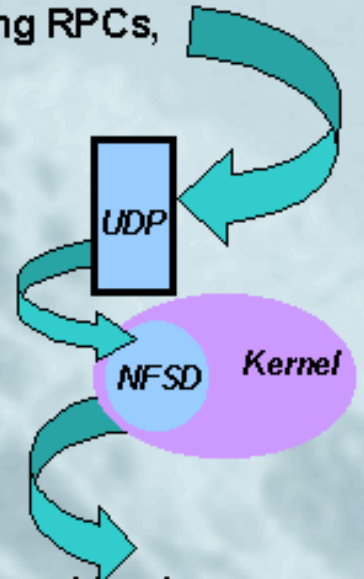
⚗ If there are not enough NFSDs to keep up with the incoming RPCs,

- ◆ The UDP socket will overflow
- ◆ NFS client requests will be dropped
- ◆ Client systems will retry

⚗ The number of times the UDP socket has overflowed can be determined using the command:

```
$ netstat -s | grep overflow
1120 socket overflows
```

⚗ This assumes that the major user of UDP services on the machine is NFS, which is not always the case



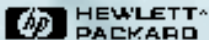
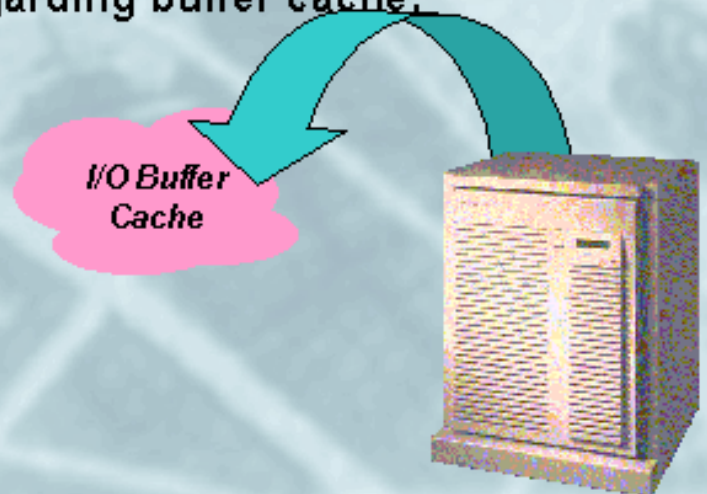


NFS Fileserver Tuning

Dedicate Buffer Cache to NFS

- ⌘ The performance bottleneck in most servers is usually the disk subsystem -- disk heads move slowly
- ⌘ Buffer cache eliminates the slow physical I/O by keeping file system blocks in RAM
- ⌘ Logical reads from buffer cache are much faster than physical reads from the disk
- ⌘ There are two important criteria regarding buffer cache:
 - ◆ Have enough of it
 - ◆ Avoid situations that will flush its contents

⌘ **DEDICATE YOUR NFS SERVER BUFFER CACHE**



112 99/09/20 Rob Lucke

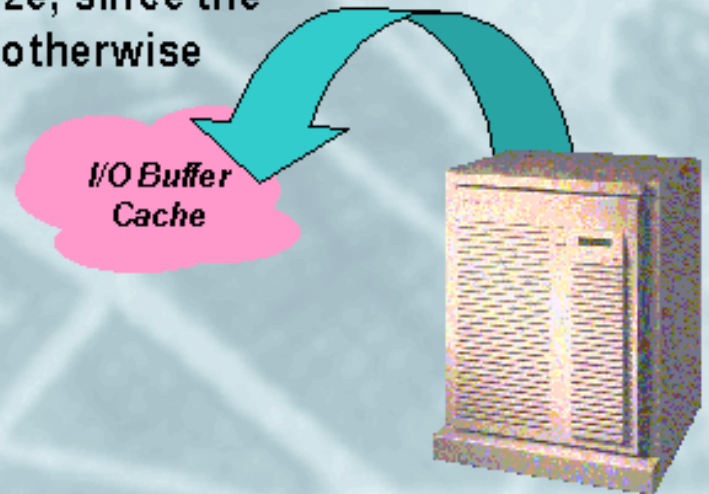
Slide 112 of 188



NFS Fileserver Tuning

Measuring Buffer Cache Working Set Size

- ⌘ The ideal file server would service all requests as fast logical reads instead of slow physical reads
- ⌘ For this to happen, there needs to be enough buffer cache to contain all of the data that the server's clients are using
- ⌘ This data is the "working set"
- ⌘ For most situations, the buffer cache should be sized to handle 85-95% of the working set size, since the required cache might be very large otherwise
- ⌘ For existing servers, the buffer cache hit rate is reported by GlancePlus and can be used to adjust the size of the buffer cache





NFS Fileserver Tuning

Measuring Buffer Cache Working Set Size

File Reports									
System:	hpux00		Last Update:	14:55:55		Ints:	11 sec		
Req Type	Requests	%	Rate	Bytes	Cum Req	%	Cum Rate	Cum Bytes	
Local Lopl Reads	23	92.0%	2.1	3kb	1371877	93.5%	5.2	2.55gb	
Local Lopl Writes	2	8.0%	0.1	0kb	791770	6.5%	0.3	2.19mb	
Local Phys Reads	0	0.0%	0.0	0kb	1614441	45.2%	0.7	1.47rb	
Local Phys Writes	12	100.0%	1.1	40kb	1960809	54.8%	0.9	2.45gb	
Local User	8	66.7%	0.7	8kb	2730453	76.4%	1.2	3.35gb	
Local Virtual Mem	0	0.0%	0.0	0kb	209791	5.9%	0.0	1.51rb	
Local System	4	33.3%	0.3	32kb	618514	17.3%	0.2	3.00gb	
Local Raw	0	0.0%	0.0	0kb	16492	0.5%	0.0	62.4mb	
Remote Lopl Reads	3	75.0%	0.2	1kb	1189647	91.5%	0.5	3.81rb	
Remote Lopl Writes	1	25.0%	0.0	0kb	110641	8.5%	0.0	463.4mb	
Remote Phys Reads	38	95.0%	3.6	0kb	1587115	92.1%	0.7	3.66rb	
Remote Phys Writes	2	5.0%	0.1	0kb	135493	7.9%	0.0	733.9mb	
Remote User	0	0.0%	0.0	0kb	574136	33.3%	0.2	389.4mb	
Remote Virtual Mem	0	0.0%	0.0	0kb	111	0.0%	0.0	864kb	
Remote System	40	100.0%	3.8	0kb	1148361	66.7%	0.5	0kb	
Remote Raw	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Event	Current	Cumulative	Curr %	Avg %	High %				
Read Cache Hits	1	19160528	100.0	99.2	100.0				
Write Cache Hits	8	2290132	50.0	78.4					
DNLC Hits	202	54751859	91.8	93.7	100.0				
DNLC Longs	0	84	0.0	0.0	5.1				

Read cache hit rates in the high 90s is a sign of a server that is doing a good job caching the client's working set

HP PACKARD

114 99/09/20 Rob Lucke

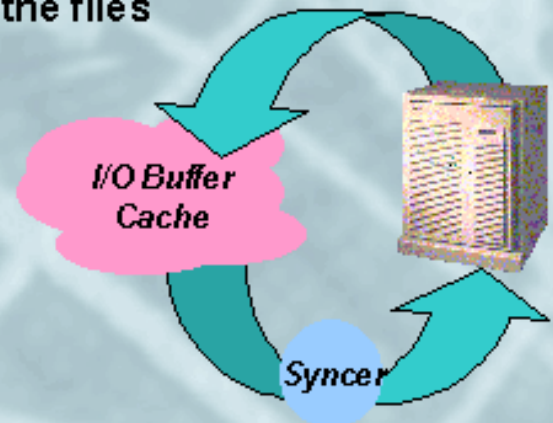
Source: GPM>Reports>Disk Info>Disk Report



NFS Fileserver Tuning

Initially Sizing Buffer Cache

- ⌘ A popular approximation is 128 KB of buffer cache per NFS IOP the server will provide
- ⌘ Note that HP-UX 9.X is limited to approximately 800 MB of buffer cache, this limitation was removed in 10.X
- ⌘ It may be impossible to keep the buffer cache hit rate as high as might be desired, in that case:
 - ◆ Consider splitting the NFS server and the files it provides into two machines
 - ◆ Make sure the disk I/O system is capable of sustaining heavy I/O rates by selecting the proper interfaces and striping the data





NFS Fileserver Tuning

NFS File Server Resource Cortention

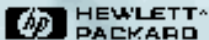
⌘ The easiest way to impact ALL users in an environment is to slow down the NFS file server with CPU or I/O intensive activities

⌘ **DO NOT RUN APPLICATIONS ON THE NFS SERVER**

⌘ The exception to this rule are “well-behaved” client-server applications that cause minimal impact, for example:

- ◆ NIS
- ◆ DNS
- ◆ X11 Font Server

⌘ If sharing NFS file server resources with other activities is unavoidable, then consider using **Process Resource Manager** to guarantee resource levels



116 99/09/20 Rob Lucke

Slide 116 of 188



NFS Fileserver Tuning

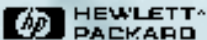
Disk Caching

- ⌘ NFS clients halt processing until the NFS server acknowledges that write requests are posted to the disk
- ⌘ HP-UX supports disks with immediate reporting mode (on-controller cache) and RAID arrays that contain write cache
 - ◆ When NFS data is written from buffer cache to the disk, the disk immediately reports the data has been written, even though it hasn't
 - ◆ The server then acknowledges the write to the NFS client
 - ◆ `# scsictl -a /dev/rdisk/c0t6d0`
`immediate_report = 1 ; queue_depth = 8` *10.20 Series 700*
 - ◆ `# scsictl -a /dev/rdisk/c0t10d0`
`immediate_report = 0 ; queue_depth = 2` *10.20 Series 800*

⌘ Benefits: speeds disk requests, disk may do more ops/sec

⌘ Problems: data may be lost on power failure

⌘ **SYSTEMS MUST HAVE UPS FOR SYSTEM AND DISKS TO USE THIS FEATURE OR DATA WILL BE LOST**



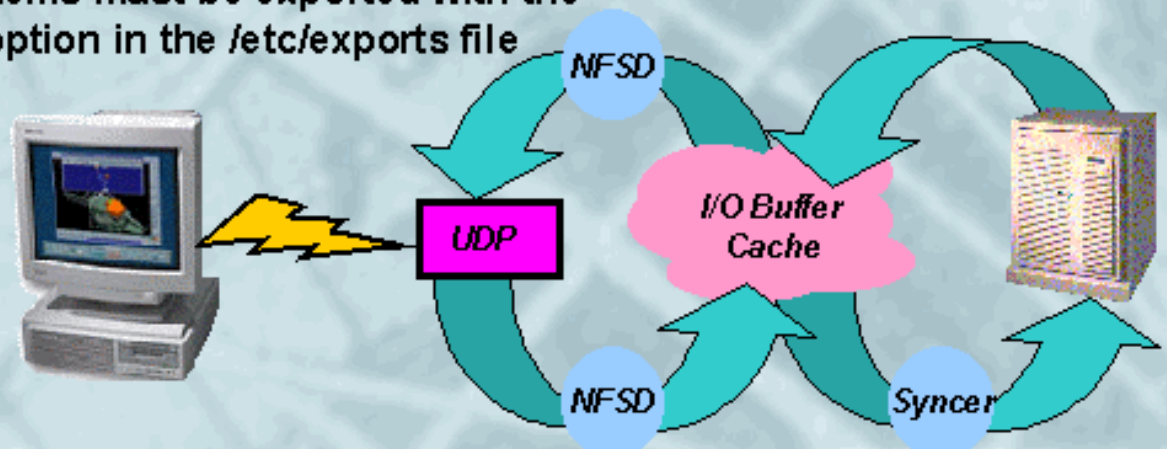
117 99/09/20 Rob Lucke




NFS Fileserver Tuning

Asynchronous NFS Mounts

- ⌘ A client request must go through buffer cache, and usually a disk queue, make it to the disk's cache before the server can acknowledge the client request
- ⌘ HP-UX supports asynchronous NFS write to cut this write path even shorter
 - ◆ Prior to placing client data in buffer cache, the HP-UX NFSD immediately reports the data as written, releasing the client much sooner in the write path
 - ◆ File systems must be exported with the `-async` option in the `/etc/exports` file



 HEWLETT
PACKARD

118 99/09/20 Rob Lucke

Slide 118 of 188



NFS Fileserver Tuning

Asynchronous NFS Mounts

<i>K400-1 NFS Writes</i>	<i>NFS IOPS</i>	<i>Server Response Time</i>
<i>Synchronous</i>	1956	46.9 ms
<i>Asynchronous</i>	2012	13.5

Benefits of NFS Asynchronous Mounts

- ◆ NFS client continues processing as soon as NFSD has data
- ◆ More NFS IOPS from server
- ◆ Improved client response time



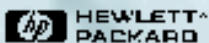
Solution is to use battery backup for the RAM containing asynchronous NFS data

HP-UX systems support UPS systems for CPU, RAM, and disks

Note that NFS PV3 has safe asynchronous writes

Risks of NFS Asynchronous Mounts

- ◆ Data is reported to the client as being committed when it isn't and is susceptible to loss



119 99/09/20 Rob Lucke

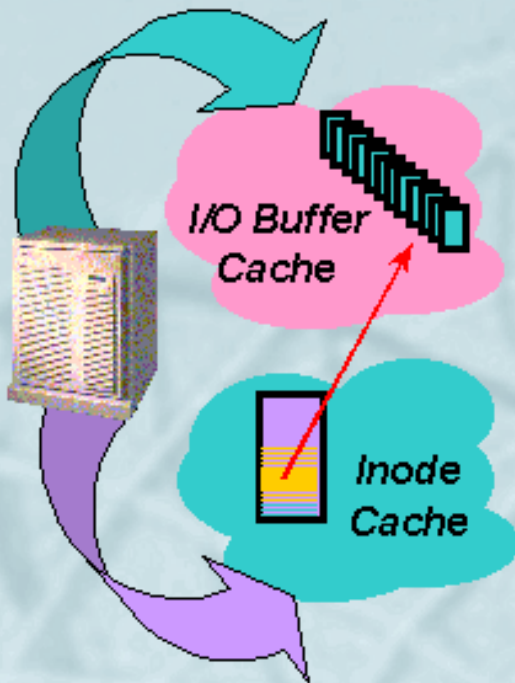



NFS Fileserver Tuning

Tunable Kernel Parameters: HFS inode cache

γ *ninode* (inode cache size)

- To access a file's data, the server must first bring the file's inode into the memory resident inode table entry, which is about 240 bytes
- The inode cache can help prevent reading or re-reading an inode whose data is already in buffer cache
- The average number of memory resident inodes required for optimum performance will vary depending on the average working set size
- As a general rule of thumb, there should be 1 to 2 times as many inodes as there are 8 KB blocks in the buffer cache
- Every 128 MB of buffer cache is 16,000 inode table entries using this rule
- Careful on multi-CPU machines with 10.X, there is a potential spinlock contention for the inode table



 HEWLETT
PACKARD

120 99/09/20 Rob Lucke

Slide 120 of 188



NFS Fileserver Tuning

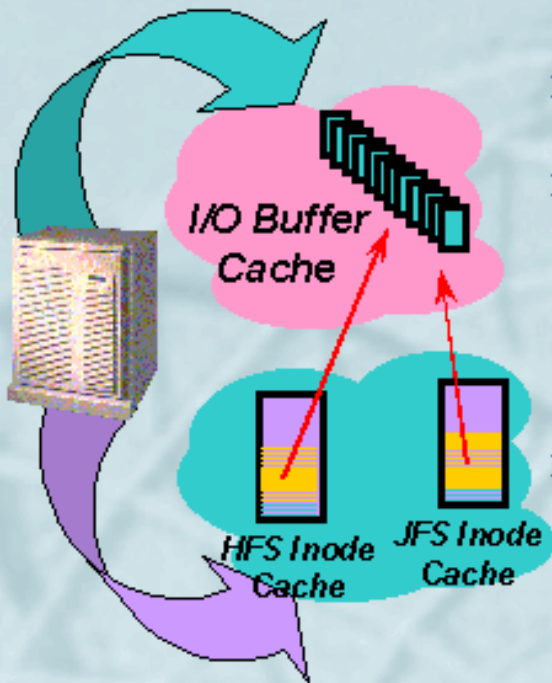
Tunable Kernel Parameters: JFS inode cache

∞ ninode (inode cache size)

- The HFS and JFS inode caches are separate entities (ninode does not directly affect the size for JFS)
- JFS prior to HP-UX 10.20 determines the size of the JFS inode cache based on memory size
- However, if the value for ninode is larger than the table value, then the value for ninode is used as the minimum size of the JFS inode cache also
- HP-UX 11.X introduced the "vx_ninode" parameter for explicitly adjusting the size of the JFS inode cache

Memory Size *JFS Inodes*

8 MB	400
16 MB	1000
32 MB	2500
64 MB	6000
128 MB	8000
256 MB	16,000
512 MB	32,000
1 GB	64,000
> 2 GB	128,000





NFS Fileserver Tuning

Tunable Kernel Parameters: ninode

Dynamic

Name

Lookup

Cache keeps frequently accessed path

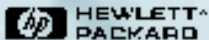
components and their inode table location in memory. The DNLC

cache size is tied to "ninode" and may not be independently set

unless a patch is installed that

introduces the "ncsize" kernel parameter. Path components longer than 15 characters are not cached.

File Reports						Help		
System: hpux0nc						Last Update: 16:25:28	Int: 10 sec	[?]
System Table	Avail	Used	Util %	High				
Proc Table (nproc)	900	236	26%	39%				
File Table (nfile)	1798	602	33%	38%				
Shared Mem Table (shmmni)	200	0	5%	5%				
Message Table (msgmni)	50	2	4%	4%				
Semaphore Table (semnmi)	64	18	28%	28%				
File Locks (nfilelocks)	200	20	13%	17%				
Pseudo Terminals (npty)	60	0	15%	15%				
Buffer Headers (nbuf)	na	14400	na	na				
System Table	Avail	Used	High					
Shared Memory	12.5gb	10.1mb	na	na				
Message Buffers	16kb	0kb	0kb	0kb				
Inode Cache (ninode)	5000	019	5000					
DNLC Cache	5000							
	Min	Max	Avail	Used	High			
Buffer Cache	12.8mb	128.0mb	84.6mb	84.6mb	85.7mb			
	Avail Size	Used Size	Reserved Size	Util %				
Swap Space	684mb	170mb	280mb	42%				



122 99/09/20 Rob Lucke

Source: GPM>Reports>System Info>System Tables Report

Slide 122 of 188



NFS Fileserver Tuning

DNLC Cache Hit rate

Disk Info - Disk Report									
File Reports									
System: hpux000 Last Update: 14:55:00 Int: 11 sec									
Req Type	Requests	%	Rate	Bytes	Cum Req	%	Cum Rate	Cum Bytes	
Local Lopl Reads	23	92.0%	2.1	3kb	1071877	93.5%	5.2	2.55gb	
Local Lopl Writes	2	8.0%	0.1	0kb	791770	6.5%	0.3	2.15rb	
Local Phys Reads	0	0.0%	0.0	0kb	1014441	45.2%	0.7	1.47rb	
Local Phys Writes	12	100.0%	1.1	40kb	1960809	54.8%	0.9	2.45gb	
Local User	8	66.7%	0.7	8kb	2700450	76.4%	1.2	0.05gb	
Local Virtual Mem	0	0.0%	0.0	0kb	209791	5.9%	0.0	1.51rb	
Local System	4	33.3%	0.3	32kb	618514	17.3%	0.2	0.00gb	
Local Raw	0	0.0%	0.0	0kb	16492	0.5%	0.0	02.4mb	
Remote Lopl Reads	3	75.0%	0.2	1kb	1189647	91.5%	0.5	3.81rb	
Remote Lopl Writes	1	25.0%	0.0	0kb	110641	8.5%	0.0	403.4mb	
Remote Phys Reads	38	95.0%	3.6	0kb	1587115	92.1%	0.7	3.66rb	
Remote Phys Writes	2	5.0%	0.1	0kb	105490	7.9%	0.0	700.9mb	
Remote User	0	0.0%	0.0	0kb	574106	33.3%	0.2	089.4mb	
Remote Virtual Mem	0	0.0%	0.0	0kb	111	0.0%	0.0	804kb	
Remote System	40	100.0%	3.8	0kb	1148061	66.7%	0.5	0kb	
Remote Raw	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Event	Current	Cumulative	Curr %	Avg %	High %				
Read Cache Hits	1	19160528	100.0	99.2	100.0				
Write Cache Hits	8	2290102	50.0	78.4					
DNLC Hits	202	54751959	91.8	93.7	100.0				
DNLC Longs	0	84	0.0	0.0	5.1				

DNLC hits indicates the cumulative number of path components that were found in the DNLC cache. DNLC longs indicates the number of components that were not found because they were longer than 15 characters



123 99/09/20 Rob Lucke

Source: GPM>Reports>Disk Info>Disk Report



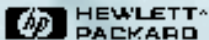
NFS Fileserver Tuning

Tunable Kernel Parameters: Buffer Cache

Kernel Parameter	Description	Values	Values	Values
<i>nbuf</i>	Number of 8 KB file system blocks in buffer cache	0	N	0
<i>bufpages</i>	Number of 4 KB physical pages in buffer cache	0	0	M
<i>dbc_min_pct</i>	Minimum percentage of physical RAM for buffer cache	5	X	X
<i>dbc_max_pct</i>	Maximum percentage of physical RAM for buffer cache	50	X	X

On 10.X Series 700 and 800, if both *nbuf* and *bufpages* are set to 0, then dynamic buffer cache is enabled (default). On 9.X Series 800, a fixed buffer cache of 10% of physical RAM is allocated in this case.

If *nbuf* is set, *bufpages* is calculated. If *bufpages* is set, *nbuf* is calculated. If both are set, then *nbuf* "wins." In either case, a fixed buffer cache is used



124 99/09/20 Rob Lucke

Slide 124 of 188

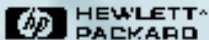
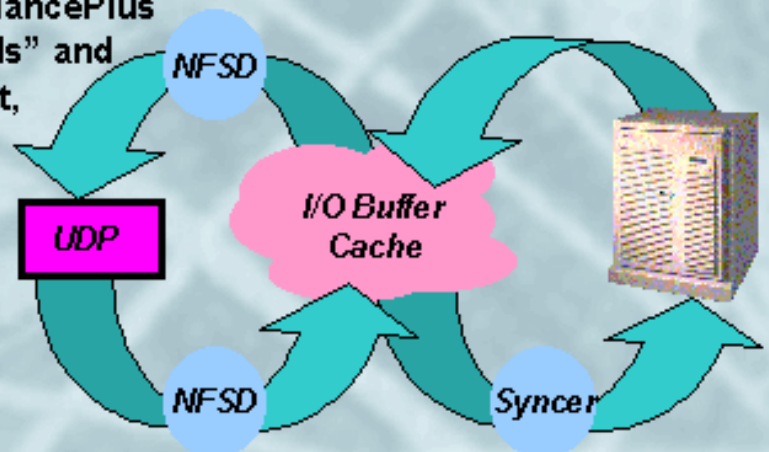


NFS Fileserver Tuning

Tunable Kernel Parameters: Fixed Buffer Cache on 9.X

⌘ If using 128 KB per NFS IOP (or 128 MB per 1000 NFS IOP)

- ◆ Use “bufpages” to set the buffer cache size if it is static
- ◆ Increment “bufpages” by 32,768 for every 128 MB increment
- ◆ Remember that on 9.X there is a practical limit of 800 MB of buffer cache
- ◆ Monitor buffer cache hit rate with GlancePlus look for high 90% range
- ◆ Do not attempt to correlate the GlancePlus “Cache Hits” with “Physical Reads” and “Logical Reads” in the disk report, as NFS code bypasses the read and write system calls that GlancePlus instruments.



125 99/09/20 Rob Lucke

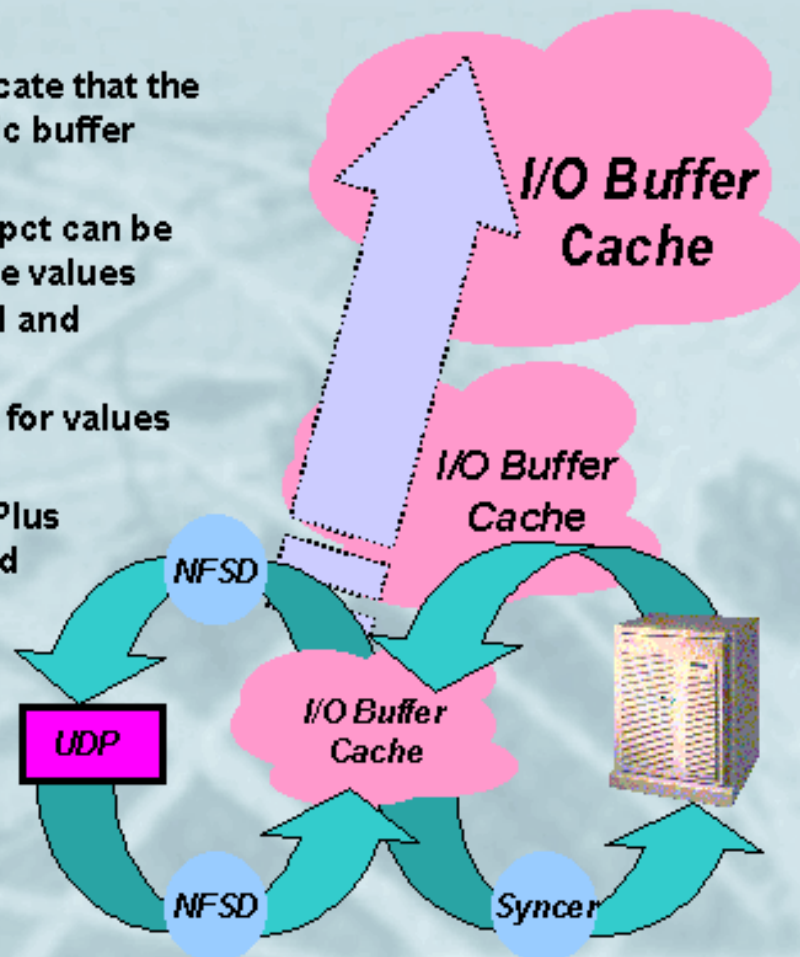
Slide 125 of 188



NFS Fileserver Tuning

Tunable Kernel Parameters: Dynamic Buffer Cache

- The SPEC_SFS_1 benchmark data indicate that the system performs better using a dynamic buffer cache than a fixed one
- Values for `dbc_min_pct` and `dbc_max_pct` can be 70% and 90% respectively, unless those values conflict with RAM needed for the kernel and applications like GlancePlus, NIS, etc.
- Monitor buffer cache hit rates and look for values in the high 90% range
- Do not attempt to correlate the GlancePlus "Cache Hits" with "Physical Reads" and "Logical Reads" in the disk report, as NFS code bypasses the read and write system calls that GlancePlus instruments

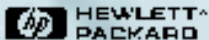
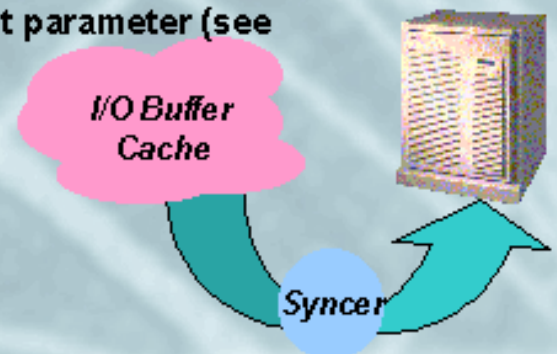




NFS Fileserver Tuning

The "syncer" Process

- _ The syncer is responsible for flushing "dirty" system data to the disk
- _ The syncer runs at 30 second intervals (default) and flushes super blocks, modified inodes (inode cache), and delayed block I/O (buffer cache) to disk
- _ The buffer cache is split into five separate areas by the syncer algorithm and each is serviced during the run interval)
- _ On a "normal" system this activity is all but invisible, but on an NFS file server, the syncer can cause visible pauses in the system activity due to the large size of the caches involved
- _ This is because it attempts to flush all "dirty" data to disk at once and runs at a relatively high priority
- _ The syncer interval can be adjusted by modifying the startup line in `/sbin/init.d/syncer` to add the new interval as the first parameter (see man page for syncer)
- _ Running the syncer more frequently can eliminate long system pauses on an NFS server at the cost of additional CPU cycles



127 99/09/20 Rob Lucke



NFS Fileserver Tuning

Tunable Kernel Parameters: Other Kernel Parameters



nfile (maximum open files)

- Estimate with $20 * \text{number_of_clients}$, then monitor with GlancePlus



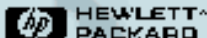
nflocks (maximum file locks)

- Estimate with $5 * \text{number_of_clients}$ then monitor with GlancePlus



nproc (maximum processes)

- Set to the maximum number of processes that can execute on the server at one time. The largest contributor might be the NFSD processes



128 99/09/20 Rob Lucke

Slide 128 of 188



NFS Fileserver Tuning

Tunable Kernel Parameters: Other Kernel Parameters



create_fastlinks

- ◆ Enables the placement of soft link text into unused space in the inode entry if set to "1"
- ◆ The link text must be less than 60 bytes
- ◆ Eliminates a logical/physical disk read for every link evaluation
- ◆ File systems with this type of link cannot be mounted on systems that do not have the fast links enabled



default_disk_ir

- ◆ Default value for disk immediate reporting
- ◆ Set to "1" to enable immediate reporting
- ◆ Ensure UPS protection where data is critical



129 99/09/20 Rob Lucke

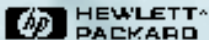
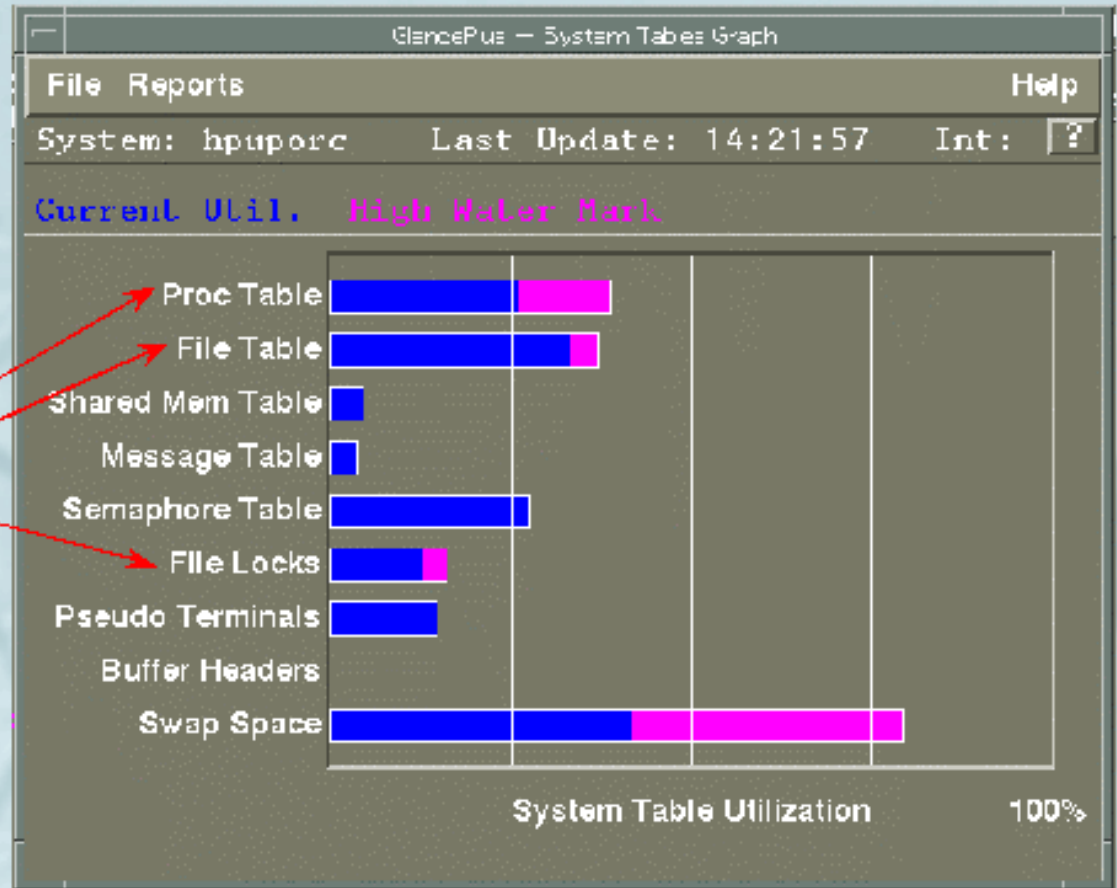
Slide 129 of 188



NFS Fileserver Tuning

Tunable Kernel Parameters: Monitoring Results

Monitor the current utilization of the important tables on the server, and note the historical "high water" marks for tuning purposes.



130 99/09/20 Rob Locke

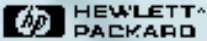
Source: GPM->Reports->System Info->System Tables Graph



NFS Fileserver Tuning

Controlling Client Access: Exporting File Systems

- ⌘ Export only the file systems that clients are accessing, avoid “exporting the world”
- ⌘ `/etc/exports` can control which clients can access data on the file server and how they can access it:
 - ◆ `ro`
 - ◆ `-rw=hostname[:hostname] ...` *(read-only for unspecified)*
 - ◆ `-access=client[:client] ...`
 - ◆ `-root=hostname[:hostname] ...`
 - ◆ `-anon=UID` *(default is -2, -1 disables)*
 - ◆ `-async`
- ⌘ Form at for this file is “directory -option,option...”

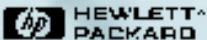




NFS Fileserver Tuning

The server's NFS mount daemon

- ⌘ The current HP-UX mountd process is single-threaded
 - The NFS lab is working on a new multi-threaded daemon for late 1999
- ⌘ The server's mount daemon must resolve client name information for each mount request
- ⌘ Multi-threaded AutoFS on the client systems may make the server response situation worse due to increased client request rates
- ⌘ If the name resolution service is slow (i.e. NIS) a performance bottleneck may result
 - Alter nfsswitch.conf to use DNS and files for name resolution
 - Consider making the NFS server a caching-only DNS server using itself for name resolution



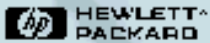
132 99/09/20 Rob Lucke

Slide 132 of 188



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning**
- ⌘ NFS PV3

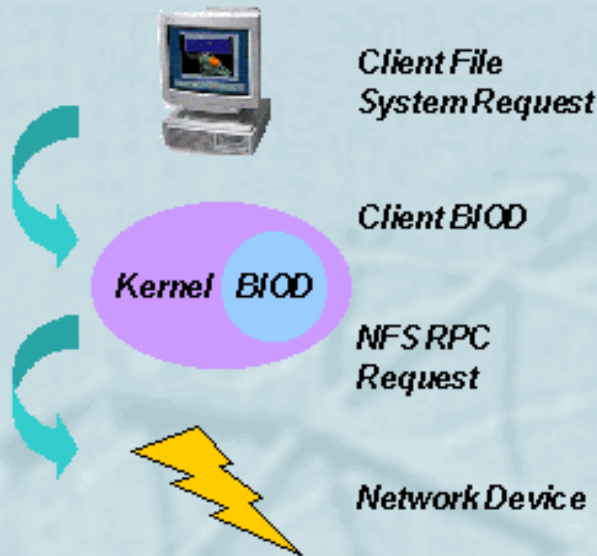


133 99/09/20 Rob Lucke



NFS Client Tuning

NFS Client Components



WRITES

- When an application performs a write to an NFS file, the kernel invokes the NFS “BIOD” (Block I/O Daemon)
- The BIOD blocks up the write requests into the block size specified at mount time (`wsize=8192`) for a “write behind”, i.e. the write is delayed until a full 8 KB can be written to the server

READS

- When an application performs a read from an NFS file, the kernel invokes the NFS BIOD.
- The BIOD reads the required block (`rsize=8192`) and does a “read ahead” of the next 8 KB block

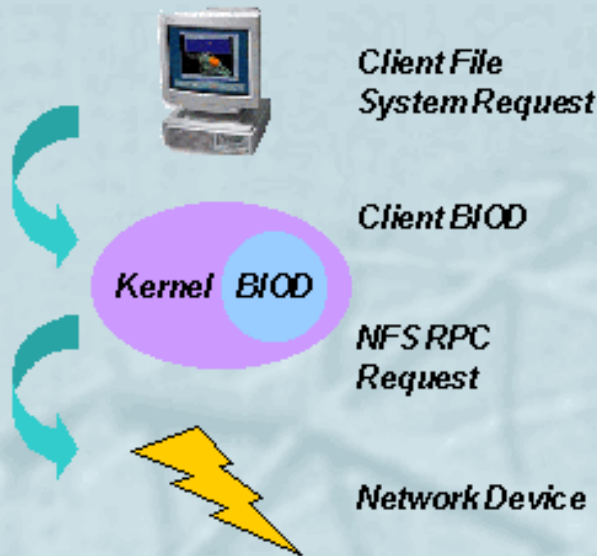
How Many BIODs?

- If an attempt is made to read or write to an NFS file and the kernel finds all BIODs busy, then the read or write will cause an immediate NFS RPC to the server, increasing server load and network traffic
- The HP-UX default, 4 BIODs is too small, try 16 and monitor with GlancePlus



NFS Client Tuning

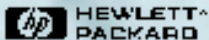
NFS Client 8 KB Read/Write Size



Client Read and Write Size

- Assumes wsize=8192 and rsize=8192
- Determined by client NFS mount options
- If the NFS server file systems are configured for 8 KB blocks and 1 KB fragments and the client mounts the server with 8 KB read and write sizes, then a client write of 8 KB of data may cause 2 to 3 server disk writes:
 - ◆ 8 KB of data to the server's disk block
 - ◆ The file's inode
 - ◆ For large files, potentially an indirect inode

- For best performance, client read/write block size should match the server's disk block size



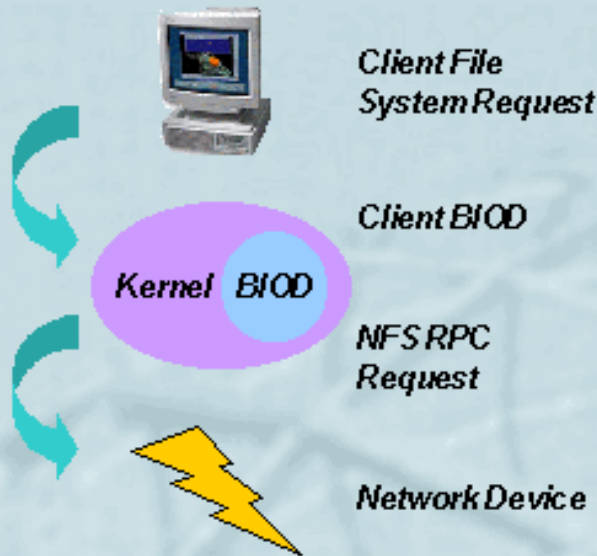
135 99/09/20 Rob Lucke

Slide 135 of 188



NFS Client Tuning

NFS Client 1 KB Read/Write Size



Client Read and Write Size

Assumes wsize=1024 and rsize=1024

Determined by client NFS mount options

- If the NFS server file systems are configured for 8 KB blocks and 1 KB fragments and the client mounts the server with 1 KB read and write sizes, then a client write of 1 KB of data may cause over 30 server disk writes:

- ◆ Write of the first 1 KB fragment followed by the file's inode, potentially followed by an indirect inode
- ◆ To write the second 1 KB, the server reads the first 1 KB fragment, combines the two, writes the 2 KB to two adjacent fragments, followed by the file's inode and potentially the indirect inode
- ◆ Continues until a full 8 KB block is written




NFS Client Tuning

Changing the Default Number of BIODs

- ⌘ The variable controlling the number of BIOD processes is “NUM_NFSIOD”, located in the “/etc/rc.config.d/nfsconf” file
- ⌘ You can test and set the value from inside of a script with the “/usr/sbin/ch_rc” command
- ⌘ “ch_rc” treats all of the data definitions in the /etc/rc.config.d directory as a database, values may be added, tested, and changed:
 - > # ch_rc -l -p NUM_NFSIOD
4
 - > # ch_rc -a -p NUM_NFSIOD=16
- ⌘ This is a very useful command if you are configuring systems from inside of a script (as with Ignite-UX)!



 HEWLETT
PACKARD

137 99/09/20 Rob Lucke

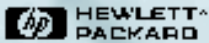
Slide 137 of 188



NFS Client Tuning

Number of Idle BIOD Processes

Glenn@File - NFS Global Activity				
File Reports				Help
System: hpupora	Last Update: 15:01:37		Int: 10 sec	
Activity	Server (Inbound)		Client (Outbound)	
	Current	Cum	Current	Cum
Read Rate	1.4	1.5	0.0	0.0
Write Rate	0.0	0.3	0.0	0.0
Read Byte Rate	0.0	0.0	0.0	0.0
Write Byte Rate	0.0	0.0	0.0	0.0
NFS IOs	14	2650	0	0
NFS Calls	0	576	0	576
Bad Calls	0	0	0	0
Service Time	0.00	3.17	0.00	0.00
Network Time			0.00	0.00
Read/Write Qlen			0.00	0.00
Idle biods			0	



138 99/09/20 Rob Lucke

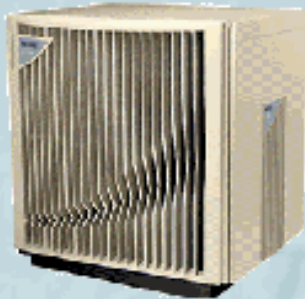
Source: GPM-> Reports-> Network Info-> NFS Global Activity

Slide 138 of 188



NFS Client Tuning


Tunable Kernel Parameters



page_text_to_local

- ◆ Set to “1” to enable paging of code pages to local client swap area, instead of reloading them across the network
- ◆ The executable file must have the “sticky” bit set on the NFS server
- ◆ “chmod 1555 <executable file>”
- ◆ Can reduce network traffic, especially if the client is running large applications or is short on RAM
- ◆ Will increase the use of the local disk, so multiple swap areas on multiple devices are a real good idea



 HEWLETT
PACKARD

139 99/09/20 Rob Lucke

Slide 139 of 188



NFS Client Tuning

Tunable Kernel Parameters




ninode

- ◆ Clients can usually benefit from a larger inode table, to make LOCAL disk access more efficient
- ◆ There is no connection between the inode cache on the client

nbuf, bufpages, dbc_min_pct, dbc_max_pct



- ◆ Behavior is the same for dynamic buffer cache on the client as was previously described
- ◆ NFS PV2 does not appear to use client-side buffer cache, *but NFS PV3 can use up to 25% of the client's buffer cache*
- ◆ *The overall effect of NFS PV3 on client buffer cache is unknown at this time*

 HEWLETT
PACKARD

140 99/09/20 Rob Lucke

Slide 140 of 188



NFS Client Tuning

NFS Client Mount Parameters: BIOD buffer size

rsize=n, wsize=n

- ◆ Set read/write buffer size to N bytes for BIODs
- ◆ Default is 8192 (8KB)
- ◆ Remember that mismatches between client system buffer size and file server block size can cause extra physical I/Os and poor performance in addition to the extra RPC requests and network loading
- ◆ Changes are most frequently seen with customers that are:
 - Trying to run NFS over a WAN (!)
 - Stuck with networking equipment that will not do proper MTU translation (i.e. FDDI ~4500 bytes -> Ethernet ~1500 bytes)
 - Have slow routers in the network that drop packets
 - Using PCNFS and have older interface cards and NFS client software for Windows





NFS Client Tuning

NFS Client Mount Parameters: Timeout Behavior

retrans=n

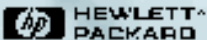
- ◆ Set the number of minor NFS timeouts per major NFS timeout
- ◆ Default is 4

tim eo=n

- ◆ Set the initial number of 0.1 second intervals in an NFS minor timeout
- ◆ Default is 7 (0.70 seconds)



These two parameters, if mismatched against the server and network performance, can severely impact the performance of an NFS client. Together they implement an exponential back-off algorithm that attempts to correct for UDP's unreliable nature. Performance can be exponentially bad if they are not set properly!



142 99/09/20 Rob Lucke

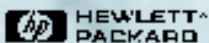
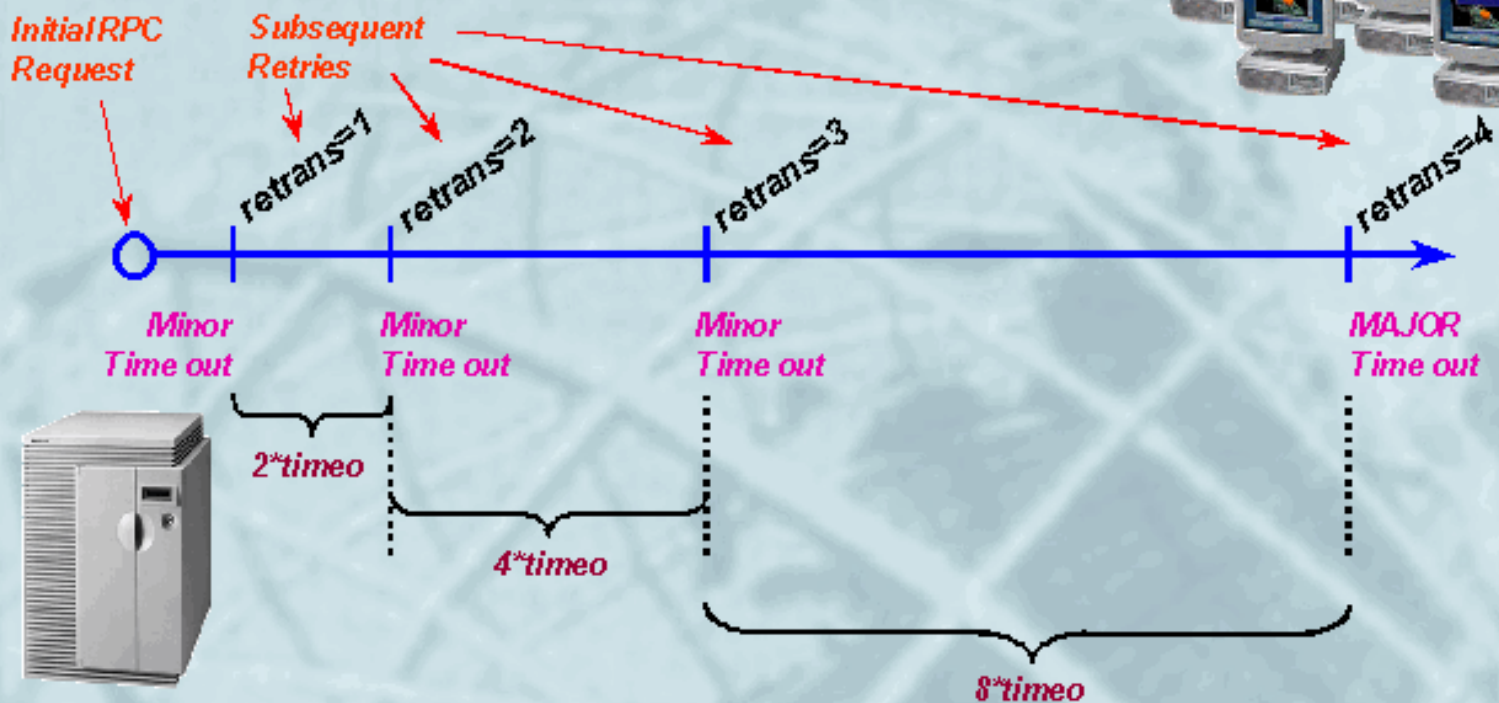
Slide 142 of 188



NFS Client Tuning

NFS Client Mount Parameters: Timeout Behavior Example

retrans=4, tim eo=7



143 99/09/20 Rob Lucke



NFS Client Tuning

NFS Client Mount Parameters: Timeout Behavior Example

retrans=4, tim eo=7



*File Server or
Network Load*





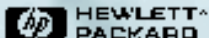
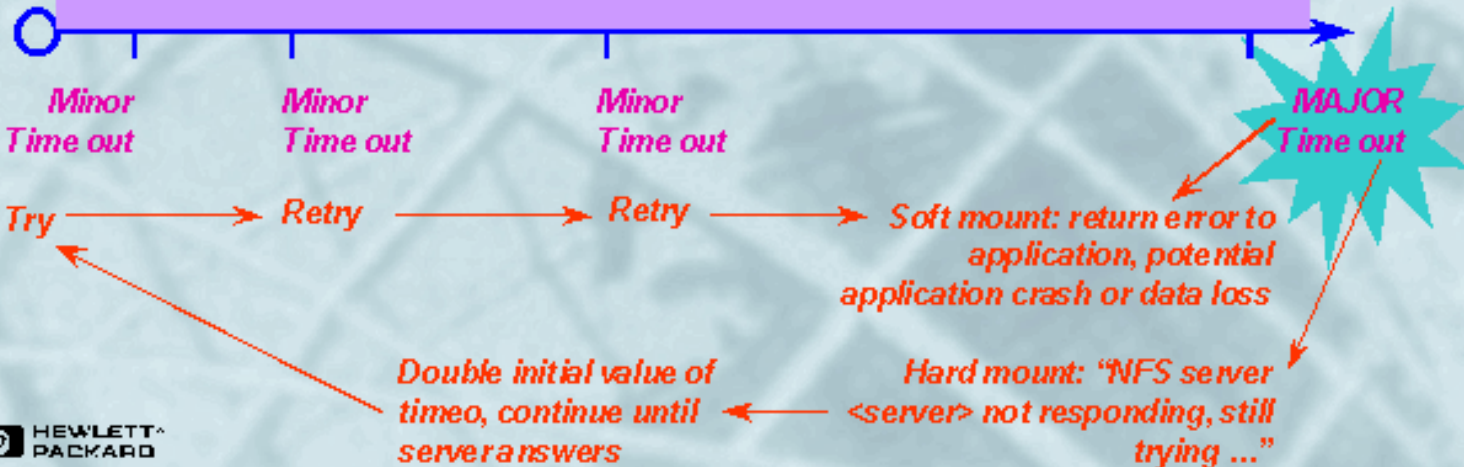
NFS Client Tuning

NFS Client Mount Parameters: Timeout Behavior Example

retrans=4, tim eo=7



File Server or Network Load

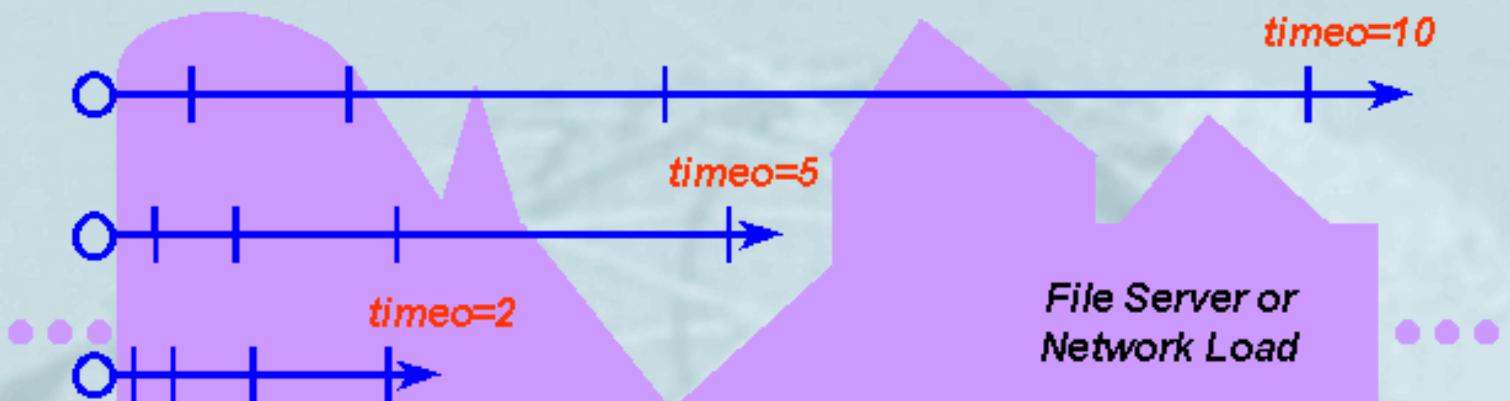


145 99/09/20 Rob Lucke



NFS Client Tuning

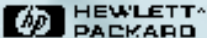
NFS Client Mount Parameters: Adjusting timeo and retrans



View the retrans/timeo pair as implementing a window in time between the original request and the first major NFS timeout

With this mental picture, adjusting the value for timeo will widen or narrow the width of the window

This is far easier than trying to deal with the effects of changing both parameters at the same time



146 99/09/20 Rob Lucke

Slide 146 of 188



NFS Client Tuning

NFS Client Mount Parameters: Data Integrity

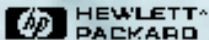
⌘ The client retry behavior, synchronous writes on the server, and the NFS stateless design ensure data integrity for the client **only for HARD** mounts



⌘ Scenario:

- ◆ Client writes data
- ◆ Server receives data, commits it to disk, acknowledges client
- ◆ Client writes next data
- ◆ Server crashes
- ◆ Client starts retrying and continues until server answers
- ◆ Server restarts
- ◆ Server receives data, commits it to disk, acknowledges client
- ◆ ...

Client appears to hang



147 99/09/20 Rob Lucke

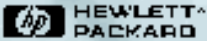
Slide 147 of 188



NFS Client Tuning

NFS Client Mount Parameters: Symptoms of Timeouts

- ⌘ “Your NFS <insert your favorite negative user comment here>”
- ⌘ Complaints about “NFS server <server> not responding, still trying ...” messages
 - ◆ It is hard not to notice these messages on the client, especially if there is no console window running in the user’s VUE/CDE session
- ⌘ Unreasonable times for things like file copies
 - ◆ One example was a file copy from Sun to Auspex server taking 10 minutes, same copy HP to Auspex server taking 10 HOURS



148 99/09/20 Rob Lucke

Slide 148 of 188



NFS Client Tuning

NFS Client Mount Parameters: Finding Timeouts

```

root@hp.prc1-01.home
root@hp.prc1-01.home
root@hp.prc1-01.home
# nfs stat -c

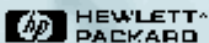
Client ops:
calls      badcalls   responses  badretd  timeout    wait      retransd
541972     000        169        4         2047       0         0

Client nfs:
calls      badcalls   ncldget    ncldeleg
538944     7          539056     0

Version 2 : (10 calls)
call      getattr   setattr   root      lookup    readlink  read
0 0%      0 0%      0 0%      0 0%      0 0%      0 0%
wrcache  write     create    remove    rename    link      symlink
0 0%      0 0%      0 0%      0 0%      0 0%      0 0%
mkdir    mkdir    readdir   statfs
0 0%      0 0%      0 0%      0 0%

Version 3 : (538944 calls)
call      getattr   setattr   lookup    access    readlink  read
0 0%      538904 4% 4848 1% 4848 0% 0 0% 3. 0% 13204- 0%
write     create    mkdir     symlink   mknod    remove    rmdir
15673 1% 132 0% 10 0% 3 0% 0 0% 17 0% 0 0%
rename    link      readdir   readlink  fstat    fsync     nethdrf
10 0% 1 0% 1097 0% 0 0% 15210 0% 15212 0% 2 0%
commit
5517 1%

```



149 99/09/20 Rob Lucke

Source: NFS Client system after server rebooted "nfs stat -c"



NFS Client Tuning

NFS Client Mount Parameters: hard/soft/intr

⌘ soft

- ◆ Once file system is mounted, return an error if a major NFS timeout occurs
- ◆ **Does NOT guarantee data integrity**

⌘ hard

- ◆ Once file system is mounted, retry until server responds to request
- ◆ **Default behavior**

⌘ intr

- ◆ Allows interruption of retries for hard mounted file systems
- ◆ Should be used only on read-only (ro) mounts or data integrity cannot be assured
- ◆ **Default behavior** (nointr disables this behavior)





NFS Client Tuning

NFS Client Mount Parameters: Mounts from /etc/fstab

⌘ retry=n

- ◆ Number of retries for each mount request if the server's mountd does not respond
- ◆ Default value is 1

⌘ fg

- ◆ Mount retries will run in the foreground
- ◆ Default behavior

⌘ bg

- ◆ Mount retries will run in the background
- ◆ Useful for cross-mounts so that systems will not hang if booted in the wrong order
- ◆ **Cross mounting is a BAD idea!**





NFS Client Tuning

NFS Client Mount Parameters: Attribute Caching

⌘ **acdirmin=n, acdirmax=n**

- ◆ Minimum (maximum) seconds to hold directory attributes after directory modification
- ◆ Default values are **acdirmin=30, acdirmax=60**

⌘ **acregmin=n, acregmax=n**

- ◆ Minimum (maximum) seconds to hold file attributes after file modification
- ◆ Default values are **acregmin=3, acregmax=60**

⌘ **actimeo=n**

- ◆ Sets values for **acdirmin, acdirmax, acregmin, acregmax** to "n" seconds
- ◆ No default value

⌘ **noac**

- ◆ Disable file and name attribute caching

⌘ **nocto**

- ◆ Suppress fresh attributes when opening a file



152 99/09/20 Rob Lucke




NFS Client Tuning

NFS Client Mount Parameters: Attribute Caching

- Without attribute caching enabled, every operation that accesses file attributes goes directly to the NFS server, this can greatly increase the load on the server and the network
- Attributes are written back to the server at the end of the maximum period, or along with write activity, to keep the NFS server's view of the file synchronized with the client's view
- Examining the NFS operation mixes shows that a good percentage of a server's operations can be `getattr` or `setattr` calls. Without attribute caching, this percentage would be even larger



 HEWLETT
PACKARD

153 99/09/20 Rob Lucke

Slide 153 of 188



NFS Client Tuning

NFS Client Mount Parameters: Attribute Caching



Operations
that can use
cached
information

```

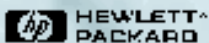
root@hpugerc: /root/home
- nfsstat c

Client rpc:
calls      badcalls  retrans   badxid    timeout   wait      reworded
541973     600       150       4          2347      0         0

Client nfs:
calls      badcalls  noiget    noisleep
538944     7         530966    0

Version 2 : (0 calls)
null      0 0%  getattr  0 0%  setattr  0 0%  root  0 0%  _doamp  0 0%  readlink  0 0%  read  0 0%
wcache   0 0%  write   0 0%  create  0 0%  remove  0 0%  rename  0 0%  link  0 0%  symlink  0 0%
mkdir    0 0%  rmdir   0 0%  readdir  0 0%  statfs  0 0%
Version 3 : (538344 calls)
null      0 0%  getattr  226604 42%  setattr  2348 0%  lookup  22960 6%  access  0 0%  readlink  35 0%  read  15233 28%
write    25678 4%  create  120 0%  mkdir   10 0%  symlink  2 0%  mknod   0 0%  remove  771 0%  rmdir   0 0%
rename   26 0%  link    1 0%  readdir  1007 0%  readdir- 0 0%  fsstat  45210 8%  fsync   45012 8%  pathconf 2 0%
commit   5517 1%

```



154 99/09/20 Rob Lucke



NFS Client Tuning

NFS Client Mount Parameters

rw

- ◆ Mount the NFS file system in read/write mode


ro

- ◆ Mount the NFS file system in read-only mode

devs, nodevs

- ◆ Allow (disallow) access to local client device files from a device file on a file system mounted over NFS
- ◆ This partially plugs a security hole big enough to drive a truck through



 HEWLETT
PACKARD

155 99/09/20 Rob Lucke

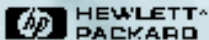
Slide 155 of 188



NFS Client Tuning

NFS Client Automounter

- ⌘ Automounter can help prevent client hangs, by dismounting unused file systems and aids in single-point administration, since clients would otherwise have to hard mount (N-1) file systems
- ⌘ Automounter is driven by “automounter maps” that may be distributed by NIS, or be on the local system’s disk
- ⌘ **Inheritance of automounter parameters that are tuned for non-HP-UX systems is a potential source of performance problems**
- ⌘ The HP-UX 10.X automounter is single-threaded, meaning problems mounting a file system or communicating with a server can hold up other operations
- ⌘ The NFS PV3 version of automounter, AutoFS, is much more robust and performs better than the “default”



156 99/09/20 Rob Lucke



Slide 156 of 188



NFS Client Tuning

NFS Client Automounter Master Map



```

root@hp.pora:/root_home
root@hp.pora: /root_home
# cat /etc/auto.master
# Changes in this file must be reflected in auto_master also
Directory          Sub-map Name      Mount Options
/et                 -hosts            rw,soft,timeo=15,retrans=7,soft
/local_efs          auto.indirect     rw,soft,timeo=15,retrans=7
/home               auto.users        rw,soft,timeo=15,retrans=7
/admin              auto.admin        rw,soft,timeo=15,retrans=7
/wscne              auto.wscne        rw,soft,timeo=15,retrans=7
/-                  auto.direct       rw,soft,timeo=15,retrans=7

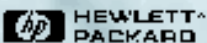
```

Special map entry

Indirect map entry

Direct map entry

Master map for automounter, distributed via NIS. Note that mount options specified in the master map will override values specified in submaps.



157 99/09/20 Rob Lucke



NFS Client Tuning

NFS Client Automounter Direct Maps

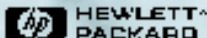


```

root@hpupora:/root_home
# cat /etc/auto.direct
# Mount point for Iron software (the latest version).
#
/cdrom0 -ro,intr,wsize=8192,rsize=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/HP-CX_0.20 Iron data
# And Instant Information
/cdrom1 -ro,intr,wsize=8192,rsize=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/HP-CX_0.20 Inst_Inf
/cdrom2 -ro,intr,wsize=0192,rsize=0192,timeo=15,retrans=7 \
hpupora:/export/fs1/HP-CX_1.00_Inst_Inf
  
```

Mount Options
Mount Target
Mount Source

Direct maps will mount the remote server's directory over the top of the local directory. They are read ONLY at automounter startup time, so changes are not reflected until reboot.



158 99/09/20 Rob Lucke

Slide 158 of 188



NFS Client Tuning

NFS Client Automounter Indirect Maps



```

root@hpupora: /root_home
# cat /etc/auto.indirect
CC          -rw,intr,wsz=3192,rsz=8192,timeo=15,retr=7 \
hpupora:/export/esc1/${OS} ${ARCH} ${REV}/CC

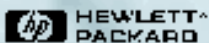
SharedX    -rw,intr,wsz=3192,rsz=8192,timeo=15,retr=7 \
hpupora:/export/esc1/${OS}_${ARCH}_${REV}/SharedX

acu        -rw,intr,wsz=3192,rsz=8192,timeo=15,retr=7 \
hpupora:/export/esc1/${OS}_${ARCH}_${REV}/acu

audio      -rw,intr,wsz=3192,rsz=8192,timeo=15,retr=7 \
hpupora:/export/esc1/${OS}_${ARCH}_${REV}/audio

blinklink  -rw,intr,wsz=3192,rsz=8192,timeo=15,retr=7 \
hpupora:/export/esc1/${OS} ${ARCH} ${REV}/blinklink
  
```

Indirect maps will mount the server's directory underneath a directory that is managed by the automounter and specified in the master map, /local_nfs, in this case. Changes in the map are recognized when the directory needs to be remounted.



159 99/09/20 Rob Lucke

Slide 159 of 188



NFS Client Tuning

Vendor Neutral Automounter Maps



⌘ In the previous slide, the automounter map contained variables:

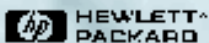
```
blinklink -rw,intr,rsize=8192,wsiz=8192,timeo=15,retrans=7 \
hpupora:/export/fs 1/${OS}_${REV}_${ARCH}/blinklink
```

⌘ The client systems start automounter from `/etc/rc.config.d/nfsconf` as follows:

```
OS=`uname -s`
REV=`uname -r | cut -c3-7`
ARCH=`uname -m | cut -c6`00
LOCAL_HOST=`hostname`
AUTOMOUNTD_OPTIONS="-D OS=${OS} -D REV=${REV} -D ARCH=${ARCH} \
-D LOCAL_HOST=${LOCAL_HOST}"
```



⌘ This defines the variables OS, REV, ARCH, and LOCAL_HOST in the automounter's environment



160 99/09/20 Rob Lucke

Slide 160 of 188



NFS Client Tuning

Vendor Neutral Automounter Maps

- The automounter will substitute the variable values into the maps, yielding the following:

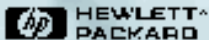


```
blinklink -rw,intr,rsize=8192,wsize=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/$(OS)_$(REV)_$(ARCH)/blinklink
```

```
blinklink -rw,intr,rsize=8192,wsize=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/HP-UX_10.20_700/blinklink
```



- This is a great tool for separating different data sets from one another, based on client-specific variable values



161 99/09/20 Rob Lucke

Slide 161 of 188



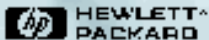
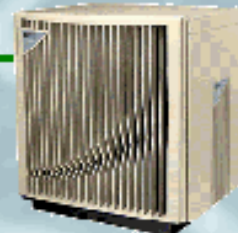
NFS Client Tuning

"Hybrid" Automounter Maps

⌘ Direct map mounts are always visible to the user, since they mimic hard mounts. They also will cause the automounter to be queried each time the directory that contains them is read, possibly causing a "mount storm"

⌘ Indirect map mount points are only visible after they have been touched and mounted, and cannot overlay existing directories. Once dismantled, they will not be accessed by tools like "find" that do recursive directory walking, but their absence might confuse users.

⌘ A mixture of the two behaviors (eliminating the mount storms) is possible by a special "trick" that "converts" a direct map into an indirect map



162 99/09/20 Rob Lucke



NFS Client Tuning

"Hybrid" Automounter Maps

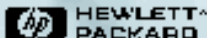
- ⌘ Create an indirect map containing the directories that need to be mounted (see the previous auto.indirect map example)
- ⌘ Create symbolic links in the client's file system that point into the directory managed by the automounter:



```
In -s /local_nfs/CC /opt/CC
In -s /local_nfs/asu /opt/asu
...
```



- ⌘ Now, when a user looks at /opt, he will see CC, asu, and the familiar directory entries, but they will not yet be mounted
- ⌘ When the user accesses /opt/CC, automounter will see /local_nfs/CC being touched, and will mount the directory under /local_nfs, thus validating the link in /opt



163 99/09/20 Rob Lucke

Slide 163 of 188



Seminar Agenda

- ⌘ Design Issues
- ⌘ NFS Fileserver Benchmark Data
- ⌘ Measuring NFS Client Requirements
- ⌘ NFS Fileserver Sizing
 - CPU
 - LAN
 - Memory
 - Disk
- ⌘ NFS Fileserver Tuning
- ⌘ NFS Client Tuning
- ⌘ NFS PV3



164 99/09/20 Rob Lucke



NFS PV3 Features



⌘ New protocol (Version 3)

- ◆ Safe asynchronous write with COMMIT request
- ◆ Post operation attributes returned from server
- ◆ Large file size (64-bit) support

⌘ AutoFS

- ◆ Multi-threaded “automountd”
- ◆ “automount” command communicates with automountd
- ◆ Will mount both PV2 and PV3 file systems



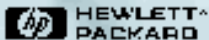
⌘ CacheFS

- ◆ Local caching of read-mostly data



⌘ NIS+

- ◆ Expanded name service switch



165 99/09/20 Rob Lucke



NFS PV3 Features

How do I get PV3?



⌘ Versions of HP-UX since 10.30 already contain the new features

⌘ HP-UX 10.20 (*GOOD NEWS!*)

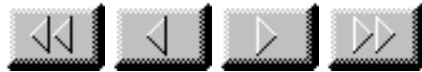
- The need for separate “networking ACEs” and dual-stream “smart” patches is eliminated (*thank you, NFS labs!*)
- Installing the latest NFS/NIS “Mega” patches (and their dependencies) will allow you to have BOTH of the NFS protocols (PV2 and PV3)
- The behavior will default to your current configuration for AutoFS (automount) and NFS PV2 (PV3)
- Non-default behavior is enabled by setting values in the `/etc/rc.config.d/nfsconf` file



⌘ See NFS.ps in `/usr/share/doc` after installation



166 99/09/20 Rob Lucke



NFS PV3 Features

Additional Mount Options



⌘ vers=n

- ◆ Specify the NFS protocol version to start out with for this mount
- ◆ If the server does not support NFS PV3, then PV2 will be used

⌘ O

- ◆ Specify a mount that is to overlay an existing mount
- ◆ Valid only for NFS, and “-F nfs” must be specified to mount

⌘ grpuid



- ◆ Force a newly created file in the mounted file system to inherit its group ID from the parent directory
- ◆ If the SETGID bit is set on the directory AND the grpuid option is specified (Default is to inherit effective GID from creating process.)



167 99/09/20 Rob Lucke



NFS PV3 Features

Additional Mount Options



⌘ **fstype**

⌘ **backfstype**

- ◆ Type of the back file system

⌘ **cachedir**

- ◆ Directory path to the HFS cache

⌘ **write-around**

- ◆ Default behavior: write to server, read from cache

⌘ **non-shared**

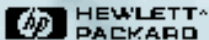
- ◆ Write to both cache and server

⌘ **noconst**

- ◆ Disable cache consistency checking

⌘ **demandconst**

- ◆ Check cache consistency only when specifically asked to with "cfsadmin -s"



HEWLETT
PACKARD
168 99/09/20 Rob Lucke

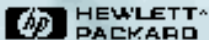


NFS PV3 Features

AutoFS



- ⌘ The “automount” command now communicates with the “automountd”, reading in the automounter maps
- ⌘ The map information may now be changed without restarting automounter (e.g. the whole system)
- ⌘ AutoFS now has kernel components, which means faster performance than the old automounter
- ⌘ AutoFS can mount both NFS PV3 and NFS PV2 file systems, the old automounter can only mount NFS PV2 file systems.



169 99/09/20 Rob Lucke

Slide 169 of 188



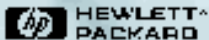
NFS PV3 Features

CacheFS Terms



CacheFS terms:

- front file system The file system being cached
- back file system The file system containing the cache data
- cold cache A cache that does not have any data in it
- warm cache A cache that contains data in the front file system



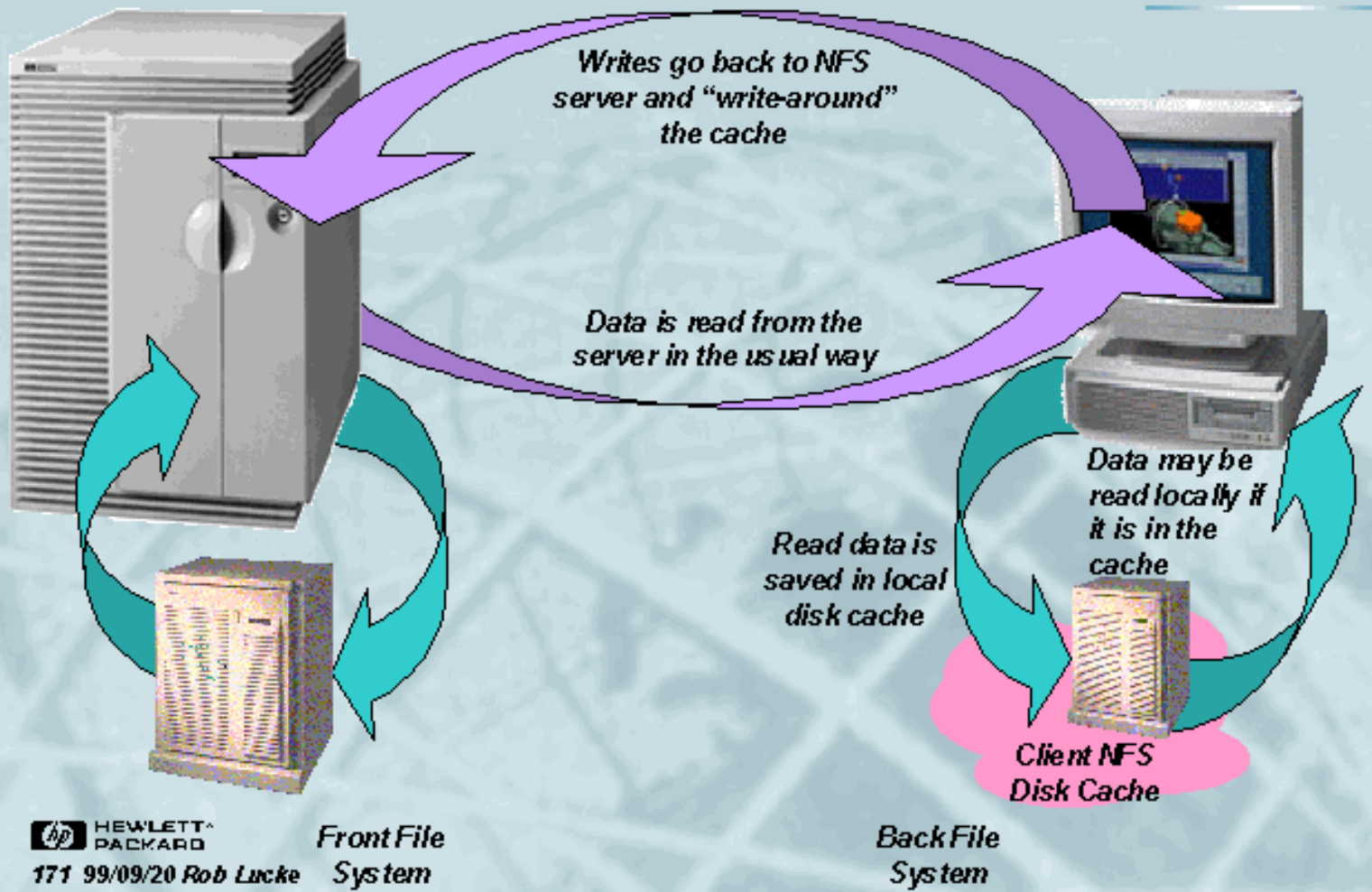
170 99/09/20 Rob Lucke

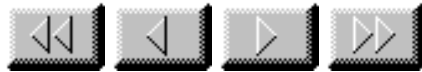
Slide 170 of 188



NFS PV3 Features

CacheFS Operation





NFS PV3 Features

CacheFS Operation

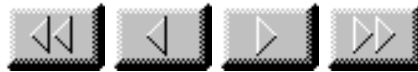


- ⌘ CacheFS maintains consistency with the back file system by checking back with the server, much like the attribute caching described previously
- ⌘ CacheFS is good for read-only or read-mostly data, for that reason (and the fact that writes are not cached)
- ⌘ CacheFS supports *only* HFS as the front file system
- ⌘ Read performance can improve 4-5x in the best case scenario
- ⌘ Use CacheFS for executable files, libraries, models, and other constant data



172 99/09/20 Rob Lucke

Slide 172 of 188



NFS PV3 Features

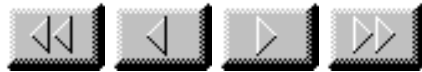
CacheFS Operation



- ⌘ You must have HFS file systems to contain the cache
- ⌘ Use the cfsadm command to create the cache
 - ◆ “cfsadm -c /cache_dir”
- ⌘ CacheFS will perform better with a dedicated cache volume (i.e. do not store other data in the file system)
- ⌘ Multiple mounts may be cached into the same front file system
- ⌘ “mount -F cachefs -o backfstype=nfs,cachedir=/cache_dir \ server:/opt/emacs /opt/emacs”
- ⌘ Add to /etc/fstab:


```
“server:/opt/emacs /opt/emacs cachefs
backfstype=nfs,cachedir=/cache 0 0”
```





NFS PV3 Features

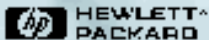
CacheFS Operation



⌘ Automounting with CacheFS:

- ◆ `/opt/emacs -ro,nosuid,fstype=cachefs,backfstype=nfs, cachedir=/cache_dir server:/opt/emacs`
- ◆ `emacs -ro,nosuid,fstype=cachefs,backfstype=nfs, cachedir=/cache_dir server:/opt/emacs`

⌘ If you changed an automounter map, issue the “automount” command to re-read the maps



174 99/09/20 Rob Lucke

Slide 174 of 188

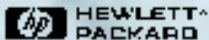


NFS PV3 Features



- ⌘ There have been reports of limitations on the number of files that may be stored in CacheFS
- ⌘ The “*maxcnodes*” parameter on the client system determines the number of “cache nodes” that can exist in the CacheFS cache
- ⌘ This number is currently “hard-coded” at 500
- ⌘ You must currently *adb* the kernel on the client **AT BOOT TIME** to set this parameter to a larger value:

```
print "maxcnodes/W 10000" | adb -w /stand/vmunix /dev/mem
```
- ⌘ This value must be smaller than the number of inodes specified by *ninode* (remember that the CacheFS cache is located on an HFS file system)



175 99/09/20 Rob Lucke

Slide 175 of 188



NFS PV3 Features

NIS+



- ⌘ Secure NFS is not supported under HP-UX
- ⌘ Secure RPC is supported
- ⌘ There is additional information in the `/etc/nsswitch.conf` file
 - ◆ automount
 - ◆ netgroup
 - ◆ networks
 - ◆ protocols
 - ◆ rpc
 - ◆ services
- ⌘ Each of these services may select from “dns”, “nis”, or “files” as the source of information
- ⌘ You may now use a local definition or the global definition in the desired sequence



176 99/09/20 Rob Lucke

Slide 176 of 188



NFS PV3 Features

NIS+



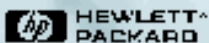
⌘ The default configuration of `/etc/nsswitch.conf` is:

hosts:	dns	nis	files
protocols:		nis	files
services:		nis	files
networks:		nis	files
netgroup:		nis	files
rpc:		nis	files

⌘ Each source can have four `status=action` pairs, with the defaults being:



SUCCESS=return
 NOTFOUND=return
 UNAVAIL=continue
 TRYAGAIN=return



177 99/09/20 Rob Lucke

Slide 177 of 188



NFS PV3 Features

NIS+




- ⌘ The default *action* configuration of *etc/nsswitch.conf* is probably not what you want, it only queries one service, then gives up. I use:

```
hosts:      dns [NOTFOUND=continue,UNAVAIL=continue]  \
           nis [NOTFOUND=continue,UNAVAIL=continue]  files
```

[...]

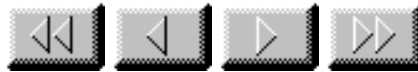
- ⌘ This will try *dns*, and continue on to *nis* and the local files if any service is unavailable or cannot find the proper entry
- ⌘ Check your configuration with the *nslookup* "policy" command



 HEWLETT
PACKARD

178 99/09/20 Rob Lucke

Slide 178 of 188



NFS PV3 Features



⌘ NFS PV3 allows increasing the read/write block size beyond the 8 KB allowed by PV2 (other vendors allow up to 32 KB, we are currently limited to 24 KB maximum)

⌘ How does this change the 4 KB NFS “magic number” which is based on the average transfer size?

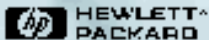
⌘ Using the NFS PV2 workload, the average NFS IOP transfer is:

⌘ NFS block size should match physical file system block size to avoid read/modify/write disk operations (use 16 KB for now)

⌘ Larger blocks mean more stress on server and network infrastructure (and more traffic for retries!)



<i>NFS Block Size</i>	<i>IOP Value</i>	<i>Rounded Value</i>
8 KB	3152	4 KB
16 KB	6183	6 KB
24 KB	9214	9 KB
32 KB	12245	12 KB



179 99/09/20 Rob Lucke



Slide 179 of 188



NFS Design Case Studies

⌘ The following slides contain actual customer configurations that have been encountered in the course of environmental assessments



⌘ Your task is to find the bottlenecks, locate areas that can be improved, or answer the customer's request



180 99/09/20 Rob Lucke

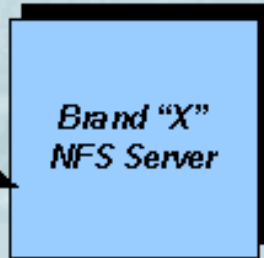
Slide 180 of 188



NFS Design Case Studies

Case Study 1: "Your NFS server is terrible"

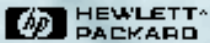
*Software builds
run here*



10baseT



*Find at least three problems
with this configuration*

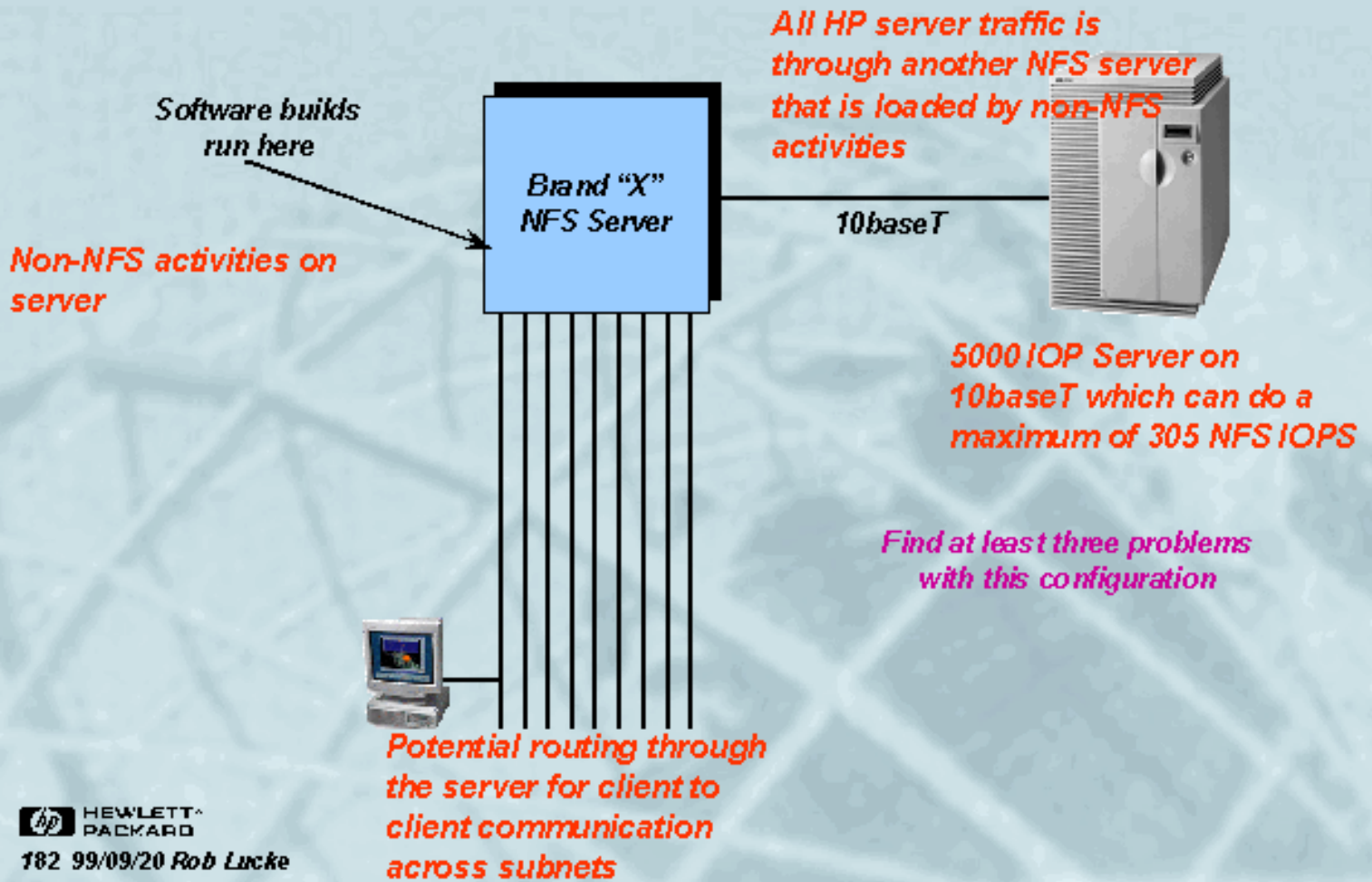


181 99/09/20 Rob Lucke



NFS Design Case Studies

Case Study 1: "Your NFS server is terrible"



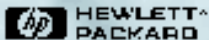
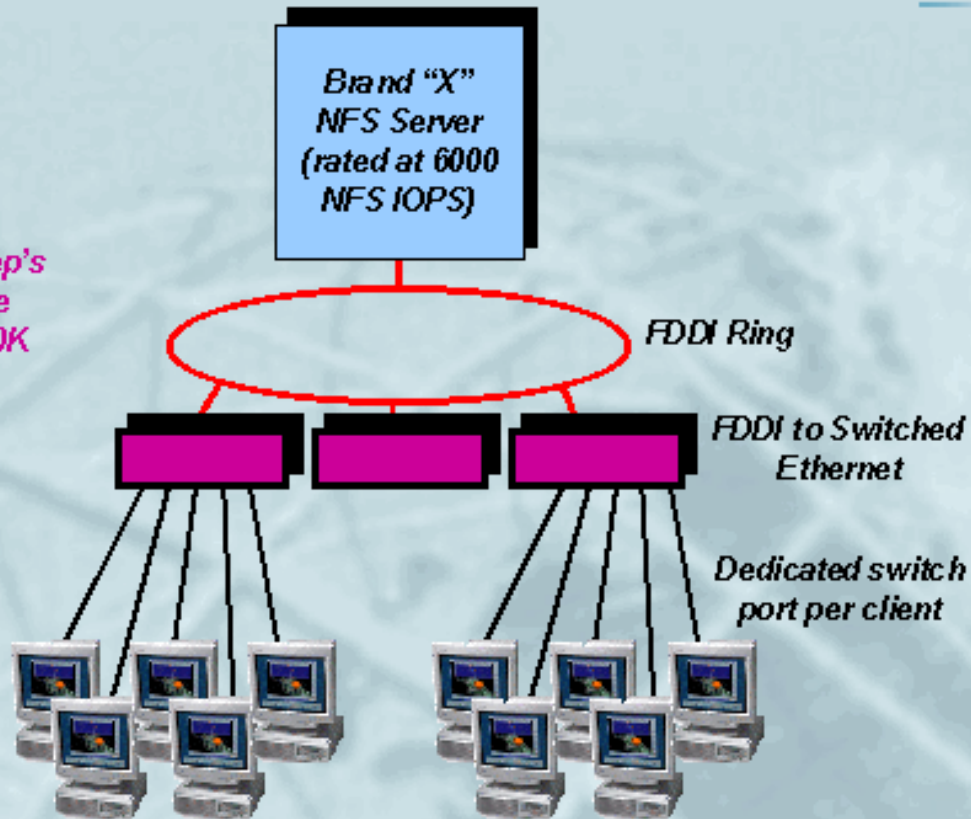


NFS Design Case Studies

Case Study 2: "My server only handles half of my clients"

The brand "S" sales rep's solution was to sell the customer a nother \$100K server. What is your solution?

600 NFS clients doing MCAD. Measured peaks of 10 NFS IOPS per client



183 99/09/20 Rob Lucke

Slide 183 of 188



NFS Design Case Studies

Case Study 2: "My server only handles half of my clients"

The brand "S" sales rep's solution was to sell the customer a nother \$100K server. What is your solution?

Brand "X"
NFS Server
(rated at 6000
NFS IOPS)

Single FDDI ring is rated at
3000 NFS IOPS maximum

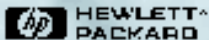


FDDI Ring



Dedicated switch
port per client

600 NFS clients doing
MCAD. Measured
peaks of 10 NFS IOPS
per client



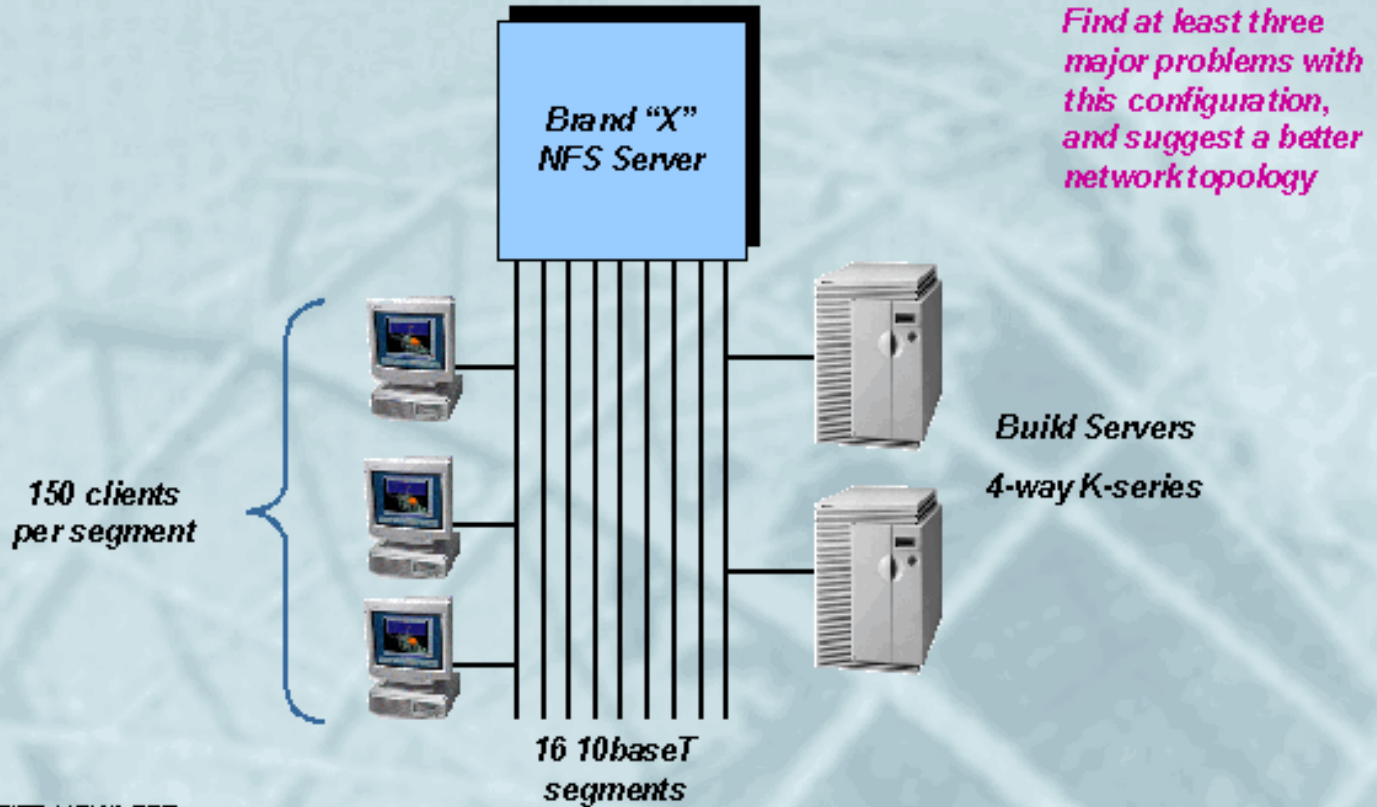
184 99/09/20 Rob Lucke

Slide 184 of 188

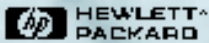


NFS Design Case Studies

Case Study 3: "My software build performance is terrible"



Find at least three major problems with this configuration, and suggest a better network topology



185 99/09/20 Rob Lucke



NFS Design Case Studies

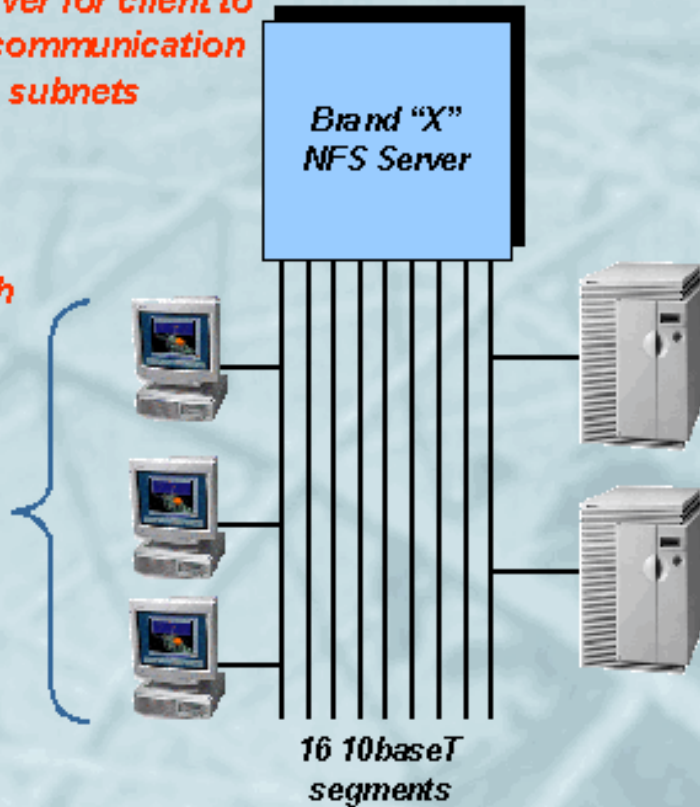
Case Study 3: "My software build performance is terrible"

Potential routing through the server for client to client communication across subnets

Find at least three major problems with this configuration, and suggest a better network topology

Too many high traffic clients sharing LANs

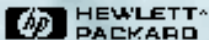
150 clients per segment



5000 IOP Server on 10baseT which can do a maximum of 305 NFS IOPS
Build Servers

4-way K-series

Replace the current network with a switch, dedicated server and client ports, and higher bandwidth links



186 99/09/20 Rob Lucke



NFS Design Case Studies

Case Study 4: "What can your NFS servers do?"

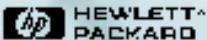
What is the largest number of users supported by one NFS server?

Assume that 10-20% of the users have workstations and the rest are accessing the server from PCs or X-terminals.

This question applies specifically to a computing environment that has 5000 users and we want to minimize administration costs.



What information do you need, and what steps are necessary to answer this customer's request?



187 99/09/20 Rob Lucke

Slide 187 of 188



NFS Design Case Studies

Case Study 4: "What can your NFS servers do?"

What is the largest number of users supported by one NFS server?

Assume that 10-20% of the users have workstations and the rest are accessing the server from PCs or X-terminals.

This question applies specifically to a computing environment that has 5000 users and we want to minimize administration costs.



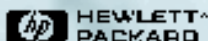
1000 workstations X 125 NFS IOPS = 125,000 NFS IOPS

4000 X-terminals X 20 NFS IOPS = 80,000 NFS IOPS

TOTAL 205,000 NFS IOPS

V2500-32 = 50,000 NFS IOPS

Total of 5 V2500 systems unless high-availability is needed, in which case up to 10 V2500 systems



188 99/09/20 Rob Lucke

NFS Design Case Studies Case Study 4: “What can your NFS servers do?”

What is the largest number of users supported by one NFS server? Assume that 10-20% of the users have workstations and the rest are accessing the server from PCs or X-terminals. This question applies specifically to a computing environment that has 5000 users and we want to minimize administration costs.

1000 workstations X 125 NFS IOPS = 125,000 NFS IOPS
4000 X-terminals X 20 NFS IOPS = 80,000 NFS IOPS

TOTAL 205,000 NFS IOPS

V2500-32 = 50,000 NFS IOPS

Total of 5 V2500 systems unless high-availability is needed, in which case up to 10 V2500 systems

[Previous slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 4: “What can your NFS servers do?”

What is the largest number of users supported by one NFS server? Assume that 10-20% of the users have workstations and the rest are accessing the server from PCs or X-terminals. This question applies specifically to a computing environment that has 5000 users and we want to minimize administration costs.

What information do you need, and what steps are necessary to answer this customer's request?

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 3: “My software build performance is terrible”

Brand “X” NFS Server

150 clients per segment

16 10baseT segments

Build Servers

4-way K-series

Find at least three major problems with this configuration, and suggest a better network topology

5000 IOP Server on 10baseT which can do a maximum of 305 NFS IOPS

Potential routing through the server for client to client communication across subnets

Too many high traffic clients sharing LANs

Replace the current network with a switch, dedicated server and client ports, and higher bandwidth links

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 3: “My software build performance is terrible”

Brand “X” NFS Server

150 clients per segment

16 10baseT segments

Build Servers

4-way K-series

Find at least three major problems with this configuration, and suggest a better network topology

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 2: “My server only handles half of my clients”

Brand “X” NFS Server (rated at 6000 NFS IOPS)

FDDI Ring

FDDI to Switched Ethernet

600 NFS clients doing MCAD. Measured peaks of 10 NFS IOPS per client

The brand “S” sales rep’s solution was to sell the customer another \$100K server. What is your solution?

Dedicated switch port per client

Single FDDI ring is rated at 3000 NFS IOPS maximum

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 2: “My server only handles half of my clients”

Brand “X” NFS Server (rated at 6000 NFS IOPS)

FDDI Ring

FDDI to Switched Ethernet

600 NFS clients doing MCAD. Measured peaks of 10 NFS IOPS per client

The brand “S” sales rep’s solution was to sell the customer another \$100K server. What is your solution?

Dedicated switch port per client

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 1: “Your NFS server is terrible”

Brand “X” NFS Server

10baseT

Software builds run here

Find at least three problems with this configuration

5000 IOP Server on 10baseT which can do a maximum of 305 NFS IOPS

Potential routing through the server for client to client communication across subnets

Non-NFS activities on server

All HP server traffic is through another NFS server that is loaded by non-NFS activities

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies Case Study 1: “Your NFS server is terrible”

Brand “X” NFS Server

10baseT

Software builds run here

Find at least three problems with this configuration

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Case Studies

- **The following slides contain actual customer configurations that have been encountered in the course of environmental assessments**
- **Your task is to find the bottlenecks, locate areas that can be improved, or answer the customer's request**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features

- **NFS PV3 allows increasing the read/write block size beyond the 8 KB allowed by PV2 (other vendors allow up to 32 KB, we are currently limited to 24 KB maximum)**
- **How does this change the 4 KB NFS “magic number” which is based on the average transfer size?**
- **Using the NFS PV2 workload, the average NFS IOP transfer is:**
- **NFS block size should match physical file system block size to avoid read/modify/write disk operations (use 16 KB for now)**
- **Larger blocks mean more stress on server and network infrastructure (and more traffic for retries!)**

NFS Block Size

IOP Value

Rounded Value

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features NIS+

- The default action configuration of `/etc/nsswitch.conf` is probably not what you want, it only queries one service, then gives up. I use: `hosts:
dns[NOTFOUND=continue,UNAVAIL=continue] \
nis[NOTFOUND=continue,UNAVAIL=continue]
files [...]`
- This will try dns, and continue on to nis and the local files if any service is unavailable or cannot find the proper entry
- Check your configuration with the `nslookup` “policy” command

[Previous slide](#)[Next slide](#)[Back to first slide](#)[View graphic version](#)

NFS PV3 Features NIS+

- The default configuration of `/etc/nsswitch.conf` is: `hosts: dns nis files protocols: nis files services: nis files networks: nis files netgroup: nis files rpc: nis files`
- Each source can have four `status=action` pairs, with the defaults being: `SUCCESS=return`
`NOTFOUND=return` `UNAVAIL=continue`
`TRYAGAIN=return`

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features NIS+

- **Secure NFS is not supported under HP-UX**
- **Secure RPC is supported**
- **There is additional information in the `/etc/nsswitch.conf` file**
 - automount
 - netgroup
 - networks
 - protocols
 - rpc
 - services
- **Each of these services may select from “dns”, “nis”, or “files” as the source of information**
- **You may now use a local definition or the global definition in the desired sequence**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features

- There have been reports of limitations on the number of files that may be stored in CacheFS
- The “maxcnodes” parameter on the client system determines the number of “cache nodes” that can exist in the CacheFS cache
- This number is currently “hard-coded” at 500
- You must currently adb the kernel on the client **AT BOOT TIME** to set this parameter to a larger value: `print “maxcnodes/W 10000” | adb -w /stand/vmunix /dev/mem`
- This value must be smaller than the number of inodes specified by ninode (remember that the CacheFS cache is located on an HFS file system)

[Previous slide](#)[Next slide](#)[Back to first slide](#)[View graphic version](#)

NFS PV3 Features CacheFS Operation

• Automounting with CacheFS:

- “/opt/emacs -ro,nosuid,fstype=cachefs,backfstype=nfs, cachedir=/cache_dir server:/opt/emacs”
- “emacs -ro,nosuid,fstype=cachefs,backfstype=nfs, cachedir=/cache_dir server:/opt/emacs”

• If you changed an automounter map, issue the “automount” command to re-read the maps

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features CacheFS Operation

- You must have HFS file systems to contain the cache
- Use the `cfsadm` command to create the cache
 - “`cfsadm -c /cache_dir`”
- CacheFS will perform better with a dedicated cache volume (i.e. do not store other data in the file system)
- Multiple mounts may be cached into the same front file system
- “`mount -F cachefs -o backfstype=nfs,cachedir=/cache_dir \ server:/opt/emacs /opt/emacs`”
- Add to `/etc/fstab`: “`server:/opt/emacs /opt/emacs cachefs backfstype=nfs,cachedir=/cache 0 0`”

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features CacheFS Operation

- **CacheFS maintains consistency with the back file system by checking back with the server, much like the attribute caching described previously**
- **CacheFS is good for read-only or read-mostly data, for that reason (and the fact that writes are not cached)**
- **CacheFS supports only HFS as the front file system**
- **Read performance can improve 4-5x in the best case scenario**
- **Use CacheFS for executable files, libraries, models, and other constant data**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features CacheFS Operation

Front File System

Back File System

Writes go back to NFS server and “write-around” the cache

Client NFS Disk Cache

Read data is saved in local disk cache

Data is read from the server in the usual way

Data may be read locally if it is in the cache

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features CacheFS Terms

• CacheFS terms:

- front file system The file system being cached
- back file system The file system containing the cache data
- cold cache A cache that does not have any data in it
- warm cache A cache that contains data in the front file system

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features AutoFS

- The “automount” command now communicates with the “automountd”, reading in the automounter maps
- The map information may now be changed without restarting automounter (e.g. the whole system)
- AutoFS now has kernel components, which means faster performance than the old automounter
- AutoFS can mount both NFS PV3 and NFS PV2 file systems, the old automounter can only mount NFS PV2 file systems.

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features Additional Mount Options

- **fstype**

- **backfstype**

- Type of the back file system

- **cachedir**

- Directory path to the HFS cache

- **write-around**

- Default behavior: write to server, read from cache

- **non-shared**

- Write to both cache and server

- **noconst**

- Disable cache consistency checking

- **demandconst**

- Check cache consistency only when specifically asked to with “cfsadmin -s”

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features Additional Mount Options

- **vers=n**

- Specify the NFS protocol version to start out with for this mount
- If the server does not support NFS PV3, then PV2 will be used

- **O**

- Specify a mount that is to overlay an existing mount
- Valid only for NFS, and “-F nfs” must be specified to mount

- **grpuid**

- Force a newly created file in the mounted file system to inherit its group ID from the parent directory
- If the SETGID bit is set on the directory AND the grpuid option is specified (Default is to inherit effective GID from creating process.)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features How do I get PV3?

- **Versions of HP-UX since 10.30 already contain the new features**
- **HP-UX 10.20 (GOOD NEWS!)**
 - The need for separate “networking ACEs” and dual-stream “smart” patches is eliminated (thank you, NFS labs!)
 - Installing the latest NFS/NIS “Mega” patches (and their dependencies) will allow you to have BOTH of the NFS protocols (PV2 and PV3)
 - The behavior will default to your current configuration for AutoFS (automount) and NFS PV2 (PV3)
 - Non-default behavior is enabled by setting values in the /etc/rc.config.d/nfsconf file
- **See NFS.ps in /usr/share/doc after installation**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS PV3 Features

• New protocol (Version 3)

- Safe asynchronous write with COMMIT request
- Post operation attributes returned from server
- Large file size (64-bit) support

• AutoFS

- Multi-threaded “automountd”
- “automount” command communicates with automountd
- Will mount both PV2 and PV3 file systems

• CacheFS

- Local caching of read-mostly data

• NIS+

- Expanded name service switch

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning “Hybrid” Automounter Maps

- **Create an indirect map containing the directories that need to be mounted (see the previous auto.indirect map example)**
- **Create symbolic links in the client’s file system that point into the directory managed by the automounter: `ln -s /local_nfs/CC /opt/CC` `ln -s /local_nfs/asu /opt/asu ...`**
- **Now, when a user looks at /opt, he will see CC, asu, and the familiar directory entries, but they will not yet be mounted**
- **When the user accesses /opt/CC, automounter will see /local_nfs/CC being touched, and will mount the directory under /local_nfs, thus validating the link in /opt**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning “Hybrid” Automounter Maps

- **Direct map mounts are always visible to the user, since they mimic hard mounts. They also will cause the automounter to be queried each time the directory that contains them is read, possibly causing a “mount storm”**
- **Indirect map mount points are only visible after they have been touched and mounted, and cannot overlay existing directories. Once dismounted, they will not be accessed by tools like “find” that do recursive directory walking, but their absence might confuse users.**
- **A mixture of the two behaviors (eliminating the mount storms) is possible by a special “trick” that “converts” a direct map into an indirect map**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning Vendor Neutral Automounter Maps

- The automounter will substitute the variable values into the maps, yielding the following:
blinklink
-rw,intr,rsize=8192,wspace=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/\${OS}_\${REV}_\${ARCH}/blinklink
blinklink
-rw,intr,rsize=8192,wspace=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/HP-UX_10.20_700/blinklink
- This is a great tool for separating different data sets from one another, based on client-specific variable values

[Previous slide](#)[Next slide](#)[Back to first slide](#)[View graphic version](#)

NFS Client Tuning Vendor Neutral Automounter Maps

- In the previous slide, the automounter map contained variables: blinklink
-rw,intr,rsize=8192,wsiz=8192,timeo=15,retrans=7 \
hpupora:/export/fs1/\${OS}_\${REV}_\${ARCH}/blinklink
- The client systems start automounter from /etc/rc.config.d/nfsconf as follows: OS=`uname -s`
REV=`uname -r | cut -c3-7` ARCH=`uname -m | cut -c6`
LOCAL_HOST=`hostname`
AUTOMOUNTD_OPTIONS=" -D OS=\${OS} -D REV=\${REV} -D ARCH=\${ARCH} \ -D LOCAL_HOST=\${LOCAL_HOST}"
- This defines the variables OS, REV, ARCH, and LOCAL_HOST in the automounter's environment

[Previous slide](#)[Next slide](#)[Back to first slide](#)[View graphic version](#)

NFS Client Tuning NFS Client Automounter Indirect Maps

Indirect maps will mount the server's directory underneath a directory that is managed by the automounter and specified in the master map, /local_nfs, in this case. Changes in the map are recognized when the directory needs to be remounted.

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Automounter Direct Maps

Direct maps will mount the remote server's directory over the top of the local directory. They are read ONLY at automounter startup time, so changes are not reflected until reboot.

Mount Options Mount Target Mount Source

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Automounter Master Map

Special map entry

Indirect map entry

Direct map entry

Master map for automounter, distributed via NIS. Note that mount options specified in the master map will override values specified in submaps.

Mount Options

Sub-map Name

Directory

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Automounter

- Automounter can help prevent client hangs, by dismounting unused file systems and aids in single-point administration, since clients would otherwise have to hard mount (N-1) file systems
- Automounter is driven by “automounter maps” that may be distributed by NIS, or be on the local system’s disk
- Inheritance of automounter parameters that are tuned for non-HP-UX systems is a potential source of performance problems
- The HP-UX 10.X automounter is single-threaded, meaning problems mounting a file system or communicating with a server can hold up other operations
- The NFS PV3 version of automounter, AutoFS, is much more robust and performs better than the “default”

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters

- **rw**

- Mount the NFS file system in read/write mode

- **ro**

- Mount the NFS file system in read-only mode

- **devs, nodevs**

- Allow (disallow) access to local client device files from a a device file on a file system mounted over NFS
- This partially plugs a security hole big enough to drive a truck through

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Attribute Caching

Operations that can use cached information

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Attribute Caching

- **Without attribute caching enabled, every operation that accesses file attributes goes directly to the NFS server, this can greatly increase the load on the server and the network**
- **Attributes are written back to the server at the end of the maximum period, or along with write activity, to keep the NFS server's view of the file synchronized with the client's view**
- **Examining the NFS operation mixes shows that a good percentage of a server's operations can be getattr or setattr calls. Without attribute caching, this percentage would be even larger**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Attribute Caching

- **acdirmin=n, acdirmax=n**

- Minimum (maximum) seconds to hold directory attributes after directory modification
- Default values are acdirmin=30, acdirmax=60

- **acregmin=n, acregmax=n**

- Minimum (maximum) seconds to hold file attributes after file modification
- Default values are acregmin=3, acregmax=60

- **actimeo=n**

- Sets values for acdirmin, acdirmax, acregmin, acregmax to “n” seconds
- No default value

- **noac**

- Disable file and name attribute caching

- **nocto**

- Suppress fresh attributes when opening a file

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Mounts from /etc/fstab

- **retry=n**

- Number of retries for each mount request if the server's mountd does not respond
- Default value is 1

- **fg**

- Mount retries will run in the foreground
- Default behavior

- **bg**

- Mount retries will run in the background
- Useful for cross-mounts so that systems will not hang if booted in the wrong order
- Cross mounting is a BAD idea!

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: hard/soft/intr

• soft

- Once file system is mounted, return an error if a major NFS timeout occurs
- Does NOT guarantee data integrity

• hard

- Once file system is mounted, retry until server responds to request
- Default behavior

• intr

- Allows interruption of retries for hard mounted file systems
- Should be used only on read-only (ro) mounts or data integrity cannot be assured
- Default behavior (nointr disables this behavior)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Finding Timeouts

Source: NFS Client system after server rebooted “nfsstat -c”

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Symptoms of Timeouts

- **“Your NFS <insert your favorite negative user comment here>”**
- **Complaints about “NFS server <server> not responding, still trying ...” messages**
 - It is hard not to notice these messages on the client, especially if there is no console window running in the user’s VUE/CDE session
- **Unreasonable times for things like file copies**
 - One example was a file copy from Sun to Auspex server taking 10 minutes, same copy HP to Auspex server taking 10 HOURS

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Data Integrity

- **The client retry behavior, synchronous writes on the server, and the NFS stateless design ensure data integrity for the client only for HARD mounts**

- **Scenario:**

- Client writes data
- Server receives data, commits it to disk, acknowledges client
- Client writes next data
- Server crashes
- Client starts retrying and continues until server answers
- Server restarts
- Server receives data, commits it to disk, acknowledges client
- ...

Client appears to hang

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Adjusting timeo and retrans

View the retrans/timeo pair as implementing a window in time between the original request and the first major NFS timeout With this mental picture, adjusting the value for timeo will widen or narrow the width of the window This is far easier than trying to deal with the effects of changing both parameters at the same time

timeo=10

timeo=5

timeo=2

File Server or Network Load

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Timeout Behavior Example

- **retrans=4, timeo=7**

Minor Time out

Minor Time out

MAJOR Time out

Minor Time out

Try

Retry

Retry

File Server or Network Load

Soft mount: return error to application, potential application crash or data loss
Hard mount: “NFS server <server> not responding, still trying ...”

Double initial value of timeo, continue until server answers

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Timeout Behavior Example

- **retrans=4, timeo=7**

Minor Time out

Minor Time out

MAJOR Time out

Minor Time out

Try

Retry

Succeed

File Server or Network Load

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Timeout Behavior Example

- **retrans=4, timeo=7**

Initial RPC Request

retrans=1

retrans=2

retrans=3

retrans=4

Subsequent Retries

Minor Time out

Minor Time out

MAJOR Time out

Minor Time out

2*timeo

4*timeo

8*timeo

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: Timeout Behavior

- **retrans=n**

- Set the number of minor NFS timeouts per major NFS timeout
- Default is 4

- **timeo=n**

- Set the initial number of 0.1 second intervals in an NFS minor timeout
- Default is 7 (0.70 seconds)

These two parameters, if mismatched against the server and network performance, can severely impact the performance of an NFS client. Together they implement an exponential back-off algorithm that attempts to correct for UDP's unreliable nature. Performance can be exponentially bad if they are not set properly!

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Mount Parameters: BIOD buffer size

• **rsize=n, wsize=n**

- Set read/write buffer size to N bytes for BIODs
- Default is 8192 (8KB)
- Remember that mismatches between client system buffer size and file server block size can cause extra physical I/Os and poor performance in addition to the extra RPC requests and network loading
- Changes are most frequently seen with customers that are:
 - Trying to run NFS over a WAN (!)
 - Stuck with networking equipment that will not do proper MTU translation (i.e. FDDI ~4500 bytes -> Ethernet ~1500 bytes)
 - Have slow routers in the network that drop packets
 - Using PCNFS and have older interface cards and NFS client software for Windows

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning Tunable Kernel Parameters

- **ninode**
 - Clients can usually benefit from a larger inode table, to make LOCAL disk access more efficient.
 - There is no connection between the inode cache on the client
- **nbuf, bufpages, dbc_min_pct, dbc_max_pct**
 - Behavior is the same for dynamic buffer cache on the client as was previously described
 - NFS PV2 does not appear to use client-side buffer cache, but NFS PV3 can use up to 25% of the client's buffer cache
 - The overall effect of NFS PV3 on client buffer cache is unknown at this time

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning Tunable Kernel Parameters

- `page_text_to_local`
 - Set to “1” to enable paging of code pages to local client swap area, instead of reloading them across the network
 - The executable file must have the “sticky” bit set on the NFS server
 - “`chmod 1555 <executable file>`”
 - Can reduce network traffic, especially if the client is running large applications or is short on RAM
 - Will increase the use of the local disk, so multiple swap areas on multiple devices are a real good idea

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning Number of Idle BIOD Processes

Source: GPM-> Reports->Network Info->NFS Global Activity

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning Changing the Default Number of BIODs

- The variable controlling the number of BIOD processes is “NUM_NFSIOD”, located in the “/etc/rc.config.d/nfsconf” file
- You can test and set the value from inside of a script with the “/usr/sbin/ch_rc” command
- “ch_rc” treats all of the data definitions in the /etc/rc.config.d directory as a database, values may be added, tested, and changed:
 - # ch_rc -l -p NUM_NFSIOD 4
 - # ch_rc -a -p NUM_NFSIOD=16
- This is a very useful command if you are configuring systems from inside of a script (as with Ignite-UX)!

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client 1 KB Read/Write Size

Kernel

Client File System Request

Client BIOD

NFS RPC Request

Network Device

- Client Read and Write Size Assumes wsize=1024 and rsize=1024
- Determined by client NFS mount options
 - If the NFS server file systems are configured for 8 KB blocks and 1 KB fragments and the client mounts the server with 1 KB read and write sizes, then a client write of 1 KB of data may cause over 30 server disk writes:
 - Write of the first 1 KB fragment followed by the file's inode, potentially followed by an indirect inode
 - To write the second 1 KB, the server reads the first 1 KB fragment, combines the two, writes the 2 KB to two adjacent fragments, followed by the file's inode and potentially the indirect inode
 - Continues until a full 8 KB block is written

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client 8 KB Read/Write Size

Kernel

Client File System Request

Client BIOD

NFS RPC Request

Network Device

- Client Read and Write Size
 - Assumes wsize=8192 and rsize=8192
 - Determined by client NFS mount options
 - If the NFS server file systems are configured for 8 KB blocks and 1 KB fragments and the client mounts the server with 8 KB read and write sizes, then a client write of 8 KB of data may cause 2 to 3 server disk writes:
 - 8 KB of data to the server's disk block
 - The file's inode
 - For large files, potentially an indirect inode
 - For best performance, client read/write block size should match the server's disk block size

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Client Tuning NFS Client Components

Kernel

Client File System Request

Client BIOD

NFS RPC Request

Network Device

- WRITES

- When an application performs a write to an NFS file, the kernel invokes the NFS “BIOD” (Block I/O Daemon)
- The BIOD blocks up the write requests into the block size specified at mount time (wsize=8192) for a “write behind”, I.e. the write is delayed until a full 8 KB can be written to the server

- READS

- When an application performs a read from an NFS file, the kernel invokes the NFS BIOD.
- The BIOD reads the required block (rsize=8192) and does a “read ahead” of the next 8 KB block

- How Many BIODs?

- If an attempt is made to read or write to an NFS file and the kernel finds all BIODs busy, then the read or write will cause an immediate NFS RPC to the server, increasing server load and network traffic
- The HP-UX default, 4 BIODs is too small, try 16 and monitor with GlancePlus

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning The server's NFS mount daemon

- **The current HP-UX mountd process is single-threaded**
 - The NFS lab is working on a new multi-threaded daemon for late 1999
- **The server's mount daemon must resolve client name information for each mount request**
- **Multi-threaded AutoFS on the client systems may make the server response situation worse due to increased client request rates**
- **If the name resolution service is slow (i.e. NIS) a performance bottleneck may result**
 - Alter nfsswitch.conf to use DNS and files for name resolution
 - Consider making the NFS server a caching-only DNS server using itself for name resolution

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Controlling Client Access: Exporting File Systems

- **Export only the file systems that clients are accessing, avoid “exporting the world”**
- **/etc/exports can control which clients can access data on the file server and how they can access it:**
 - ro
 - -rw=hostname[:hostname] ... (read-only for unspecified)
 - -access=client[:client] ...
 - -root=hostname[:hostname] ...
 - -anon=UID (default is -2, -1 disables)
 - -async
- **Format for this file is “directory -option,option...”**
 -

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: Monitoring Results

Monitor the current utilization of the important tables on the server, and note the historical “high water” marks for tuning purposes.

Source: GPM->Reports->System Info->System Tables Graph

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: Other Kernel Parameters

- `create_fastlinks`
 - Enables the placement of soft link text into unused space in the inode entry if set to “1”
 - The link text must be less than 60 bytes
 - Eliminates a logical/physical disk read for every link evaluation
 - File systems with this type of link cannot be mounted on systems that do not have the fast links enabled
- `default_disk_ir`
 - Default value for disk immediate reporting
 - Set to “1” to enable immediate reporting
 - Ensure UPS protection where data is critical

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: Other Kernel Parameters

- nfile (maximum open files)
 - Estimate with $20 * \text{number_of_clients}$, then monitor with GlancePlus
- nflocks (maximum file locks)
 - Estimate with $5 * \text{number_of_clients}$ then monitor with GlancePlus
- nproc (maximum processes)
 - Set to the maximum number of processes that can execute on the server at one time. The largest contributor might be the NFSD processes
-

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning The “syncer” Process

I/O Buffer Cache

Syncer

- The syncer is responsible for flushing “dirty” system data to the disk
- The syncer runs at 30 second intervals (default) and flushes super blocks, modified inodes (inode cache), and delayed block I/O (buffer cache) to disk
- The buffer cache is split into five separate areas by the syncer algorithm and each is serviced during the run interval)
- On a “normal” system this activity is all but invisible, but on an NFS file server, the syncer can cause visible pauses in the system activity due to the large size of the caches involved
- This is because it attempts to flush all “dirty” data to disk at once and runs at a relatively high priority
- The syncer interval can be adjusted by modifying the startup line in /sbin/init.d/syncer to add the new interval as the first parameter (see man page for syncer)
- Running the syncer more frequently can eliminate long system pauses on an NFS server at the cost of additional CPU cycles

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: Dynamic Buffer Cache

- **The SPEC_SFS_1 benchmark data indicate that the system performs better using a dynamic buffer cache than a fixed one**
- **Values for `dbc_min_pct` and `dbc_max_pct` can be 70% and 90% respectively, unless those values conflict with RAM needed for the kernel and applications like GlancePlus, NIS, etc.**
- **Monitor buffer cache hit rates and look for values in the high 90% range**
- **Do not attempt to correlate the GlancePlus “Cache Hits” with “Physical Reads” and “Logical Reads” in the disk report, as NFS code bypasses the read and write system calls that GlancePlus instruments**

I/O Buffer Cache

UDP

Syncer

I/O Buffer Cache

I/O Buffer Cache

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: Fixed Buffer Cache on 9.X

- **If using 128 KB per NFS IOP (or 128 MB per 1000 NFS IOP)**
 - Use “bufpages” to set the buffer cache size if it is static
 - Increment “bufpages” by 32,768 for every 128 MB increment
 - Remember that on 9.X there is a practical limit of 800 MB of buffer cache
 - Monitor buffer cache hit rate with GlancePlus look for high 90% range
 - Do not attempt to correlate the GlancePlus “Cache Hits” with “Physical Reads” and “Logical Reads” in the disk report, as NFS code bypasses the read and write system calls that GlancePlus instruments.

I/O Buffer Cache

UDP

Syncer

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: Buffer Cache

On 10.X Series 700 and 800, if both nbuf and bufpages are set to 0, then dynamic buffer cache is enabled (default). On 9.X Series 800, a fixed buffer cache of 10% of physical RAM is allocated in this case.

If nbuf is set, bufpages is calculated. If bufpages is set, nbuf is calculated. If both are set, then nbuf “wins.” In either case, a fixed buffer cache is used

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning DNLC Cache Hit rate

Source: GPM->Reports->Disk Info->Disk Report

DNLC hits indicates the cumulative number of path components that were found in the DNLC cache. DNLC longs indicates the number of components that were not found because they were longer than 15 characters

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: ninode

Source: GPM->Reports->System Info->System Tables Report

Dynamic Name Lookup Cache keeps frequently accessed path components and their inode table location in memory. The DNLC cache size is tied to “ninode” and may not be independently set unless a patch is installed that introduces the “ncsize” kernel parameter. Path components longer than 15 characters are not cached.

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: JFS inode cache

- ninode (inode cache size)
 - The HFS and JFS inode caches are separate entities (ninode does not directly affect the size for JFS)
 - JFS prior to HP-UX 10.20 determines the size of the JFS inode cache based on memory size
 - However, if the value for ninode is larger than the table value, then the value for ninode is used as the minimum size of the JFS inode cache also
 - HP-UX 11.X introduced the “vx_ninode” parameter for explicitly adjusting the size of the JFS inode cache

I/O Buffer Cache

HFS Inode Cache

JFS Inode Cache

JFS Inodes

Memory Size

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tunable Kernel Parameters: HFS inode cache

- **ninode (inode cache size)**
 - To access a file's data, the server must first bring the file's inode into the memory resident inode table entry, which is about 240 bytes
 - The inode cache can help prevent reading or re-reading an inode whose data is already in buffer cache
 - The average number of memory resident inodes required for optimum performance will vary depending on the average working set size
 - As a general rule of thumb, there should be 1 to 2 times as many inodes as there are 8 KB blocks in the buffer cache
 - Every 128 MB of buffer cache is 16,000 inode table entries using this rule
 - Careful on multi-CPU machines with 10.X, there is a potential spinlock contention for the inode table

I/O Buffer Cache

Inode Cache

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Asynchronous NFS Mounts

- Benefits of NFS Asynchronous Mounts
 - NFS client continues processing as soon as NFSD has data
 - More NFS IOPS from server
 - Improved client response time
- Risks of NFS Asynchronous Mounts
 - Data is reported to the client as being committed when it isn't and is susceptible to loss
- Solution is to use battery backup for the RAM containing asynchronous NFS data
- HP-UX systems support UPS systems for CPU, RAM, and disks
- Note that NFS PV3 has safe asynchronous writes

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Asynchronous NFS Mounts

- **A client request must go through buffer cache, and usually a disk queue, make it to the disk's cache before the server can acknowledge the client request**
- **HP-UX supports asynchronous NFS write to cut this write path even shorter**
 - Prior to placing client data in buffer cache, the HP-UX NFSD immediately reports the data as written, releasing the client much sooner in the write path
 - File systems must be exported with the `-async` option in the `/etc/exports` file

I/O Buffer Cache

UDP

Syncer

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Disk Caching

- **NFS clients halt processing until the NFS server acknowledges that write requests are posted to the disk**
- **HP-UX supports disks with immediate reporting mode (on-controller cache) and RAID arrays that contain write cache**
 - When NFS data is written from buffer cache to the disk, the disk immediately reports the data has been written, even though it hasn't
 - The server then acknowledges the write to the NFS client
 - # scsictl -a /dev/rdisk/c0t6d0 immediate_report = 1 ; queue_depth = 8
 - # scsictl -a /dev/rdisk/c0t10d0 immediate_report = 0 ; queue_depth = 2
- **Benefits: speeds disk requests, disk may do more ops/sec**
- **Problems: data may be lost on power failure**
- **SYSTEMS MUST HAVE UPS FOR SYSTEM AND DISKS TO USE THIS FEATURE OR DATA WILL BE LOST**

10.20 Series 700

10.20 Series 800

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning NFS File Server Resource Contention

- **The easiest way to impact ALL users in an environment is to slow down the NFS file server with CPU or I/O intensive activities**
- **DO NOT RUN APPLICATIONS ON THE NFS SERVER**
- **The exception to this rule are “well-behaved” client-server applications that cause minimal impact, for example:**
 - NIS
 - DNS
 - X11 Font Server
- **If sharing NFS file server resources with other activities is unavoidable, then consider using Process Resource Manager to guarantee resource levels**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Initially Sizing Buffer Cache

- **A popular approximation is 128 KB of buffer cache per NFS IOP the server will provide**
- **Note that HP-UX 9.X is limited to approximately 800 MB of buffer cache, this limitation was removed in 10.X**
- **It may be impossible to keep the buffer cache hit rate as high as might be desired, in that case:**
 - Consider splitting the NFS server and the files it provides into two machines
 - Make sure the disk I/O system is capable of sustaining heavy I/O rates by selecting the proper interfaces and striping the data

I/O Buffer Cache

Syncer

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Measuring Buffer Cache Working Set Size

Source: GPM->Reports->Disk Info->Disk Report

Read cache hit rates in the high 90s is a sign of a server that is doing a good job caching the client's working set

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Measuring Buffer Cache Working Set Size

- **The ideal file server would service all requests as fast logical reads instead of slow physical reads**
- **For this to happen, there needs to be enough buffer cache to contain all of the data that the server's clients are using**
- **This data is the “working set”**
- **For most situations, the buffer cache should be sized to handle 85-95% of the working set size, since the required cache might be very large otherwise**
- **For existing servers, the buffer cache hit rate is reported by GlancePlus and can be used to adjust the size of the buffer cache**

I/O Buffer Cache

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Dedicate Buffer Cache to NFS

- **The performance bottleneck in most servers is usually the disk subsystem -- disk heads move slowly**
- **Buffer cache eliminates the slow physical I/O by keeping file system blocks in RAM**
- **Logical reads from buffer cache are much faster than physical reads from the disk**
- **There are two important criteria regarding buffer cache:**
 - Have enough of it
 - Avoid situations that will flush its contents
- **DEDICATE YOUR NFS SERVER BUFFER CACHE**

I/O Buffer Cache

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning NFS UDP Socket Overflow

- **If there are not enough NFSDs to keep up with the incoming RPCs,**
 - The UDP socket will overflow
 - NFS client requests will be dropped
 - Client systems will retry
- **The number of times the UDP socket has overflowed can be determined using the command: `$ netstat -s | grep overflow 1120 socket overflows`**
- **This assumes that the major user of UDP services on the machine is NFS, which is not always the case**

Kernel

NFSD

UDP

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning NFS UDP Socket Depth

- **Client NFS RPC requests are received into the UDP socket on port 2049**
- **The UDP socket has a 256 KB FIFO that is emptied by the NFSD processes**
- **If the UDP socket continually has data in it, it is a sign that there are not enough NFSDs to keep it serviced**
- **There will be times when there will be data in transit, so the average queue length is probably of more interest**
- **The number of bytes in the UDP receive buffer can be displayed using the command: `$netstat -an | grep -e Proto -e 2049 Proto Recv-Q Send-Q Local Address Foreign Address (state) udp 0 0 *.2049 *.*`**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Measuring NFSD Usage

- The HP-UX 9.X goal is to have just enough NFSDs to service the incoming RPCs, this is handled automagically on 10.X
- The number of NFSDs needed can actually be measured on an HP server using the command “nfsstat -s”

- \$nfsstat -s Server rpc: calls badcalls nullrecv badlen xdrcall nfsdrun 2335351 5333280
- \$nfsstat -s Server rpc: calls badcalls nullrecv badlen xdrcall nfsdrun 2335351 2335351

9.X

10.X

5.3 M NFSDs started to service 2.3 M RPC calls, the number of NFSDs could be cut in half. Remember that this is an average, and more NFSDs might be needed during peaks.

One to one ratio on 10.X is ideal behavior

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Changing the Default Number of NFSDs

- The variable controlling the number of NFSD processes is “NUM_NFSD”, located in the “/etc/rc.config.d/nfsconf” file
- You can test and set the value from inside of a script with the “/usr/sbin/ch_rc” command
- “ch_rc” treats all of the data definitions in the /etc/rc.config.d directory as a database, values may be added, tested, and changed:
 - # ch_rc -l -p NUM_NFSD 4
 - # ch_rc -a -p NUM_NFSD=32
- This is a very useful command if you are configuring systems from inside of a script (as with Ignite-UX)!

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning How many NFSDs Do I Start?

- **A common “rule of thumb” is that a server should have 2 NFSDs per disk (or logical disk). One NFSD to receive a request and move the disk head, and a second queuing up the next RPC that will move the disk head (double buffering)**
- **In practice, there is no correlation between NFSDs and disks:**
 - If the server has adequate buffer cache, the file data being requested may will be in RAM and not require disk access
 - If the server had an adequately sized RAM resident inode table, the inode data being requested may well be in RAM and not require disk access
- **It turns out that this is still a good first order approximation for a starting point**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning HP-UX 10.X Issues with NFSD

- There appears to be no cost in starting too many NFSDs on HP-UX 10.X systems (other than the small amount of process space and the proc table entries)
- Regardless of how many NFSD processes are available, only a single NFSD is awakened for each RPC that arrives at the server
- This allows a site to start the maximum number of NFSD processes that should be needed simultaneously, knowing that only the actual number required will be awakened
- The V-Class NFS IOP number was obtained with 270 NFSD processes!

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning HP-UX 9.X Issues with NFSD

- **Starting an excessive number of NFSDs may have a negative impact on server performance**
- **If 32 NFSDs are available, and a single NFS RPC comes into the server, then all 32 NFSDs will be place on the server's run queue**
- **The HP-UX scheduler will context switch the first NFSD, which will service the incoming RPC, and the other 31 NFSDs will successively run, find no RPC to service, and then go back to sleep**
- **The cost of context switching the NFSDs may limit performance over about 30 instances of NFSD**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning Tuning Network Components

- **Fileserver LAN parameters are largely untouchable (except MTU size)**
- **NFS RPCs arrive in the NFS UDP socket, number 2049**
- **The UDP socket is emptied by NFSDs**
- **The socket buffer is a FIFO, with 256 KB of buffer**
- **Server needs enough NFSDs to keep the UDP socket “drained” (NFSDs can be a bottleneck)**
- **Too few NFSDs will not keep the socket emptied**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning NFS Fileserver Read Components

UDP

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning NFS Fileserver Write Components

UDP

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Tuning NFS Client/Server Components

Syncer

I/O Buffer Cache

Kernel

NFSD

“The Network”

UDP

Network Interfaces

Kernel

Application

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning On-Line JFS operations: After Disk/Directory Defragmentation

No changes

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning On-Line JFS operations: Examining Fragmentation

1 KB Blocks to reclaim

Directories that are or could be contained only in the inode data space

Source: fsadm_vxfs(1M) man page

Directories that may be reduced in size

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning On-Line JFS operations

- **Examining JFS file attributes**

- “gettext -F vxfs -D <file>”

- **Reserving space for a JFS file (On-Line JFS only)**

- “setext -F vxfs -e <extent size> -r <blks> -f contig -f trim”

- **Checking dir/ext fragmentation (On-Line JFS only)**

- “fsadm -F vxfs -E -D <mount point>”

- **Defragmenting files, sorting/compacting dirs**

- “fsadm -F vxfs -e -d <mount point>”

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning JFS mount options

• **nodatainlog**

- Normally small synchronous data is put into the log (inode and write), requiring only one disk access to update. Disables that behavior: 50% slower than standard behavior.

• **blkclear**

- Used in increased security environments to ensure that uninitialized storage never appears in files.

• **mincache=[closesync, direct, dsync,unbuffered,tmpcache]**

- closesync synchronize file data on file close (15% slower)
- direct, dsync, unbuffered Alter the system's buffering behavior (see man page)
- tmpcache Trades less integrity for better performance (Temp file systems)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning JFS mount options

• log (default)

- JFS guarantees that all structural changes to the file system have been logged on the disk when the system call returns.

• delaylog

- Some system calls return before the log is written. This logging delay improves performance, but some changes are not guaranteed until a short time after the system call returns. This mode approximates the traditional UNIX file system behavior

• tmplog

- Logging is almost always delayed. Only recommended for temporary file systems

• nolog

- Logging is disabled. Fast recovery does not work with this mode, potential for data loss is greater than with UFS/HFS

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning JFS File System

- **Veritas® recommends that the log size be increased for synchronous or NFS workloads to improve performance. This will slow down recovery time.**
 - “mkfs -F vxfs logsize=n” (see man page for mkfs_vxfs)
- **The number of inodes is dynamically adjusted as files are created, unlike HFS**
- **Mount options and application calls may specifically alter the performance characteristics of the file system (see man page for vxfsio)**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning JFS File System

- **Journalized File System (VxFS) keeps a log of structural changes being made to the file system**
- **The internal structure is different from HFS**
- **When system failures occur, the log is replayed to fix the file system, the repair time is dramatically shorter than running the “fsck” command for HFS**
- **Logging the “meta” data impacts performance**
- **JFS may be defragmented in place if the “Online JFS” product is purchased**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning Walking Directory Paths

/local_home/robl/Tools/SystemAdmin/PatchTools/LoadFilesets/README

Each level in the path involves inode and file I/O: Read directory file inode from disk Read directory data from disk ... Read file inode from disk Read file data from disk Close file: Update data and inode on disk

Each symbolic (soft) link in the path can contribute two disk I/Os: Read soft link inode Read soft link data

Inserting NFS into this scenario means that each operation becomes one or more RPC calls (NFS IOPS) to the remote server. Network latency and NFS server performance become the major performance components

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning HFS File System Structure

File System inodes

File System Data Blocks

Two Related Trees of Data Structures

This will be important later, when we talk about buffer cache and inode cache

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning HFS File System Structure

File Mode/Type Number of links Owner UID Owner GID File Size Last Access Time Last Modified Time Last Inode Change Indirect Pointers

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning HFS File System Structure

File Mode/Type Number of links Owner UID Owner GID File Size Last Access Time Last Modified Time Last Inode Change

8 KB File System Block

1 KB File System Fragment

inode Information

Direct Block Pointers

Indirect Block Pointers (2 single, one double)

This space (60 bytes) may be used for symbolic link information if the kernel parameter “create_fastlink” is set to 1

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning HFS File System

- There are several “issues” with HFS and NFS
- The ENTIRE buffer cache must be searched on file deletes or closes (unlike JFS, no “dirty” block list is maintained)
- Performance degrades for both clients and servers that use HFS, based on buffer-cache size
- DO NOT EXPORT HFS FILE SYSTEMS VIA NFS
- MINIMIZE THE USE OF HFS ON CLIENT SYSTEMS

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning HFS File System

- **HFS (also known as UFS) has been around a long time**
- **The file system structures are designed for multiple platter and multiple head disk mechanisms**
- **System failures can require hours of file system recovery via “fsck”, since the entire structure must be checked**
- **The file system may only be defragmented by copying from one volume to another with “dcopy”**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Tuning File System Time Line

HP-UX 10.00

HP-UX 10.01

HP-UX 10.10

HP-UX 10.10

HP-UX 10.20

HP-UX 11.00

HFS

JFS Version 2

JFS Version 3

Large File System Support (128 GB)

Large File Support Large UID/GID Support

JFS Version 2 file systems are upwards compatible, but JFS Version 3 may not be mounted on previous HP-UX versions. The default type for creation is JFS Version 3. JFS Version 2 file systems may be upgraded to JFS Version 3 with the “vxupgrade” command.

JFS Version 3.3

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Sizing Mass Storage Considerations: Fibre Channel

- **Storage Area Network: SAN (is coming)**

SAN

WAN

LAN

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Sizing Mass Storage Considerations: Fibre Channel

- **Note that Gigabit Ethernet uses the Fibre Channel physical layer**
- **Fibre Channel is capable of carrying other protocols**
- **Switched Fibre Channel is used in the EPS systems for high-speed (266 Mbps) CPU to CPU Hyperfabric™ connections**

Up to nine disk arrays may be connected to a hub. Each disk array may have 8 separate LUNs. Each LUN is a separate file system.

Up to 2 Km

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Sizing Mass Storage Considerations: Fibre Channel

- **HP supports two arbitrated loop topologies: direct connect and hub connect**
- **Maximum of two cascaded hubs per F/C interface**
- **Fibre Channel uses “fair” arbitration to avoid starving devices (as in the SCSI 6-0,15-8 priority scheme)**
- **Hub connect is implemented with short-wave and long-wave Fibre-Channel hubs**
 - Short-wave
 - Supports up to 500 m between devices
 - Nine short-wave and one long wave ports per hub
 - Long-wave
 - Supports up to 2 Km between devices
 - 10 short-wave ports per hub
 - Sum of lengths of all links < 5 Km
- **Fibre Channel to SCSI MUX has up to four F/W SCSI ports supporting up to 15 devices per port, and up to two short-wave F/C adapters**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Sizing Mass Storage Considerations: Fibre Channel

- **Fibre Channel comes in three topologies:**
 - Point-to-point
 - Arbitrated loop
 - Switched
- **HP's Fibre Channel implementation is arbitrated loop, 100 MB/S half duplex and 200 MB/S full duplex (1063 Mbps per link)**
- **Fibre Channel interfaces are only available for PCI or HSC bus connections**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Sizing Mass Storage Considerations: Channel Performance

1There are only 4 expansion slots per HSC bus converter 2The HSC bus is limited to 112 MB/S sustained throughput

3We must wait for bus expansion to fully use one of these channels

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

File Server Sizing Mass Storage Considerations: Channel Performance

In reality, the maximum transfer rate on the SCSI bus for single device operations may be limited by the maximum transfer rate of the SCSI device. For example, the 7200 RPM disks can transfer 5.9 - 9.0 MB/S. Note that higher priority devices (larger SCSI ID) may also starve lower priority devices.

**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: AutoRAID

- **AutoRAID: Performance without the RAID trade offs**
 - Dynamic data migration between RAID 0/1 and RAID 5 based on access patterns
 - Frequently accessed data kept in RAID 0/1, infrequently accessed data moved to RAID 5
 - RAID mode cannot be specified for LUNs
 - Data location (RAID 0/1 or RAID 5) cannot be determined at any given time
 - Model 12 Model 12H 24 MB memory per ctrlr 96 MB memory per ctrlr 19.5/39.5 GB capacity 37.8/82.3/166.4 GB capacity Supports 2.1/4.3 GB disks Supports 4.3/9.1/18.2 GB disks

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: XP-256

Cache

Memory

i960 i960 i960 i960

CHIP

i960 i960 i960 i960

CHIP

i960 i960 i960 i960

CHIP

i960 i960 i960 i960

CHIP

Shared

Memory

17 GB to 9 TB of Storage Up to 32 SCSI channels Up to 64 FC channels 256 MB to 16 GB of cache

Two 240MB/sec Data Buses Two 130MB/sec Control Buses

High speed 12000 RPM super-high reliability disks

15GB & 36.9GB

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: FC-60 RAID Array

1.6m Rack = 1 FC60 + 8 SC10 = 1.31 TB (18 GB disks) 2 FC60 + 7 SC10 = 655.20 GB (9.1 GB disks)

2.0m Rack = 1 FC60 + 10 SC10 = 1.638 TB (18 GB disks) 2 FC60 + 9 SC10 = 819 GB (9.1 GB disks)

Controller A

Controller B

Write Cache

Write Cache

Fibre Channel 100 MB/S

Fibre Channel 100 MB/S

0

1

2

3

4

5

LVD Ultra2 SCSI bus 0-6 (40 MB/s each)

256 MB of cache per controller Supports RAID 1, 0/1, and 5 One or two FC60 controllers

Aggregate performance 170 MB/s Real-time hardware monitoring and event detection

Battery Backup Unit

Battery Backup Unit

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: Model 30/FC RAID Arrays

SCSI B

RAID modes 1, 1/0, and 5 30 Disk slots 115/247/498.4 GB capacity with 4.2/8.8/17.8 GB Disks
Up to 64 MB of write cache per controller Fast Wide Differential SCSI, 20 MB/S

Battery Backup Unit

Service Processor A

Service Processor B

Power Supply

Power Supply

Write Cache

Write Cache

Fibre Channel 100 MB/S

Fibre Channel 100 MB/S

RS-232

RS-232

SCSI A

SCSI C

SCSI D

SCSI E

Power Supply

Fan Units

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: Model 20 RAID Arrays

SCSI B

RAID modes 1, 1/0, and 5 20 Disk slots 73.8/153.8/320.4 GB capacity with 4.2/8.8/17.8 GB Disks
Up to 64 MB of write cache per controller Fast Wide Differential SCSI, 20 MB/S

Battery Backup Unit

Service Processor A

Service Processor B

Power Supply

Power Supply

Write Cache

Write Cache

F/W/D SCSI

F/W/D SCSI 20 MB/S

RS-232

RS-232

SCSI A

SCSI C

SCSI D

SCSI E

Fan Units

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: Model 10 RAID Arrays

SCSI A

SCSI B

RAID modes 1, 1/0, and 5 10 Disk slots 36.9/79.4 GB capacity with 4.2/8.8 GB Disks Up to 16 MB of write cache per controller

Battery Backup Unit

Service Processor A

Service Processor B

Power Supply

Power Supply

Write Cache

Write Cache

F/W/D SCSI 20 MB /S

F/W/D SCSI

RS-232

RS-232

Fan Unit

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: RAID Modes

• RAID 3

- Disk striping, dedicated parity drive
- Combines 5 drives to act as a large virtual drive
- RAID 3 provides poor performance for any multi-user random access activity
- 25% storage overhead

• RAID 5

- Disk striping, parity data distributed for performance
- Independent data paths to drives
- RAID 5 provides redundancy, but poor write performance for multi-user random-access activity
- RAID 5 provides good performance for read-mostly file systems like (/usr, /opt, /home)
- RAID 5 supports 3-16 drives in a group, 5 is typical

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: RAID Modes

• RAID 1

- Disk mirroring
- Drives operate asynchronously
- Provides redundancy with 100% storage overhead

• RAID 0/1

- Disk striping and mirroring
- Provides best performance of RAID modes
- Provides redundancy with 100% storage overhead

\$\$

\$\$

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: Striping Stand-Alone Disks

• Striping Stand-Alone Disks with LVM

- Avoids exceeding the throughput of a single disk by spreading the I/O over multiple devices
- Stripes may be made across channels, as well as across devices
- Supported LVM Striping:
 - “lvcreate -D y -s g -i <# pvols> -I <stripe size>”
 - Stripe size can be 4, 8, 16, 32, or 64 KB
 - Cannot be mirrored
- “Extent striping”
 - “lvcreate <no size parameter>”
 - “lvextend -L <size + 1 extent> ... <lv_name> <physical device>”
 - Can be mirrored
 - Stripe is limited to Volume Group extent size (minimum 1 MB)
 - Long set up time (each extend takes about 1 second)
 - There are tools to automatically generate a script to perform the operations

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: SC-10/FC-10

Fan Unit

3.5 EIA Units, 10 Disks per enclosure [11 enclosures per 2m rack] 2m Rack = 110 disks = 0.99 TB (9.1 GB disk) or 1.98 TB (18 GB disk) Loop Control Card (LCC) Bus Control Card (BCC) Fibre Channel 100 MB/s (FC-10) [Dual FC to each disk] LVD Ultra2 SCSI 40 MB/s (SC-10) Hot Plug Disks, Power Supplies, Fans, LCC/BCC Real-time hardware monitoring and event detection

LCC/BCC B

Power Supply

Power Supply

Expansion Fibre Channel or LVD Ultra2 SCSI

LCC/BCC A

Primary Fibre Channel or LVD Ultra2 SCSI

Fan Unit

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: HP Smart Enclosures

Power Supply

Power Supply

Power Supply

Power Supply

Controller B

Controller A

Primary Groups

Mirror Groups

- Controllers are GSC F/W SCSI or Fibre Channel I/Fs
- No single point of failure
- F/W or S/E devices
- Can be used with FC SCSI Mux
- Enclosures can be “split in half”
- Hot-plug disks, fans, power supplies

Single “HP Smart” Enclosure

Single “HP Smart” Enclosure

Fan Units

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: Mirroring Stand-Alone Disks

• Mirroring Stand-Alone Disks

- 100% storage overhead (Must duplicate space for mirror)
- Performance benefit: Reads will come from least busy mirror
- Cannot mirror and stripe at the same time (more on this later)
- MirrorDisk/UX now available for 700 and 800 systems
- Writes are duplicated to each mirror group (2X the I/O)
- Supports up to 2 mirrors
- Mirror groups may be “split” and “joined” and will synchronize data

Primary Groups

Mirror Groups

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations: Stand-Alone Disks

- **Also known as JBOD (Just a Bunch Of Disks)**

- Provide good performance but no redundancy
- Performance from lots of independently moving heads
- Use MirrorDisk/UX and LVM to gain redundancy
- Stripe stand-alone disks with LVM to gain performance
- Can quickly run out of expansion ability because of limited I/O channels

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Mass Storage Considerations

- **Areas to consider for fileserver mass storage:**

- Storage Technology
 - JBOD
 - RAID Modes and Arrays
 - AutoRAID
- Disk I/O Channel Performance
 - SCSI
 - Fibre Channel

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Memory Subsystem

- **System Memory is needed for 3 primary reasons:**
 - Fileserver kernel
 - Assume 32-50 MB for kernel, data structures, network memory, root login, and GlancePlus
 - Fileserver Applications
 - Add memory for additional administration and performance tools
 - Add memory for additional server processes like NIS, DNS, SDU, X11 Font Server, etc.
 - Remember that any large, CPU or file-intensive applications may damage NFS fileserver performance
 - Fileserver I/O Buffer Cache
 - Start with ~128 KB for each NFS IOP (128 MB per 1000 NFS IOPs)
 - Measure and tune from this starting point

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Examining the Network: Specific Notes

- **Cabletron equipment appears to have issues with properly fragmenting an FDDI packet that is heading for an ethernet segment. If you are getting unexplained errors or retries, try lowering the FDDI MTU size to 1500 bytes to see if they disappear**
- **Cisco switches appear to drop UDP traffic under heavy loads or burst conditions. If you are seeing unexplained NFS retries, then try enabling some backup buffering in the switch. For software revisions 5.3 and higher, there is an undocumented command: “set option flowcontrol enable” that will allow additional buffering for UDP traffic.**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Examining the Network: Looking for Errors

Source: GPM->Reports->Network Info-> Network Graph

Look for low error and collision rates. There are low collision rates here because the link is a dedicated switch port. But should full-duplex 100baseT EVER have collisions? NO!

If the error rate is greater than zero, start looking at network interfaces, cabling, hubs, switches, etc. and check that you are really running full-duplex.

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Examining the Network Interface: landiag

Results of “landiag->l” then “nmid 6” then “d”

Statistics for server's 100baseT link

Packets can be:

Unicast specific host

Multicast multiple hosts

Broadcast everybody on the LAN

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Examining the Network Interface: lanscan

Use the lanscan command to see all network interfaces in the server. The Network Management ID (NMID) can be used with landiag to examine lower-level statistics, including the MTU size, the connection rate, framing errors, etc.

MAC (Media Access Control) Address or LLA (Link Level Address)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing But what about “traditional” routers and ATM?

- **Traditional routers are still needed for security and access to the WAN**
- **ATM is complicated, expensive, and requires LANE, but is useful for interfaces at the edge of the LAN or where one technology is needed for LAN, WAN, voice, data, and video**

Collision Domain

Broadcast Domain

L2

L2

L3

WAN Interface

The Internet or WAN

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Link Aggregation (Trunking)

• Link Aggregation Solves the “Big Pipe” Problem

- Parallel active links
- Load sharing
- Resiliency
- Spanning Tree algorithm issues
- No standard, yet (IEEE 802.3AD)
- 3Com allows 6 links in a trunk, up to 4 trunks per switch

6 GbE Links = 6 Gbps 196,608 NFS IOPS

3 GbE Links = 3 Gbps 98,304 NFS IOPS

3 FE Links = 300 Mbps 9600 NFS IOPS

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Layer 3 Switching Technology

• Layer 3 Switch Characteristics

- Optimized for LAN routing
- Wire-speed (Mpps) forwarding
- Forwards inter-subnet traffic based on Network layer addresses (layer 3)
- Forwards intra-subnet traffic based on MAC addresses (layer 2)
- Contains the flooding of broadcast traffic

Collision Domain

Broadcast Domain

L2

L2

L2

L2

L3

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Layer 2 Switching Technology

LAN Collision Domain

Before Installing a Switch

After Installing a Switch

Collision Domain

Broadcast Domain

Switching with VLANs

- Switches may be used to reduce the size of collision domains
- Switches will propagate (flood) broadcasts to all ports
- Virtual LANs (VLANs) can determine which ports get which broadcasts
- Warning: Look for “oversubscribed” backplanes in switches!

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Layer 2 Switching Technology

• Switching Technology is the way to go for LANs

- Smart learning bridges that make independent hardware connections between ports at the MAC layer
- High throughput and low latency compared to routers
- Create Smaller Collision Domains
- Simple, easy to understand, “Plug-and-Play”
- Protocol Transparent
- Low price per port

High-Speed Uplink(s)

Shared LAN

Shared LAN

Dedicated Workstations

100baseTX

10baseT

Dedicated Servers

100BaseFX GbE FDDI ATM

Intelligent Switching Matrix

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Ethernet Switching: “Switch the LAN, route the WAN”

• Ethernet Switching Replaces Multiple Network Interfaces in Servers

- Ethernet switch is 3X-10X router performance at 30% of the cost
- Ethernet switching extends useful life of Ethernet investment
- Single network investment (switch) is available to all servers
- Need a high-speed link (“large pipe”) between servers and switches
- Switches handle FDDI, 10baseT, 100baseT, and other network media
- Reduction in system administration due to moves, route configuration

Switch

One Ethernet Subnet Multiple Segments

Separate Ethernet Subnets

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Routed Ethernet Backbone

• Routed Ethernet Backbones are Dead (1980s Architecture)

- Routers are inappropriate for the I/O rates of network OS
- Routing is not necessary; need more segments not subnets
- Router latency and bandwidth becomes a system bottleneck
- Increasing the number of segments yearly is expensive using routers

Router

Ethernet Backbone

Ethernet Subnets

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: ATM OC12 (622.08 Mbps)

• ATM OC12 Loading Rules

- ATM OC12 = 622 Mb/S; 622 Mb/S = 81.53 MB/S (Maximum theoretical)
- 81.53 MB / 52 bytes per ATM Cell = 1,567,822 Cells/S
- 1,567,822 Cells/S x 48 data bytes per Cell = 75.26 MB/S (Maximum theoretical ATM saturation)
- 75.26 MB / 4 KB per NFS IOP = 18,373 NFS IOPS

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: ATM OC3 (155.52 Mbps)

• ATM OC3 Loading Rules

- ATM OC3 = 155.52 Mb/S; 155.52 Mb/S = 19.44 MB/S (Maximum theoretical)
- 19.44 MB / 52 bytes per ATM Cell = 392,006 Cells/S
- 392,006 Cells/S x 48 data bytes per Cell = 17.94 MB/S (Maximum theoretical ATM saturation)
- 17.94 MB / 4 KB per NFS IOP = 4594 NFS IOPS
- ATM has no collisions
- The MTU size for ATM may be 9000 bytes (!)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: Is GbE == ATM?

- IEEE 802.3Z Gigabit Ethernet
- IEEE 802.1P Traffic Priority Tagging
- IEEE 802.1Q Virtual LAN Tagging
- IEEE 802.3X Full Duplex Flow Control
- IEEE 802.3AD Link Aggregation
- Switches Readily Available
- No LANE needed!
- Original Ethernet MTU size retained (~1500 bytes)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: Gigabit Ethernet (GbE)

• Gigabit Ethernet (GbE) IEEE 802.3Z

- 1000Base-SX 62.5 μ m Multi-mode Fiber 260 m 50 μ m Multi-mode Fiber 550 m
- 1000Base-LX 62.5 μ m Multi-mode Fiber 440 m 50 μ m Multi-mode Fiber 550 m
Single-mode Fiber 3 Km
- 1000Base-CX Twinax or Quad 25 m
- 1000Base-T Category 5 UTP 100 m

Short wavelength

Long wavelength

4 conductor coaxial cable

Unshielded Twisted Pair

Transceiver

Transmission media

Distance

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: 100baseTX Collision Domains

• Fast Ethernet Collision Domain Loading Rules

- 100baseTX = 100 Mb/S; 100 Mb/S = 12.5 MB/S (Maximum theoretical)
- 100baseTX = 12.5 MB/ 4 KB per NFS IOP = 3200 NFS IOPS (Maximum theoretical 100baseTX saturation)
- Is 100baseTX = 10X the performance of 10baseT? (experience so far shows between 5-7X the performance)
- 100baseTX SHARED = up to 1000 NFS IOPS
- Notice that FDDI is still better at moving data than 100baseTX, due to no collisions and larger MTU size (4500 bytes versus 1500 bytes)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: FDDI Token Ring

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: FDDI Token Ring

• FDDI Segment Loading Rules

- FDDI = 100 Mb/S; 100 Mb/S = 12.5 MB/S (Maximum theoretical)
- FDDI = 12.5 MB/ 4 KB per NFS IOP = 3200 NFS IOPS (Maximum theoretical FDDI saturation)
- Studies show FDDI performance breaks down at 80% utilization
- Therefore, don't plan to load FDDI at more than 80% (2500 NFS IOPS)
- Leave 50% headroom for growth and interactive traffic (e.g. telnet, X-windows, ftp, etc.)
- Maximum Operating NFS IOPS per FDDI = 1200 IOPS (about 12 10baseT ethernets)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: 10baseT Ethernet Utilization Curves

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Loading: Ethernet Collision Domains

• Ethernet Segment Loading Rules for 10baseT

- 10baseT Ethernet = 10 Mb/S; 10 Mb/S = 1.25 MB/S (Maximum theoretical)
- 10baseT Ethernet = 1.25 MB/4KB per IOP = 300 NFS IOPS (Maximum theoretical Ethernet saturation)
- Studies show that interactive X-Windows clients suffer when Ethernet usage exceeds 25%
- Therefore: Don't plan to load Ethernet more than 25-30% (100 NFS IOPS)
- Headroom should be left for interactive traffic beyond the NFS traffic (e.g. telnet, X-Windows, ftp, etc.)
- Maximum NFS IOPS per Ethernet LAN = 100 IOPS (This will still annoy people with poor interactive response)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Design: Network Technologies

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Network Design

"There is no network from the user's point of view. There are only services. The goal of all network design and management is to preserve this misunderstanding."

INTEROP fall 1991 Scott Bradner Harvard University

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing High Availability NFS

Highly reliable systems highly reliable hardware high quality OS predictive support up time > 99.8%

Data protection disk array

Data availability mirrored arrays

System availability mirrored arrays redundant systems

High Availability Features: Dynamic page deallocation JFS/LVM Hot-Pluggable Disks RAID Arrays Mirrored Disks Multiple I/O paths Integrated UPS MC/ServiceGuard

Increasing Availability

System and Network Management Tools 24x7 Support Services Disaster Recovery Services

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

L2000 System Architecture

Memory bus 1.3 GB/s

ECC Memory 256 MB to 8 GB

System Bus 1.3 GB/s

I/O

C

o

n

t

r

o

l

l

e

r

6 Independent channel, hot-plug PCI slots

250 MB/s each

Core

I/O

Ultra2 SCSI Bus (independent of internal disks) 3 RS-232 Ports 10/100BaseT LAN LAN, Remote, Local Console Ports Ultra SCSI Bus 0 and 1 (4 internal hot plug disks) Removable media bay (DVD or DAT)

1 to 4 PA-8500 CPUs 360 or 440 MHz

4 shared channel PCI slots

250 MB/s

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

L1000 System Architecture

Memory bus 1.3 GB/s

ECC Memory 256 MB to 4 GB

System Bus 1.3 GB/s

I/O

C

o

n

t

r

o

l

l

e

r

5 Independent channel, hot-plug PCI slots

250 MB/s each

Core

I/O

Ultra2 SCSI Bus (independent of internal disks) 3 RS-232 Ports 10/100BaseT LAN LAN, Remote, Local Console Ports Ultra SCSI Bus 0 and 1 (4 internal hot plug disks, 2 per channel) Removable media bay (DVD or DAT)

1 to 2 PA-8500 CPUs 360 or 440 MHz

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

N4000 System Architecture

Memory bus 7.6 GB/s aggregate

ECC Memory 512 MB to 16 GB

IA64 System Bus 0

1.92 GB/s

IA64 System Bus 1

1.92 GB/s

I/O

C

o

n

t

r

o

l

l

e

r

I/O

C

o

n

t

r

o

l

l

e

r

4 Twin Turbo PCI

480 MB/s each

2 Turbo PCI

NFS Fileserver Sizing

240 MB/s each

6 Twin Turbo PCI

480 MB/s each

480 MB/s

Core

I/O

Ultra2 SCSI Bus (independent of internal disks) RS-232 Port 10/100BaseT LAN LAN, Remote, Local Console Ports Ultra SCSI Bus 0 and 1 (internal hot plug disks)

2 PA-8500 CPUs

1 to 8 PA-8500 CPUs 360 MHz or 440 MHz

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

V2200/2250 System Architecture

Synchronous DRAM 256 MB to 16 GB 32-way Interleaved

8 X 8 Port Non-blocking Hyperplane™ Crossbar 15.36 GB/S Total Memory Bandwidth
x 8

980 MB/S Each direction per port

x 8

980 MB/S Each direction per port

PCI Bus Converter 120 MB/S

PCI Bus Converter 120 MB/S

240 MB/S I/O Bus

2 PA-8200 CPUs and Cache

1 to 16 PA-8200 CPUs 200 MHz V2200 240 MHz V2250

Up to 2 GB/S Aggregate I/O Bandwidth (240 MB/S x 8)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

K570/80 System Architecture

ECC Memory 128 MB-- 4 GB

Memory Controller

Memory bus

766 MB/S Processor/Memory Bus

PA-8200 PA-RISC Processors

HSC Bus Converter 80 MB/S

Core I/O

S/E SCSI-2 F/W Diff. SCSI 802.3 LAN Modem UPS Port RS-232 Console

1 to 6 PA-8200 Processors 2048x2048 KB Cache per CPU 200 MHz 240 MHz K570 K580

HP-PB Bus Converter 20 MB/S

HSC Slot

Source: HP 9000 Enterprise Servers Configuration Guide 5/98

HSC Bus Converter 116 MB/S

ECC Memory 128 MB-- 4 GB

Memory Controller

Note: Either Extra CPUs -or- Extra I/O not both

HSC Bus Converter 116 MB/S

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

K460/EG/XP System Architecture

ECC Memory 128 MB-- 4 GB

Memory Controller

Memory bus

766 MB/S Processor/Memory Bus

PA-8000 PA-RISC Processors

HSC Bus Converter 80 MB/S

Core I/O

S/E SCSI-2 F/W Diff. SCSI 802.3 LAN Modem UPS Port RS-232 Console

1 to 4 PA-8000 Processors 1024x1024 KB Cache per CPU 180 MHz

HP-PB Bus Converter 20 MB/S

HSC Slot

Source: HP 9000 Enterprise Servers Configuration Guide 5/98

HSC Bus Converter 116 MB/S

ECC Memory 128 MB-- 4 GB

Memory Controller

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

ECC Memory 128 MB-- 4 GB

Memory Controller

Memory bus

766 MB/S Processor/Memory Bus

PA-8200 PA-RISC Processors

HSC Bus Converter 80 MB/S

Core I/O

S/E SCSI-2 F/W Diff. SCSI 802.3 LAN Modem UPS Port RS-232 Console

1 to 6 PA-8200 Processors 2048x2048 KB Cache per CPU 200 MHz 240 MHz D370 D380

HP-PB Bus Converter 20 MB/S

2 HSC Expansion Slots

HSC Slot

Source: HP 9000 Enterprise Servers Configuration Guide 5/98

HSC Bus Converter 116 MB/S

K370/80 System Architecture

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

D370/80 System Architecture

ECC Memory 64 MB -- 3 GB

Memory Controller

144 bit Memory bus

766 MB/S Processor/Memory Bus

PA-8000 PA-RISC Processors

HSC Bus Converter 116 MB/S

Core I/O

S/E SCSI-2 802.3 LAN RS-232 Mouse/Keyboard Parallel

EISA Bus Converter 13 MB/S

7 EISA Expansion Slots

5 HSC Expansion Slots

D370 Turbo Slot

Source: HP 9000 Enterprise Servers Configuration Guide 5/98

1 or 2 PA-8000 Processors 160 MHz 180 MHz 512x512 1024x1024 KB Cache D370 D380

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

D370/80 System Architecture

ECC Memory 64 MB -- 3 GB

Memory Controller

144 bit Memory bus

766 MB/S Processor/Memory Bus

PA-8000 PA-RISC Processors

HSC Bus Converter 116 MB/S

Core I/O

S/E SCSI-2 802.3 LAN RS-232 Mouse/Keyboard Parallel

1 or 2 PA-8000 Processors 160 MHz 180 MHz 512x512 1024x1024 KB Cache D370 D380

EISA Bus Converter

7 EISA Expansion Slots

5 HSC Expansion Slots

D370 Turbo Slot

Source: HP 9000 Enterprise Servers Configuration Guide 5/98

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: System Bus Capabilities/Scaling

J2240 System Architecture

ECC Memory 128 MB -- 4 GB

Memory Controller

144 bit Memory bus

766 MB/S Processor/Memory Bus

PA-8200 PA-RISC Processors

GSC Bus Converter 116 MB/S

Core I/O

LV/D or F/W SCSI WS/E SCSI-2 802.3 LAN RS-232 Mouse/Keyboard Parallel

EISA Bus Converter 13 MB/S

1 EISA Expansion Slot

1 or 2 PA-8200 Processors 236 MHz 2 MB x 2 MB Cache

GSC Bus Converter 116 MB/S

PCI Bus Converter 128 MB/S

3 GSC Expansion Slots

PCI Bus Converter 313 MB/S

64-bit slots

32-bit slots

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing CPU Selection: SPEC_SFS_1 Ratings

Numbers in blue are estimated. Reported SPEC data as of 10/99.

Network I/Fs

SPEC SFS 1

Data Flow

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Secondary Areas of Concern

- **High Availability**

- CPU failover
- Disk failover
- LAN failover
- Power failover

- **Performance Tools**

- **Service and Diagnostics**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Sizing Areas of Primary Concern

• CPU subsystem

- SPEC_SFS_1 rating
- CPU and I/O bus capacities
- CPU growth path

• LAN subsystem

- Numbers and types of LAN connections supported
- I/O Bus capacity
- Path to other LAN technologies

• Disk subsystem

- I/O Bus and channel capacities
- JBOD versus RAID
- LVM, HFS, and JFS (VxFS)

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

PPT Slide

```
if [[ -f ${LOGFILE} ]]; then
mv ${LOGFILE} ${LOGFILE}.OLD
fi
exec 9>${LOGFILE}
print -u9 "${HOSTSTAMP}"
print -u9 " IOPS retrans timeout"
while true
do
TIMESTAMP=$(date '+%b %d %H:%M:%S' )
#
# Print out the values for "calls", "retrans", and "timeout" from
# the client RPC information
#
INFO=$( ${NFSSTAT} ${LOG_OPTIONS} | \
awk '{if (NR == 4) {printf("%s\t\t%s\t\t%s", $1, $3, $5)}}')
${NFSSTAT} ${RESET_OPTIONS} > /dev/null
print -u9 "${TIMESTAMP} = ${INFO}"
sleep ${INTERVAL}
done
exit 0
#####
#####
```

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

PPT Slide

```
#!/sbin/sh
###
## This script will log the number of NFS RPC calls
## made during the interval tracked by the value in
## ${1}. The default interval is set to 10 seconds
## if no value is passed in ${1}. The lowest possible
## interval is one second, which should be small enough.
###
INTERVAL=10
NFSSTAT=/usr/bin/nfsstat
LOG_OPTIONS="-rc"
RESET_OPTIONS="-z"
LOGFILE=/tmp/NFS_IOPS.log
HOSTNAME=$(hostname )
HOSTSTAMP="${HOSTNAME} ${INTERVAL} Seconds"
if [[ -n ${1} ]]; then
INTERVAL=${1}
fi
${NFSSTAT} ${RESET_OPTIONS} > /dev/null
###
## Save the old log file, just in case we really want
## the data that is already there.
###
```

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Measuring Client NFS Activity NFS IOPS (NFS I/Os per second = RPC calls per second)

- Rebooting or executing “nfsstat -z” as root will zero out the NFS counters used by “nfsstat”.
- Method 1: Monday morning zero the counters, then Friday evening total the counters with the “nfsstat -c”. Divide to obtain the average.
- Method 2: Monday morning zero the counters, then at set time intervals measure the client activity with “nfsstat -c”.
- Method 3: Determine a standard mix of user activities, like “load drawing”, and measure IOPS for each operation.

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Measuring Client NFS Activity NFS Activity with the nfsstat command

“nfsstat -s” gives accumulated server NFS RPCs and mix

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Measuring Client NFS Activity NFS Activity with the nfsstat command

“nfsstat -c” gives accumulated client NFS RPCs and mix

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Measuring Client NFS Activity GPM

Client NFS PV2 & PV3 Operations

Source GPM->Reports->Network Info->NFS By Operation

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Measuring Client NFS Activity GPM

Client NFS PV2 & PV3 Operations

Source: GPM->Reports->Network Info->NFS By Operation

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Measuring Client NFS Activity GPM

Global NFS Activity Screen

Source: GPM->Reports->Network Info->NFS Global Activity

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Benchmark Data The NFS IOP “Magic” Number, 4KB

- **SPEC_SFS_1 defines 1000 NFS IOPS as 1000 NFS RPC calls of the previously noted mixture.**
- **NFS read and writes are 8192 (8KB). All other operations are approximately 192 bytes.**
- **With the defined mixture, the average NFS I/O in the SPEC_SFS_1 benchmark is approximately 3.2KB.**
- **The 3.2KB is rounded to 4KB/NFS IOP to account for things like packet headers, variable length data, etc.**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Benchmark Data Typical NFS IOPS requirements**

**Typical numbers only. There is no substitute for actual measurements of client NFS IOPS.

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Benchmark Data

SPEC_SFS_2 Information

- **SPEC_SFS_2 differs from SPEC_SFS1 in several ways:**
 - Benchmark workload is based on a survey of over 1000 servers in different environments and is more realistic
 - Supports NFS protocol version 2 (PV2) and protocol version 3 (PV3) and separate reporting
 - SPECsfs97.v2/SPECsfs97.v3
 - Supports both UDP and TCP transport mechanisms
- **We do not yet know how this benchmark affects some of the things we will discuss. Be careful!**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

PPT Slide

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

PPT Slide

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Benchmark Data

SPEC_SFS_1 operation mixture

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Benchmark Data (continued)

-
- **The SPEC SFS benchmarks measure fileserver performance in terms of NFS I/O operations per second (IOPS) and response time.**
- **The server is subjected to larger and larger numbers of requests until the fileserver cannot keep up, causing throughput to decrease or response times to become unacceptable. (> 50 milliseconds)**
- **Reporting SPEC numbers entails also reporting hardware and software configuration.**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Fileserver Benchmark Data

- **NFS server performance is specified by the SPEC (System Performance Evaluation Committee) benchmark(s).**
- **The specific benchmark is System FileServer.**
 - Nhfsstone
 - LADDIS is a re-implementation of Nhfsstone
 - SPEC_SFS_1 / SPECsfs93
 - NFS PV2, UDP, specific operation mix
 - SPEC_SFS_2 / SPEC SFS97
 - NFS PV2/PV3, TCP or UDP, different operation mix
- **Details may be found at <http://WWW.SPEC.ORG>**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Design Approach

- **Determine client NFS IOP needs**
- **Size the NFS server resource(s)**
- **Size the network infrastructure**
- **Tune the clients and the server**
- **Measure performance**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Where is the Network Today?

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

The NFS Client Drives the Network

Performance

1980

Time

2000

1985

1990

1995

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

NFS Workgroup Environment

- **There is more to deploying an NFS fileserver than simply sizing (or plugging in) the server!**
- **Multiple clients using a central fileserver involves designing a virtual computer, which must be balanced.**
 - Fileserver throughput and response time
 - LAN bandwidth and latency
 - Client throughput

Disks

CPU Boards

Backplane

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Seminar Agenda

- **Design Issues**
- **NFS Fileserver Benchmark Data**
- **Measuring NFS Client Requirements**
- **NFS Fileserver Sizing**
 - CPU
 - LAN
 - Memory
 - Disk
- **NFS Fileserver Tuning**
- **NFS Client Tuning**
- **NFS PV3**

[Previous slide](#)

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)

Sizing and Tuning NFS Clients and Servers

Everything you wanted to know about using NFS, but were afraid to ask

**Rob Lucke Americas Technical Computing
Advanced Technology Center**

[Next slide](#)

[Back to first slide](#)

[View graphic version](#)