HP CIFS Server hp-ux 11i Performance Tuning

Eric Roseme
Technical Consultant
Systems Networking and Security Lab
Hewlett-Packard

HP WORLD 2003
Solutions and Technology Conference & Expo

HP CIFS Server

- NEW NAME

CIFS Server

HP CIFS Server Tuning

- HP CIFS Server is a user space file sharing application for the Windows client platform that is deployed on customer systems ranging from one-way workstations to 64-way Superdomes.



- HP CIFS Server can be optimized to run most efficiently for a particular usage on a particular hardware platform.
- This sessions identifies key ***known*** tuning areas for optimizing HP CIFS Server performance.

HP CIFS Server Tuning

- Goal: provide data to configure system for basic sizes and tasks

- Ultimate Goal: Tune for specific client operations: read/write sizes and speed, lookups, metadata.
- Windows client behavior still very much a mystery!!

HP CIFS Server Tuning

- Existing Customer Production Platform Examples
  - 1-way A-Class: 650 connected clients 
  - 3-way L-Class: 850 connected clients 
  - 8-way rp7410: 4000 connected clients 

Agenda

- HP CIFS Server Tuning Strategy
  - SMB Architecture
  - Samba smb.conf Tuning
  - HP-UX for HP CIFS Server Tuning
  - Microsoft Client Effects
  - Connection Establishment
  - Summary

Tuning Strategy

- When to tune?
- What to tune?

Tuning Strategy: When to Tune?

- Observed behavior
  - Something has changed
- Basis for comparison
  - Expectations
  - Windows file server
  - Dedicated file server
- No performance spec (like SPEC SFS)
  - Common metric is unavailable

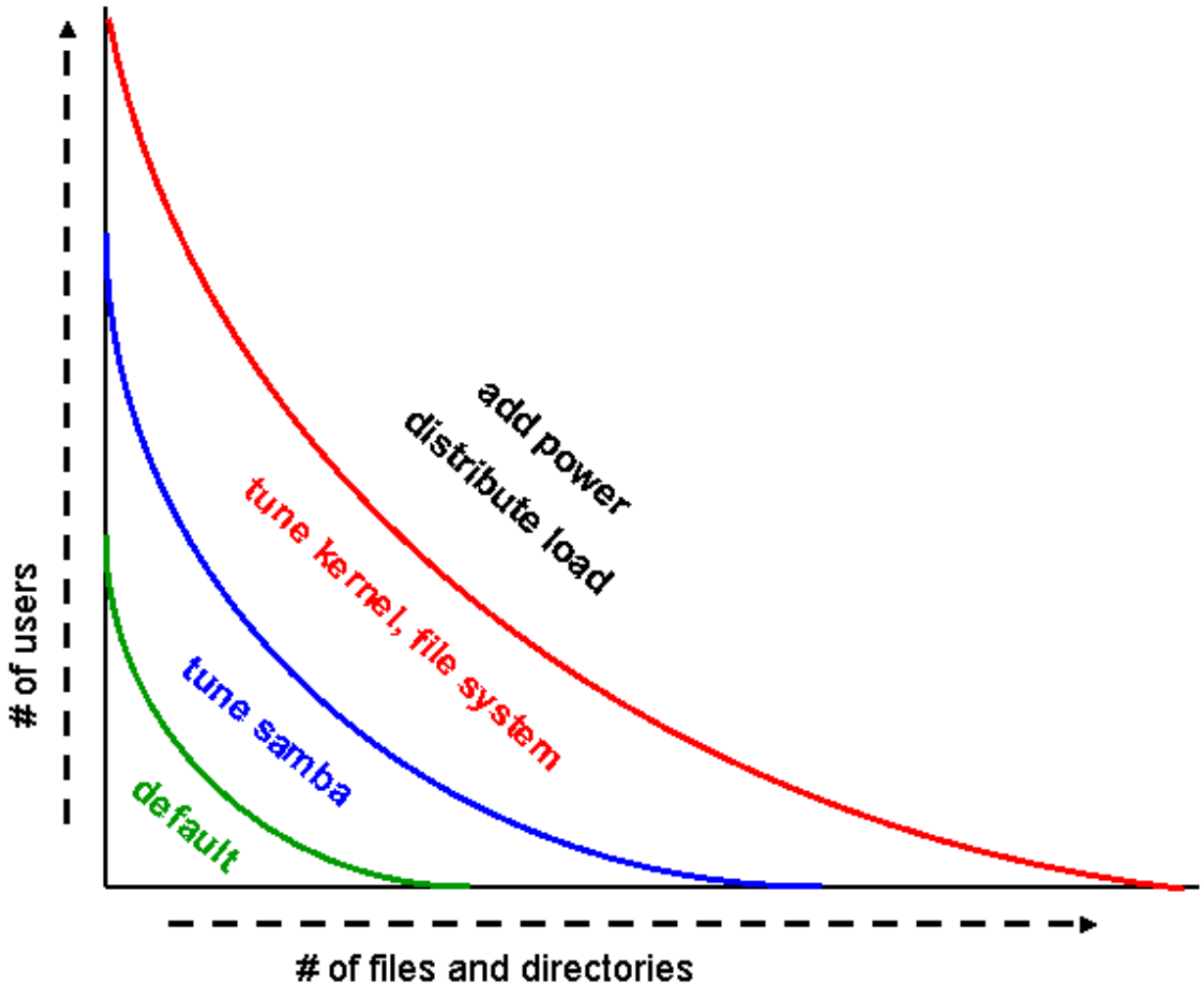
Tuning Strategy: When to Tune?

- Tuning objective
  - Maximum number of users
  - Maximum number of files
  - Big files
  - Application usage
  - Short sessions (connects/disconnects)
  - Read only
  - Client functions
  - Secure transfers
  - Multi-purpose co-existence

Tuning Strategy: When to tune?

- Most common tuning exercises
  - Maximize ***users*** (usually thousands)
  - Maximize ***data -Thousands of files*** (usually CAD)

Tuning Strategy: When to tune



Tuning Strategy: When to tune?

- HP CIFS Tuning Prerequisites
  - Evaluate local system performance
  - Disk bottleneck
  - Network bottleneck
  - Memory bottleneck
  - Other applications

Tuning Strategy: What to tune

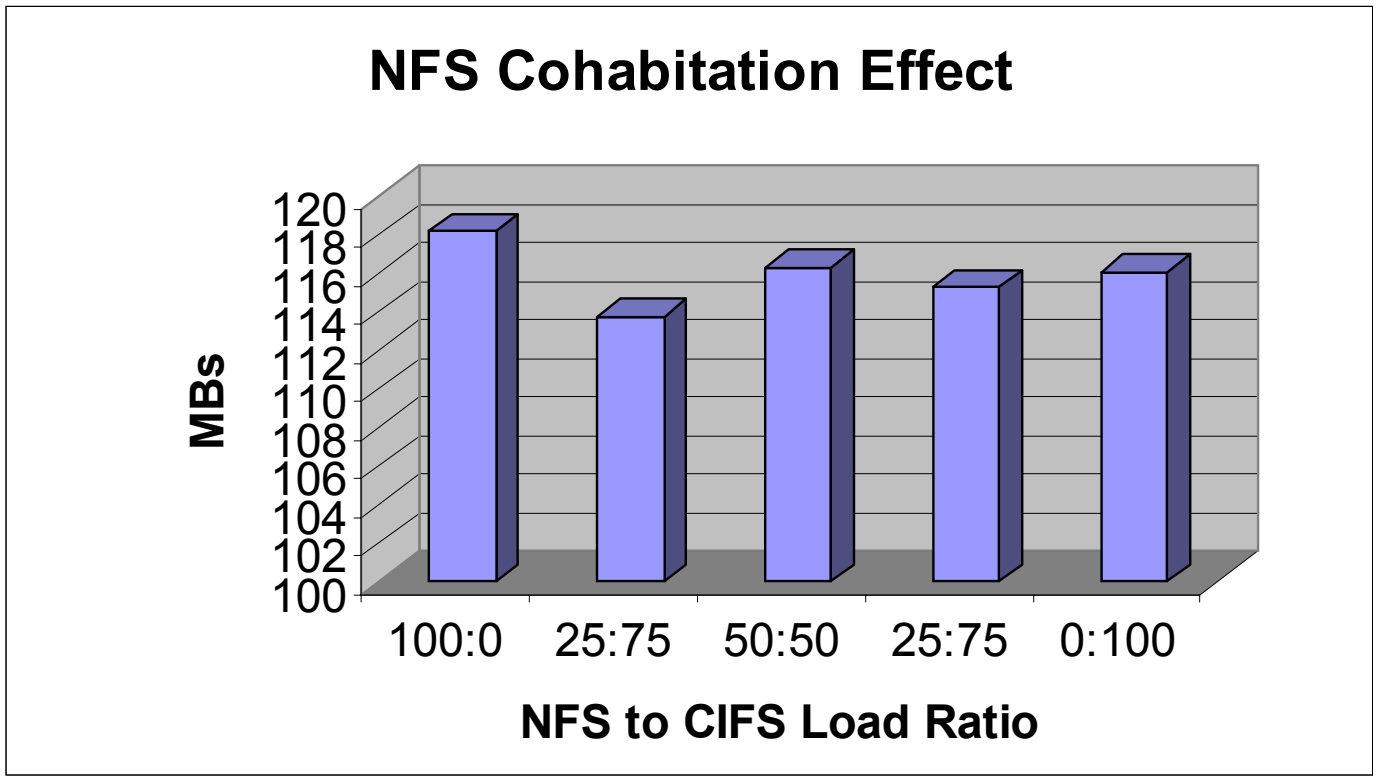
- Looks like our presentation agenda:
  - Samba (smb.conf)
    - locking
    - file system tuning
    - tuning parms
    - miscellaneous
  - hp-ux Server
    - kernel
    - file system
    - OS level & patches
    - hardware
  - Client
    - features
    - applications
    - user behavior
  - Connection Establishment
    - NetBIOS
    - WINS
    - NIS
    - DNS
    - Files

Tuning Strategy: NFS!

- CIFS deployments on existing NFS servers
- User space vs kernel
- tuning collisions
- NFS: common scenarios
  - sharing NFS-mounted file systems
    - Samba says 30% performance degradation
  - carefully configure automounter or AutoFS
- NFS best practices:
  - buy “optimizing nfs performance”, by Dave Olker
  - visit the HP Press booth

Tuning Strategy: NFS!

- NFS Server and CIFS Server Mix Throughput



Tuning Strategy: Effectiveness

- Samba executes in user space
- Defaults are very effective
- Enhancing for specific purposes
  - Not as easy
  - Extensive testing
  - Diminishing returns
- Samba is very configurable
  - create repeatable, measurable test
  - test one thing at a time
  - run each test multiple times: establish baseline

Large System Thumbnail

- 5000 user threshold
  - smbd overhead eats into throughput
- “connected” vs “concurrent”
  - concurrent: active processing
  - connected: idle processes
- Thousands of users
  - how many are active?
- MP Scaling works very well
- GbE Scaling works very well
- Startup memory: 1.5MB per client

Agenda

- CIFS/9000 Server Tuning Strategy
- **SMB Architecture**
- Samba smb.conf Tuning
- HP-UX for HP CIFS Server Tuning
- Microsoft Client Effects
- Connection Establishment
- Summary

SMB Architecture

- SMB and CIFS
  - CIFS is common name for the Specification
    - "Draft" presented to IETF by Microsoft in 1996
    - Not completed by Microsoft
    - SNIA multi-corporation group completed document in 2003
      - Thank You: IBM, EMC, HP, Microsoft, Network Appliance, Thursby, Veritas, Univ. of Minnesota
    - CIFS "Technical Reference" – not an official specification
  - SMB is common name for the protocol
    - Actually initiated by IBM in 1985
    - Adopted by Microsoft around 1987
    - Early definition published by XOpen
- CIFS Technical Reference
  - http://www.snia.org/tech_activities/CIFS

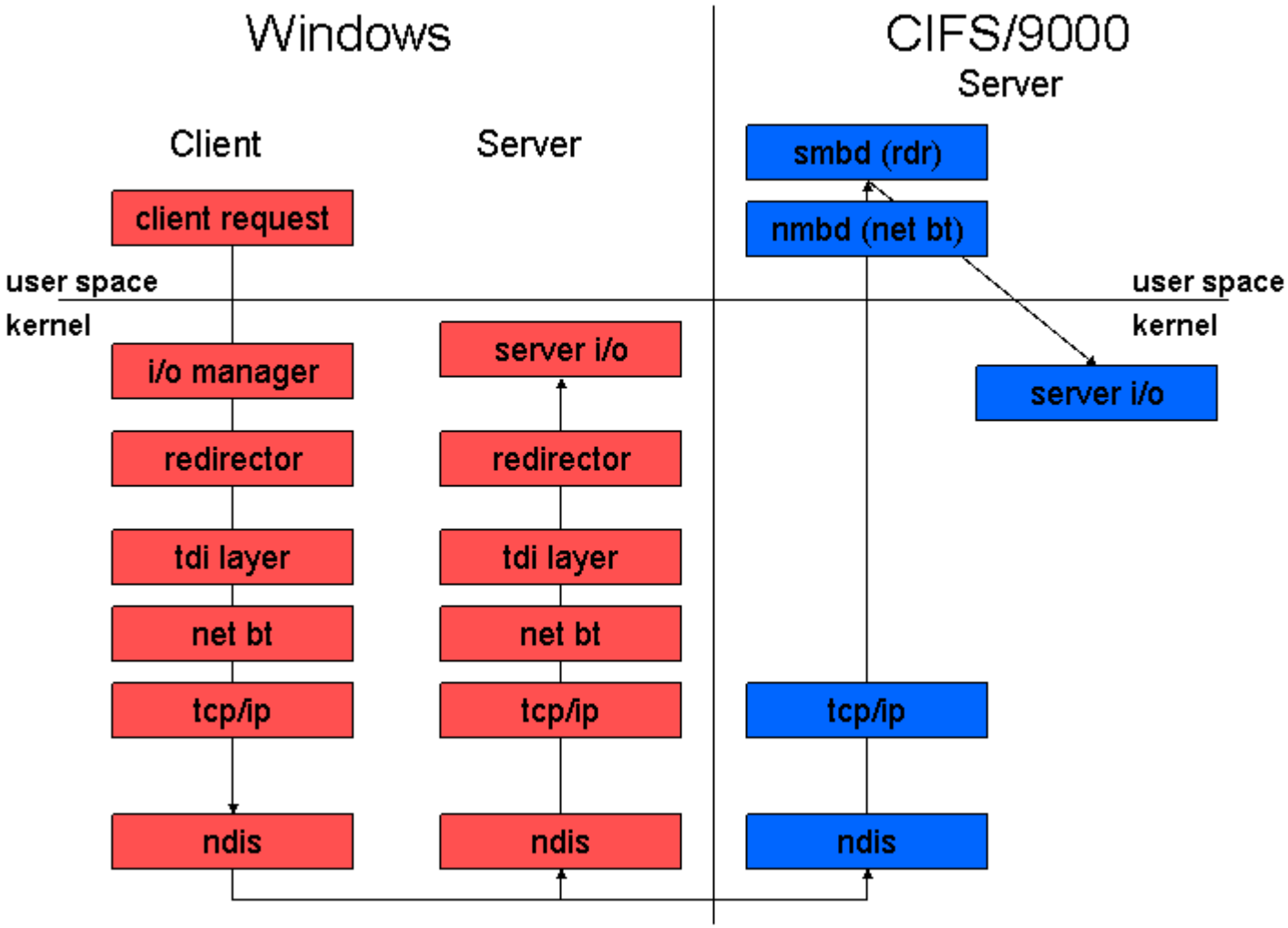
SMB Architecture

- SMB on Windows
  - Native distributed file system
  - Integrated in kernel and redirector
  - On client ***and*** server
- NTFS is native file system
  - SMB integrated in NTFS
  - File system handles many SMB operations

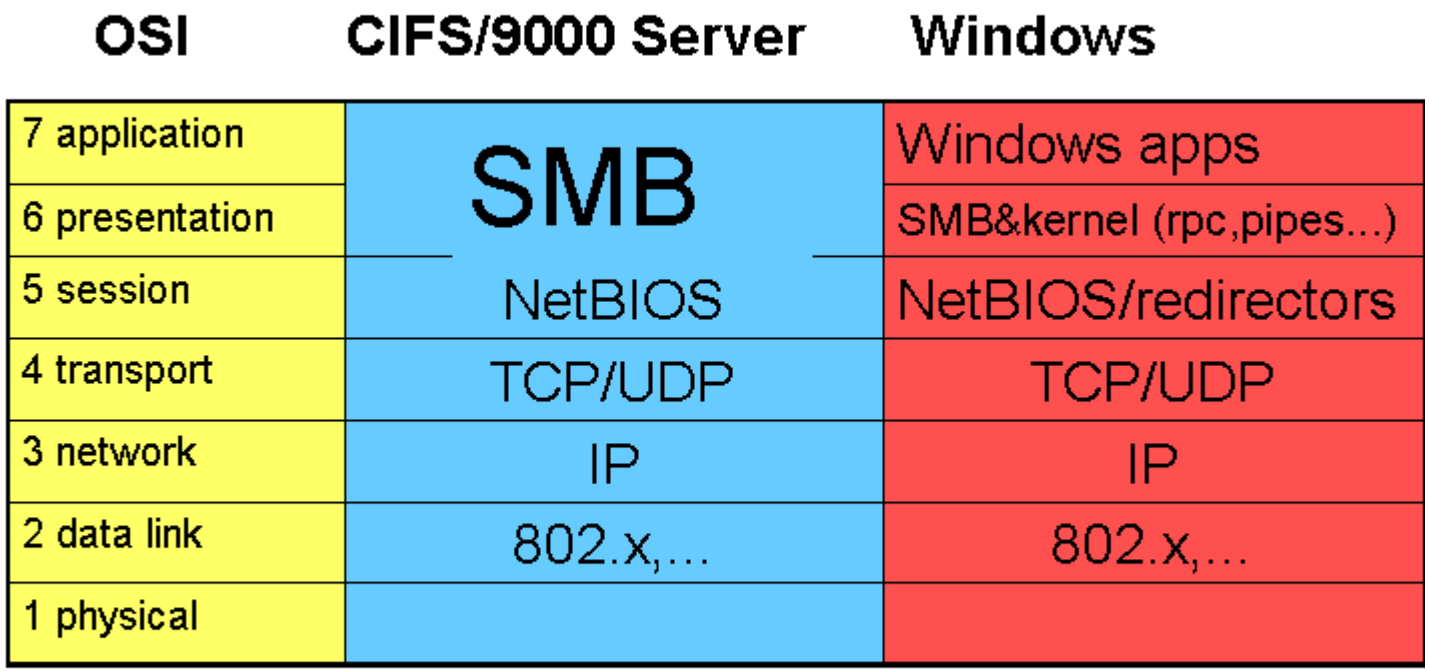
SMB Architecture

- SMB on hp-ux
  - Samba
  - User space application
  - Maybe more accurate implementation of “Spec”
  - Beware of the term “native” (distributed) file system
    - versus “emulation”
- JFS is native hp-ux file system
  - POSIX is integrated on JFS

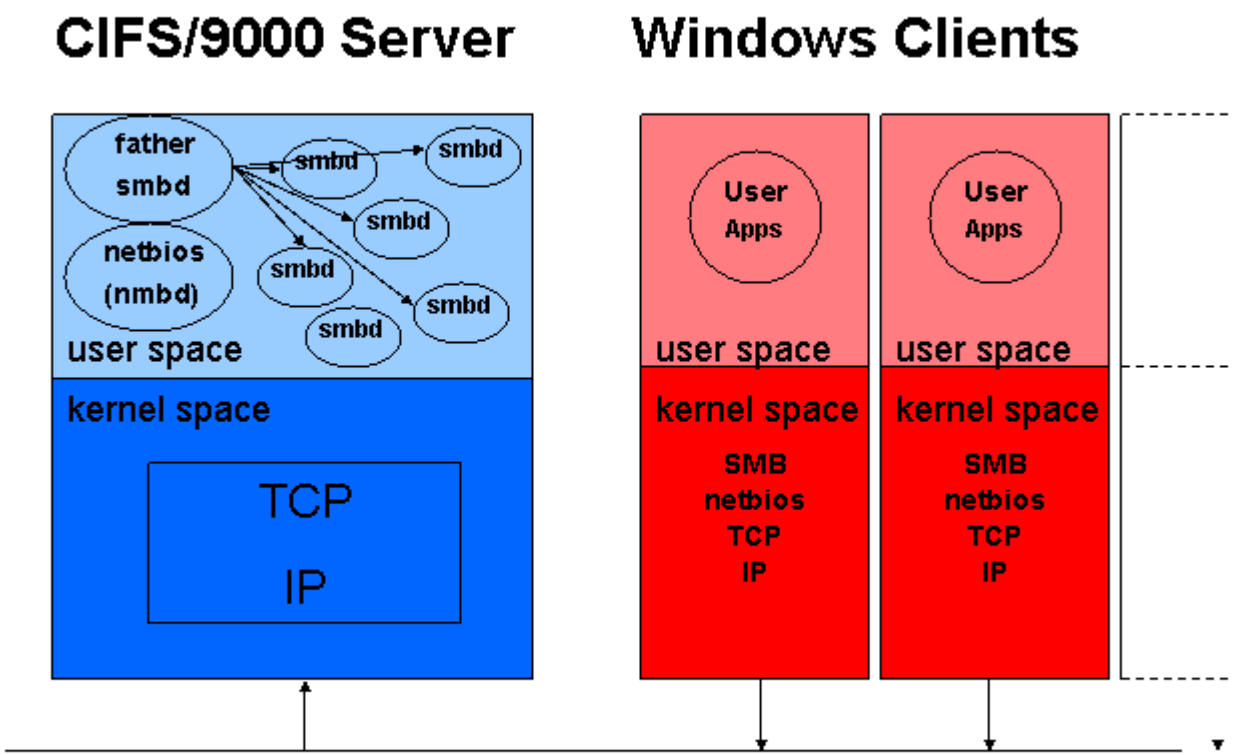
SMB Architecture



SMB Architecture



SMB Architecture



SMB Architecture

- Father-Child process generation
- (client-connects + 1) = total processes
  - 1000 connect clients = 1001 hp-ux processes
- Each process owns its I/O stack
  - Active TCP-IP connection per process
  - Even for idle clients
- Performance effect
  - System load for smbd I/O
  - System load to maintain pool of processes

Agenda

- CIFS/9000 Server Tuning Strategy
- SMB Architecture
- **Samba smb.conf Tuning**
- HP-UX for HP CIFS Server Tuning
- Microsoft Client Effects
- Windows Infrastructure Effects
- Summary

smb.conf Tuning

- Samba user space application tuning
- CIFS/9000 Server – Samba Version
  - 2.0.9: A.01.07 (Obsolete)
  - 2.2.8a: A.01.10 (2.2.g)
  - 3.0: TBD
- Best case: run pilot before upgrade!
- smb.conf parameter file tuning topics
  1. File Locking
  2. File System
  3. Tuning options (mainly transmission sizes)
  4. Miscellaneous Samba Features

smb.conf: File Locking

- oplocks
  - locking feature anomaly
  - not really locking
  - client-side caching
- oplocks=yes
  - useful for reducing network latency effect
  - minimal effect on a fast network
  - oplock unreliability often offsets perceived improvement
  - Not effective for MS Office applications
  - For writes, just delays the inevitable write
  - Minimal or no positive effect for reads

smb.conf: File Locking

- Oplock Considerations
  - exclusive shares (yes)
  - multi-user shares (no)
  - read-only shares (no)
  - NFS and UNIX access (no)
  - MS Office applications (no)
  - slow and/or unreliable networks (maybe)
  - multi-user databases (no)
  - PDM served shares (no)
  - “force user = username” (no)
  - high availability (no)

smb.conf: File Locking

- oplocks – recommended default
  - “oplocks = no”
- Performance degradation vs improvement
- see whitepaper
  - “Oplock Usage Recommendations”

smb.conf: File Locking

- strict locking
  - default is no
  - only check locks when client app makes a locking call

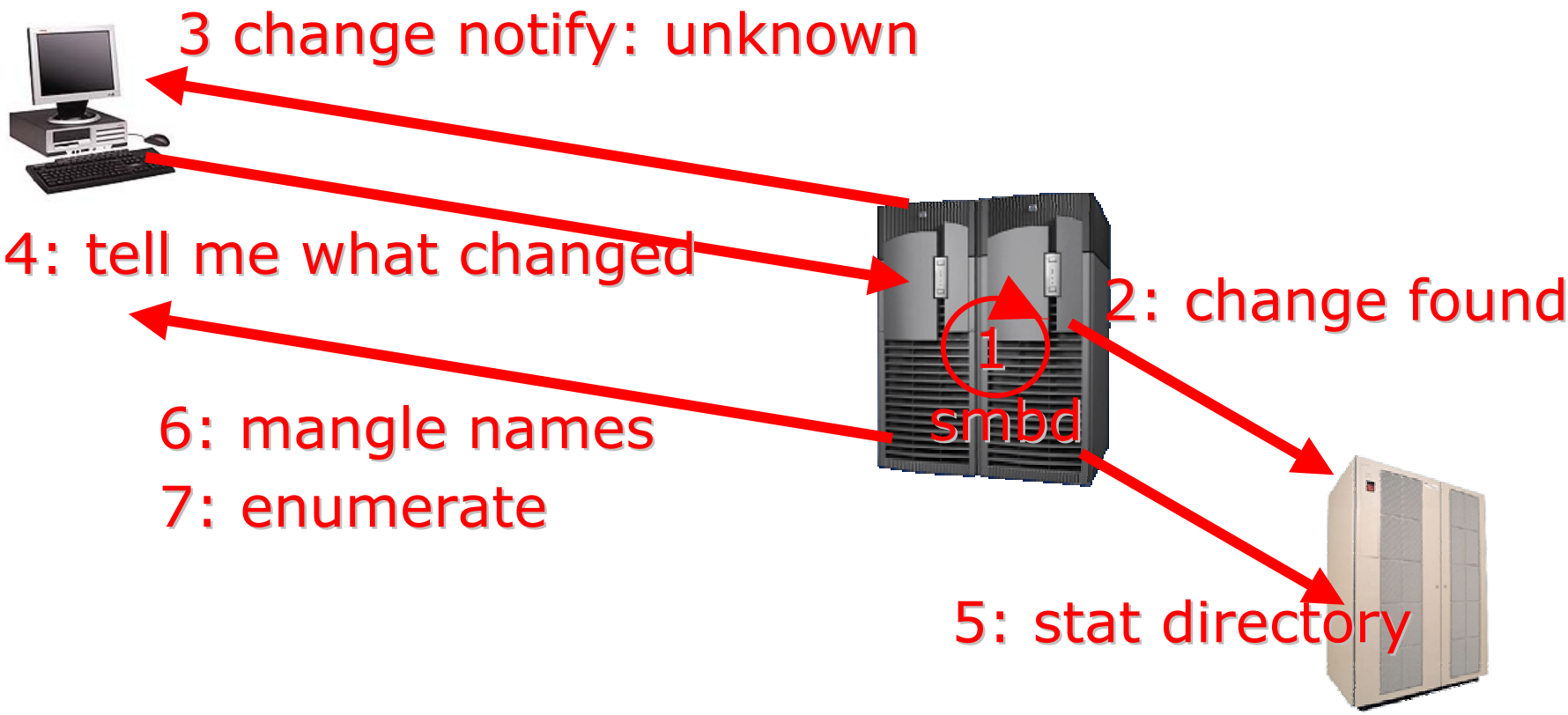
- strict locking = yes
  - smbd checks for file locks on every read and write
  - will slow response time and throughput (on share)

smb.conf: File System

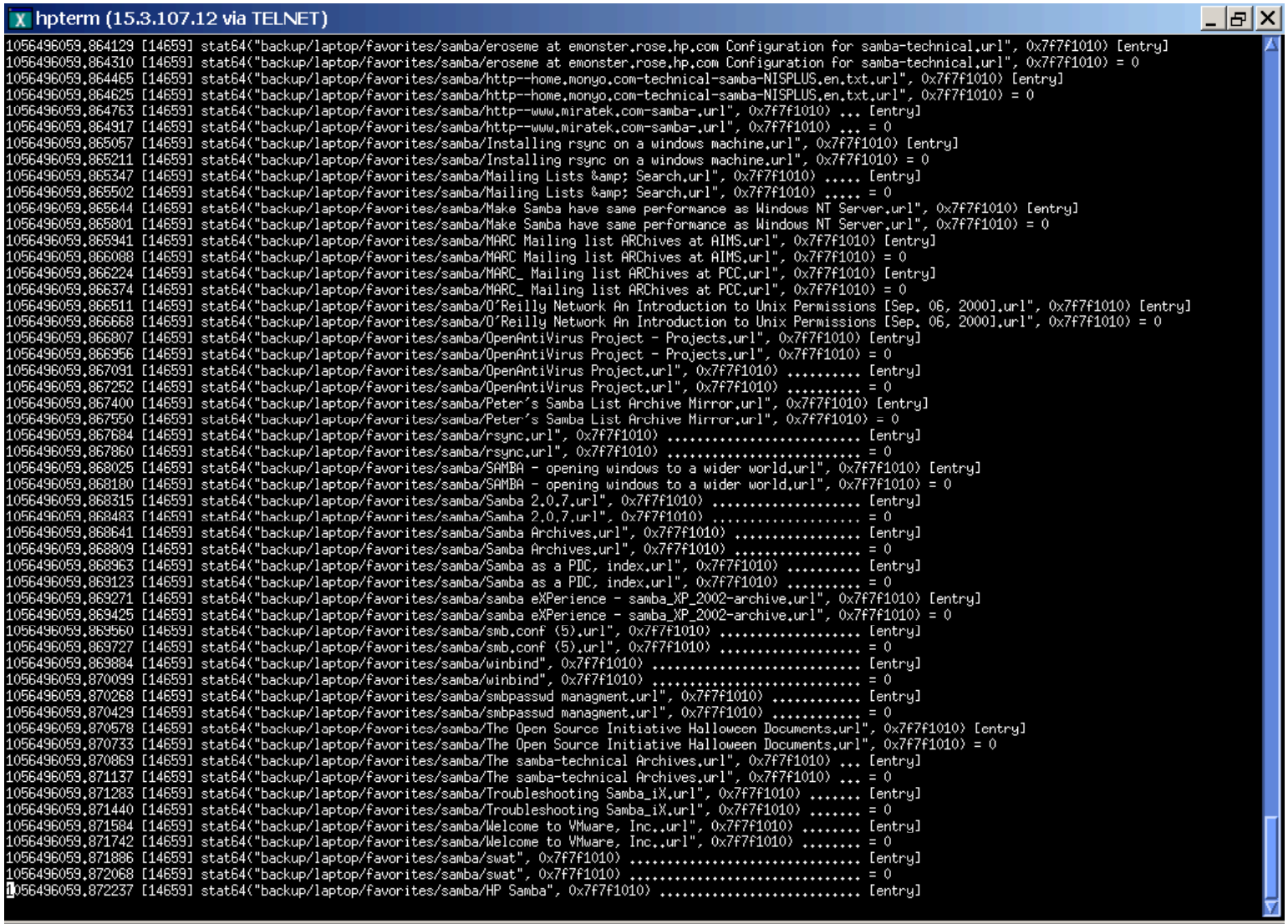
- change notify timeout
  - CIFS Spec SMB “NT_TRANSACT_NOTIFY_CHANGE”
  - On Windows real-time directory change notification
  - On Samba periodic polling directory for changes
- Default change notify timeout = 60 (seconds)
  - every 60 seconds the share (directory) is enumerated
  - stats the share directory structure
  - very expensive for large or numerous directories
- Set default to 600 seconds
  - Users can manually refresh their client explorer windows

smb.conf: File System

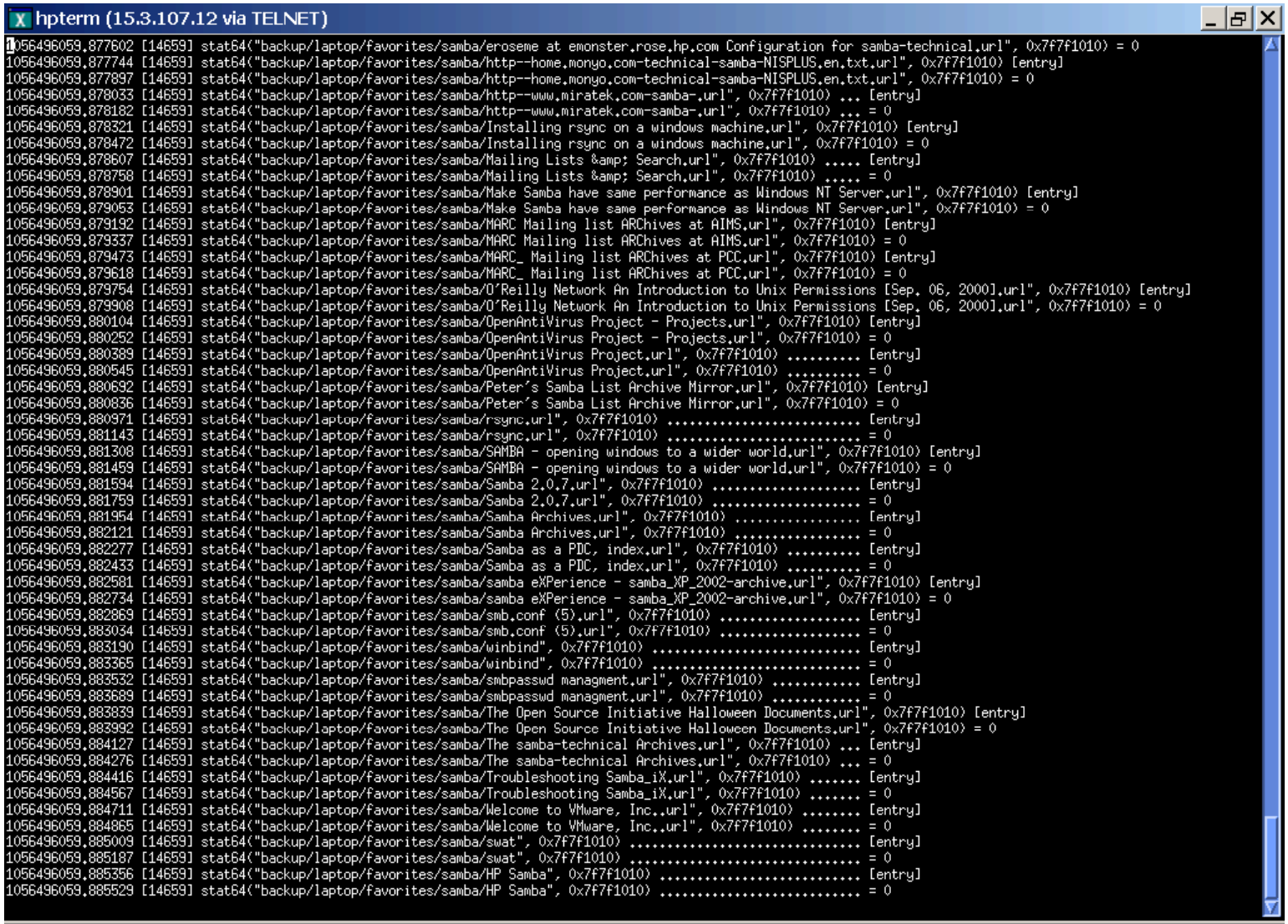
- change notify timeout



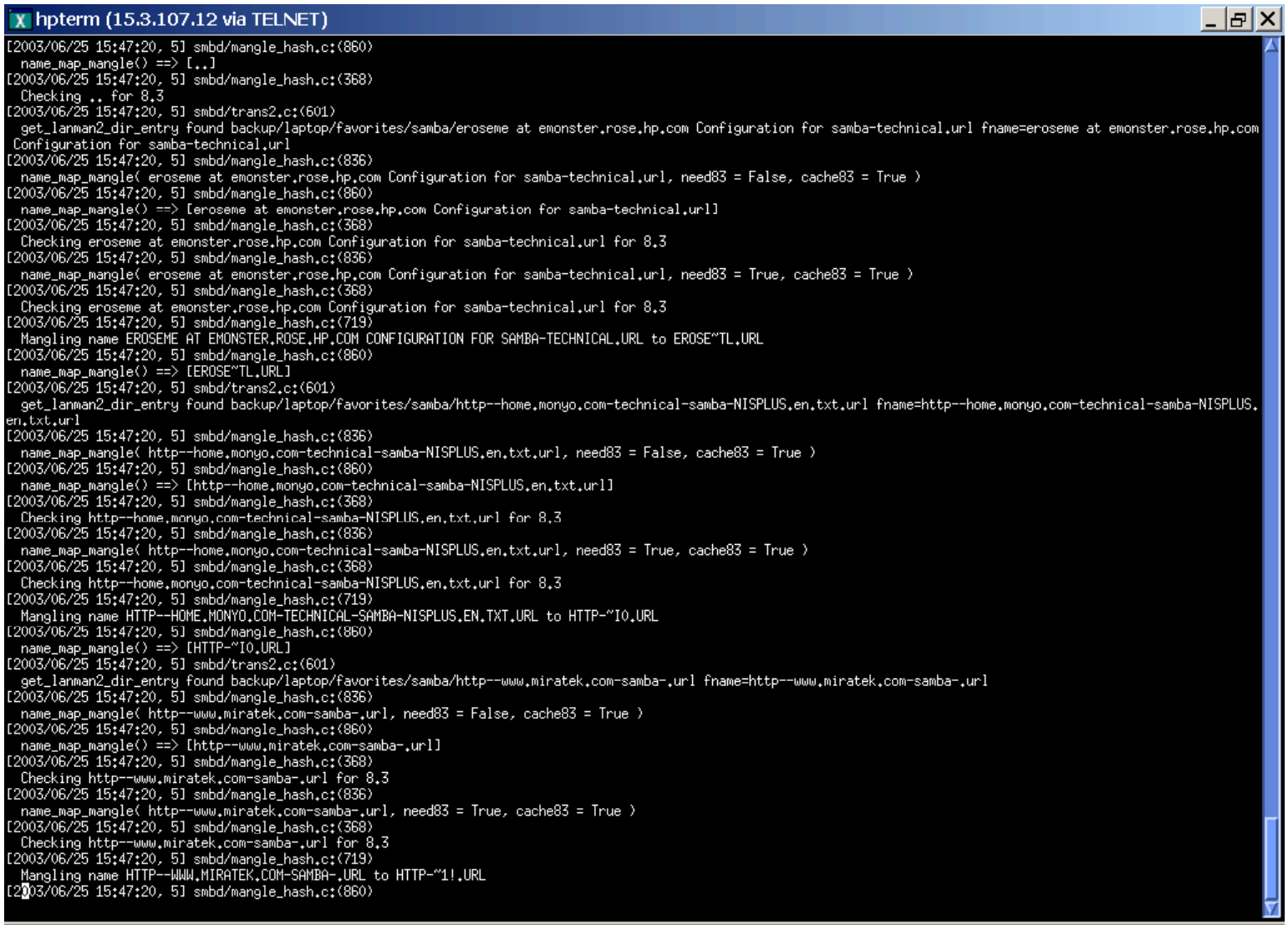
smb.conf: File System



smb.conf: File System



smb.conf: File System



smb.conf: File System

- synchronous writes
  - Samba defaults to asynchronous writes
    - smbd writes to memory buffer, then returns to processing. Buffer is flushed to disk later.
    - Most efficient behavior
- Windows CreateFile API
  - FILE_FLAG_WRITE_THROUGH flag
  - requests synchronous writes
    - smbd writes to memory buffer, blocks until buffer contents are written to disk
    - Poor performance, better data integrity

smb.conf: File System

- strict sync = yes (default = no)
  - Samba honors FILE_FLAG_WRITE_THROUGH
  - synchronous writes when called by CreateFile API
- sync always = yes (default = no)
  - Samba executes all writes synchronously
  - Requires that “strict sync = yes”
- Any synchronous write will slow throughput

smb.conf: File System

- "strict sync" and "sync always"

strict sync	sync always	file_write_through	synchronous writes
no	no	yes	**no**
yes	no	no	**no**
yes	no	yes	**Yes (slow)**
no	yes	yes	**no**
yes	yes	no	**Yes (very slow)**

smb.conf: File System

- strict allocate = yes (default = no)
  - disables sparse files
  - sparse files optimize storage for blank data blocks
  - hp-ux enables sparse files by default
  - Windows allocates storage space to blank data blocks
  - “strict allocate = yes” enables Windows compatibility
- dont descend = yes (default = no)
  - dont descend (sic) suppresses share object display
  - share subdirectories will appear to be blank
  - directory is enumerated, but not displayed
  - appears to provide performance benefit, but does not
    - but does prevent sub-directory usage

smb.conf: File System

- wide links = yes (default)
  - enables smbd to follow links outside the shared directory
  - smbd does a simple lookup on each link
  - HA failover: is link target on shared volume?
  - requires “follow symlinks = yes”

- wide links = no
  - prevents smbd from following links outside shared directory
  - smbd must lookup the link to see where it goes
  - extra lookups make additional system calls – stat
    - testing shows 3 to 12 times number of stats
  - can slow throughput **30%**

smb.conf: File System

- wide links: with existing wide link in an exported share

follow symlinks	wide symlinks	Performance effect
yes	yes	negligible
yes	no	Yes: up to 30%
no	no	Yes: up to 30%
no	yes	negligible

- Note: with no existing wide link in share
  - "wide links = yes" has no performance effect
  - "wide links = no" does degrade performance
    - but not as much as with an existing link

smb.conf: Tuning Parameters

- Samba network-specific tuning parameters
- Primarily adjust transmission sizes and timing
- Samba and CIFS/9000 defaults
  - Typical usage (Office apps)
  - System load: 20 to 4000 users observed
  - Tested up to 5000 users
- Non-defaults
  - May be effective for custom applications
  - Some are for non-hp-ux
    - not applicable

smb.conf: Tuning Parameters

- socket options
  - options are protocol specific – TCP
  - setsockopt configures hp-ux TCP socket
  - hp-ux supports a subset of listed Samba options

- TCP_NODELAY
  - default is yes
  - turns of Nagle algorithm
  - tcp_naglim_def=1 (default is 65535)
  - sends small packets immediately
  - Try changing only on
    - VERY WIDE LAN
    - Very large writes (> 65535)

smb.conf: Tuning Parameters (socket options)

- IPTOS_LOWDELAY
  - all over Samba docs – not on hp-ux
- SO_KEEPALIVE
  - disable TCP connection timeout. Not needed.
- SO_SNDBUF (TCP receive Window)
  - hp-ux default = 32768
  - Custom data streaming as high as 262144
- SO_RCVBUF (TCP receive window)
  - hp-ux default = 32768
  - Custom data streaming as high as 262144

smb.conf: Tuning Parameters

- SMB servers support multiple Windows client versions
- Each Windows client version supports subset of read/write SMBs
- SMB servers must support all SMB read/write behavior
  - including curious protocol cases (notify change)
- Max buffer size negotiated at session setup
  - Session_Setup_AndX and Negprot response
- Thus explaining multitude of transmission sizing parameters

smb.conf: Tuning Parameters

- max xmit = 16644
  - new default, best config for Windows NT, 2000, XP clients
- large readwrite = yes
  - new default, best config for Windows NT,2000, XP clients
- read raw = yes (pre-NT40)
  - support 65535 reads
  - obsolete clients – no need to change it
- write raw = yes (pre-NT40)
  - support 65535 write
  - obsolete clients – no need to change
- read size = 16384
  - threshold for network/disk readahead writebehind

smb.conf: Tuning Parameters

- **block size = 1024**
  - **How smbd reports disk blocks to client**
- **write cache size = 0**
  - **only valid for “oplocks = yes”**
  - **threshold for smbd memory cache for writes**
  - **optimize disk access**
- **getwd cache** = yes
  - deep directory path caching
  - speeds lookups
  - (not limited to “wide links = no”)

smb.conf: (Misc) smb.conf size

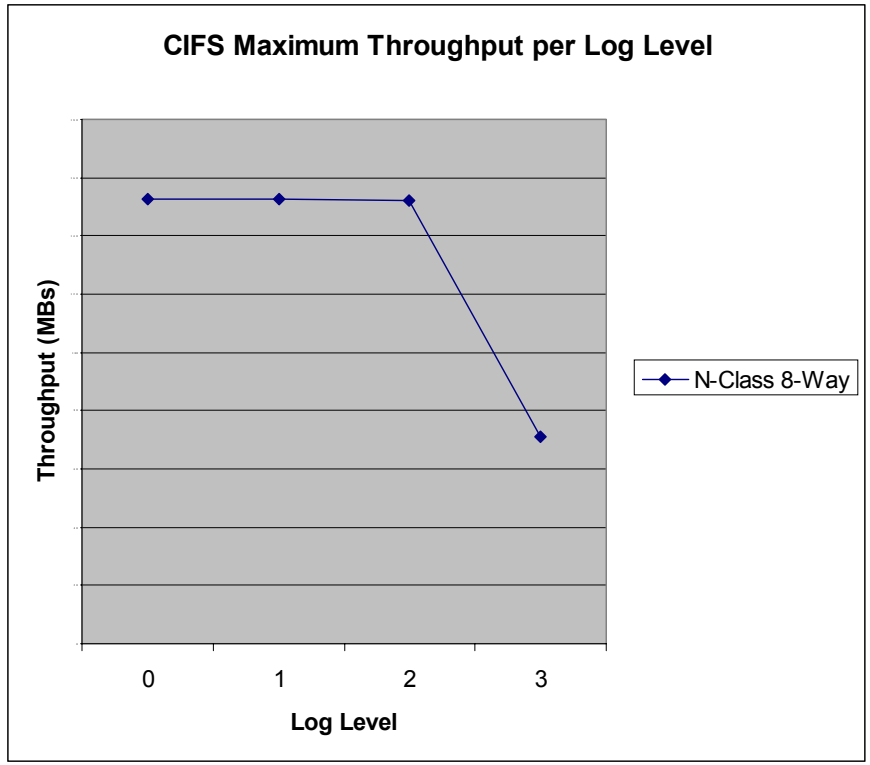
- smb.conf file size
  - smb.conf size affects memory footprint for every smbd
  - example: smb.conf of 90000 lines, 1.5MB
    - smbd memory footprint of 11.2MB (every smbd)
  - include =
    - user-specific smb.conf files
    - include = /etc/opt/samba/smb.conf.%m
  - %m,%U
    - user name or NetBIOS name defines home directories
    - [%U]
      path = /data/%U
  - [homes] share

- Use substitution variables to minimize smb.conf size

smb.conf: (Misc) Log Level

- log level
  - "log level" value 0 to 10
  - 0-2 no effect on performance
  - 3-10 progressively degrades

- Actual cases of sites
  - running production
  - with "log level = 3"
  - 2000 users
  - at maximum system load



smb.conf: (Misc) Log Level

- Log Level
  - performance degradation due to number of writes
  - Manual Explorer refresh of share

Level	Writes
0	1692
3	3267
6	7808
10	11250

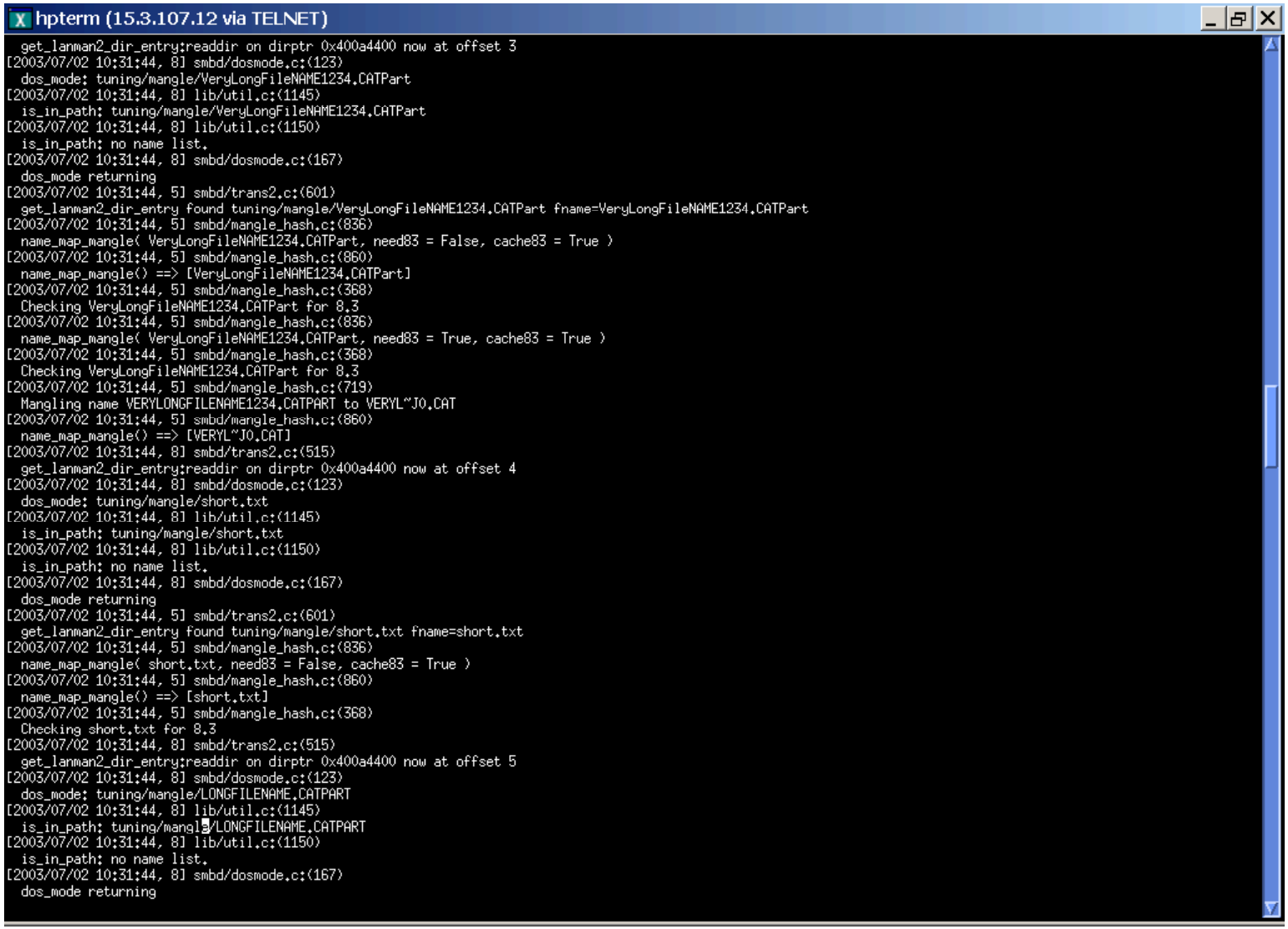
smb.conf: (Misc) Mangling

- Name Mangling
  - Samba feature for 8.3 file naming translation
    - “down level” clients: DOS, W3.51
  - Windows mangles names too (in file system)
  - longfilename.txt = lo~name.txt
- Name Mangling has little/no effect for average use
- Big directories see a slowdown
  - as number of files increases
  - as file names get longer

smb.conf: (Misc) Mangling

- Name Mangling processing load
  - not via system calls
  - not vi smb wire traffic
  - load is computational in smbd
  - caches mangled names in process memory
- Example
  - Share of 6000 file and 500 directories
  - name mangle = **no**
  - search for VeryLongFileName1234.PARTname
    - ***zero*** mangling log entries
  - name mangle = **yes**
    - ***55625*** mangling log entries

smb.conf: (Misc) Mangling



smb.conf: (Misc) *case* sensitivity

- ***case*** sensitivity needs a separate 2-hour presentation
- hp-ux (UNIX) is (case sensitive, case preserving)
- Windows is (case **in**sensitive, case preserving)
- Samba ***case*** configuration options give excellent results
  - but can cost processing cycles
- Default: case sensitive = no
- case defaults have no effect for average usage
- for very large directories
  - **“case sensitive = yes”** can help performance
  - decreases stat calls by about 15%
  - application-Windows client testing required!

smb.conf: (Misc) interfaces

- Samba defaults to bind first discovered LAN card
  - usually Core I/O 10/100BT
  - A system can have 4 GbE NICs, but only using Core I/O
- 10 concurrent users can bottleneck a single GbE card
- multiple LAN cards Scale (should be gigabit ethernet)
  - see later topic
- Configure Samba to recognize LAN all cards
  - interfaces = 192.1.1.1/255.255.255.0 192.1.1.2/255.255.255.0
  - bind interfaces only = yes
- Additional NIC configuration in later topic

Agenda

- CIFS/9000 Server Tuning Strategy
- SMB Architecture
- Samba smb.conf Tuning
- **HP-UX for HP CIFS Server Tuning**
- Microsoft Client Effects
- Connection Establishment
- Summary

Kernel Parameters

- Kernel tuning REQUIRED!!!
- Default 11i kernel parameters
  - HP CIFS Server exhausts nflocks at ~ 20 connected clients
  - HP CIFS Server exhausts nfiles at ~ 30 connected clients
- Remember: 1 connected client = 1 system process (smbd)
- Thus, most CIFS/9000 installations require kernel rebuild

- Kernel Tuning
  - Usage to date: Maximize number or client connections!
  - Big servers, thousands of connected clients

Kernel Parameters

- Basic philosophy
  - calculate HP CIFS Server requirement
  - calculate all other system baseline requirement
  - merge and adjust
- Typical system example
  1. Use HP CIFS Server recommendations
  2. Use “optimizing nfs performance” baseline for NFS
  3. Adjust baseline for system usage
  4. Try it
     a. merge
     b. apply
     c. test
     d. measure
     e. evaluate
     f. adjust
     g. repeat

Kernel Parameters

- maxusers
  - Not the best choice for HP CIFS Server
  - Macro that adjusts other kernel parameters
    - Often quoted as user limit
  - HP CIFS Server requires specific values
- For Example
  - set maxusers=1000
  - incorrect for CIFS/9000
    - nflocks should be > 10000
    - nfile should be > 25000

parm	current	pending
nfile	910	14849
nflock	200	200

Kernel Parameters

- nflocks
  - (10*maximum smbd)+(other apps + system)
  - example 1000 connected clients and baseline NFS system
    - (10*1000)+(2048) = 12048
- nfile
  - ((23+opens_per_smbd)*maximum smbd)+(other apps+system))
  - example 1000 connected clients and baseline NFS system
    - ((23+7)*1000)+(8192)=38192
- nproc
  - (maximum smbd)+(other apps+system)
  - example 1000 connected clients and baseline NFS system
    - (1000)+(1024)=2024
- ninode
  - whoa! See next page!

Kernel Parameters

- ninode
  - Most tuning documents specify some ninode formula
  - ninode directly tunes the inode table – **ON HFS!**
  - ninode does not directly tune the inode table on JFS!
- JFS 3.3 is the recommended file system for HP CIFS Server!
- ninode tuning for HFS with HP CIFS Server
  - no unique CIFS requirements
  - tune for maximum system uniquely opened files

Kernel Parameters

- vx_ninode
  - used to tune JFS (VxFS) ninode table
  - default is 0
    - if 0 at boot, then system sizes VxFS ninode table
    - sizing is based upon amount of ***system physical memory***
    - if system VxFS ninode sizing is less than ninode
    - then ninode value is used
- Example:
  - Memory recommendation for 1000 users: about 3GB
  - system vx_ninode default is 128,000
  - probably needlessly large

Kernel Parameters

- vx_ninode
- to tune vx_ninode, JFS patch required
  - 11.0: PHKL_18543
  - 11.i: PHKL_24783
- For static tuning
  - tune ninode for /stand (/stand must be on HFS): 4000
  - tune vx_ninode based upon system usage
- See
  - HP-UX Performance Cookbook (Stephen Ciullo)
    - http://h21007.www2.hp.com/dspp/files/unprotected/devresource/Docs/TechPapers/UXPerfCookBook.pdf
  - optimizing nfs performance (Dave Olker)
    - HP Press booth

Buffer Cache

- HP CIFS Server not as sensitive to buffer cache as NFS
  - HP CIFS Server does not tune the client (like NFS)
  - Read performance not affected
  - SMB protocol itself does not read ahead very well, anyway
- CIFS/9000 buffer cache sizing primarily for large writes
  - Only 1 known case of buffer cache sizing empirical data
  - Increasing buffer cache from 500MB to 2GB
    - Produced 18% throughput improvement
    - No significant gains after 2GB
    - bufpages = 512000
- Each environment is unique
- Testing required for each environment!
  - Especially for co-hosting NFS

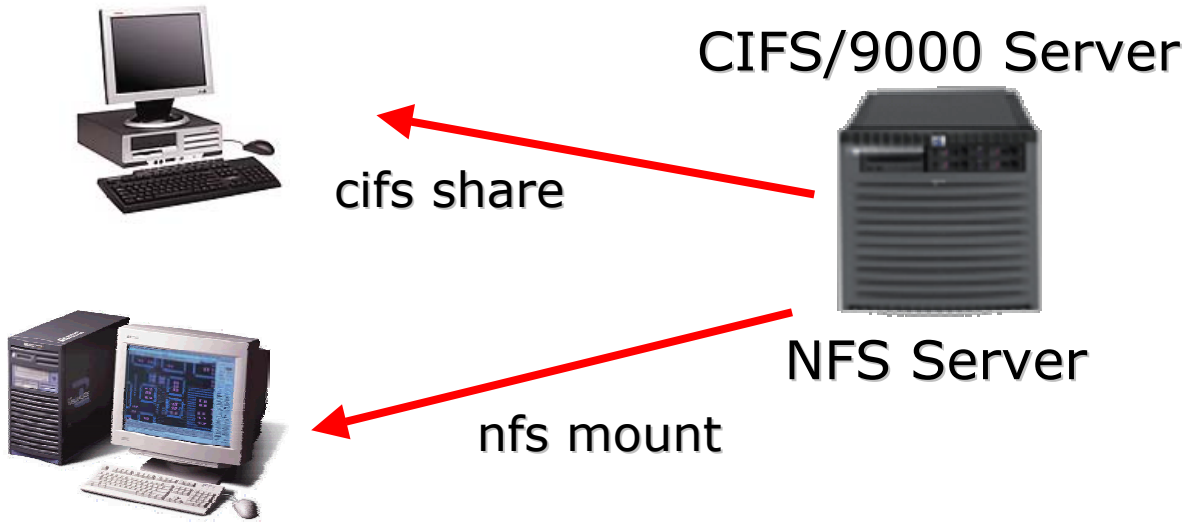
Buffer Cache

- CIFS/9000 Server and NFS Client

- Size buffer cache for NFS client
  - especially on hp-ux 11.0 (large cache on NFS client is bad)
  - CIFS/9000 Server only as fast as NFS client reads
    - Optimize NFS readahead
    - Note NFS client read optimization techniques (biod, buffer cache)
- Standard throughput degradation is **30%**

Buffer Cache

- CIFS/9000 Server and NFS Server



- Evaluate CIFS load versus NFS load
- Size buffer cache for NFS server
- NFS is likely more sensitive to buffer cache tuning

File System

- JFS vs HFS
- HFS default file system on hp-ux 11i
- JFS 3.3 recommended for CIFS/9000 Server
  - file layout 4
  - enables POSIX ACLs
- JFS 3.3 improvements
  - Journaling: file system check (vs chkdsk)
  - Maximums
  - Performance
  - ACLs
  - online resizing/defragmenting
  - other stuff

File System

- Large directories
- **Most common performance inhibitor**
- Cause: Applications that enumerate all files
- Samba/SMB+Windows+UNIX = stat64
- Threshold appears to be ~ 2000 files
- Symptom: Always drives CPU to 100%
- Long file names exacerbates condition

File System

- Large directories
- **Example**
  - system with directories of
    - (1) 31,700 files
    - (2) 13,000 files
    - (1) 12,000 files
    - (5) 8,000 files
  - application searches all directories for:
    - VeryLongFileName1234.PARTname
  - If file is not found, starts another search
- Excessive stat64 calls
- Excessive file name mangling

File System

- Large directories- what to do?
  - evaluate application efficiency
    - preferably before committing to the application
  - optimize file naming convention
    - shorten filenames as much as possible
  - flatten out directories

  - also
  - Test with
    - name mangle = no
    - case sensitive = yes

File System

- /dev
- Samba uses /dev for random number generation
  - At connection establishment
- stats every file in /dev for each new connection
- Thousands of files in /dev
  - degrades connection establishment
- Samba 2.2.5 enhancement
  - First connection caches /dev stats
  - subsequent connections access cache
- Upgrade to CIFS/9000 Server 2.2.g A.01.10a

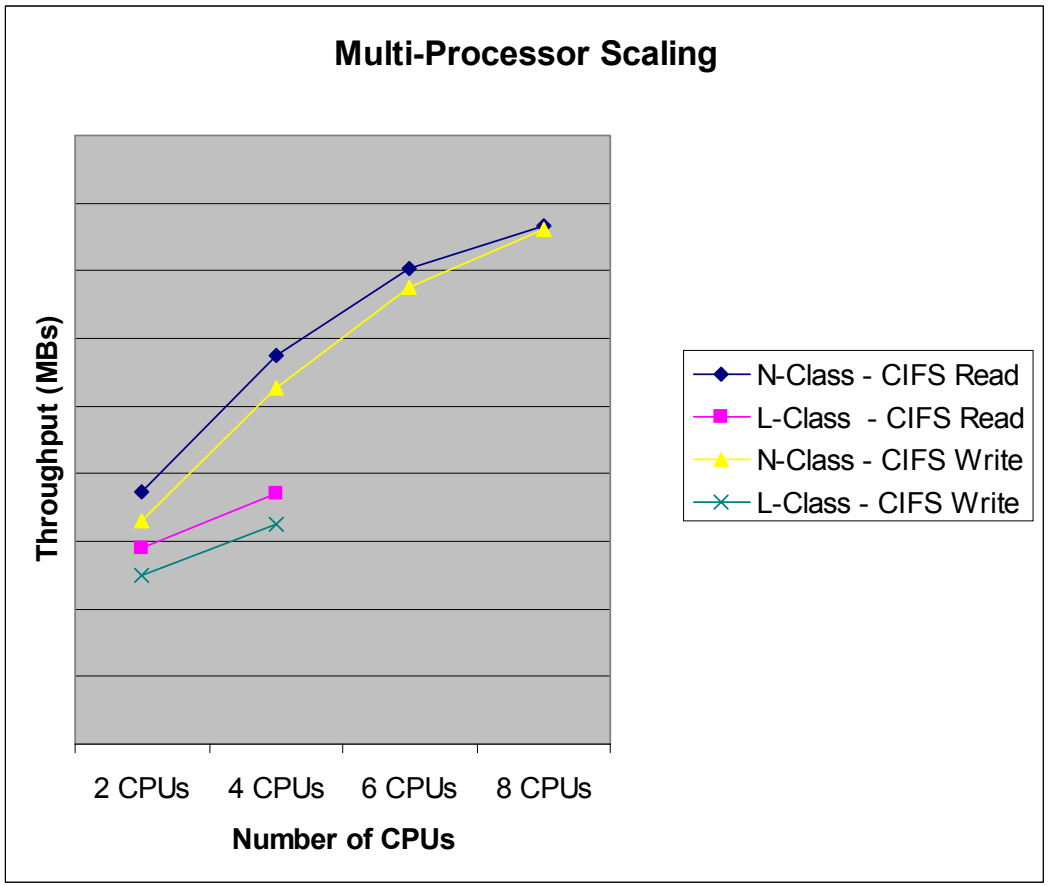
System Hardware

- Goal: Maximize CPU utilization
  - eliminate all other bottlenecks
- Goal: Maximize connected users
- Goal: Maximize system throughput (MBs)

- CPU Scaling
- NIC Scaling
- Memory Scaling
- Disk Access

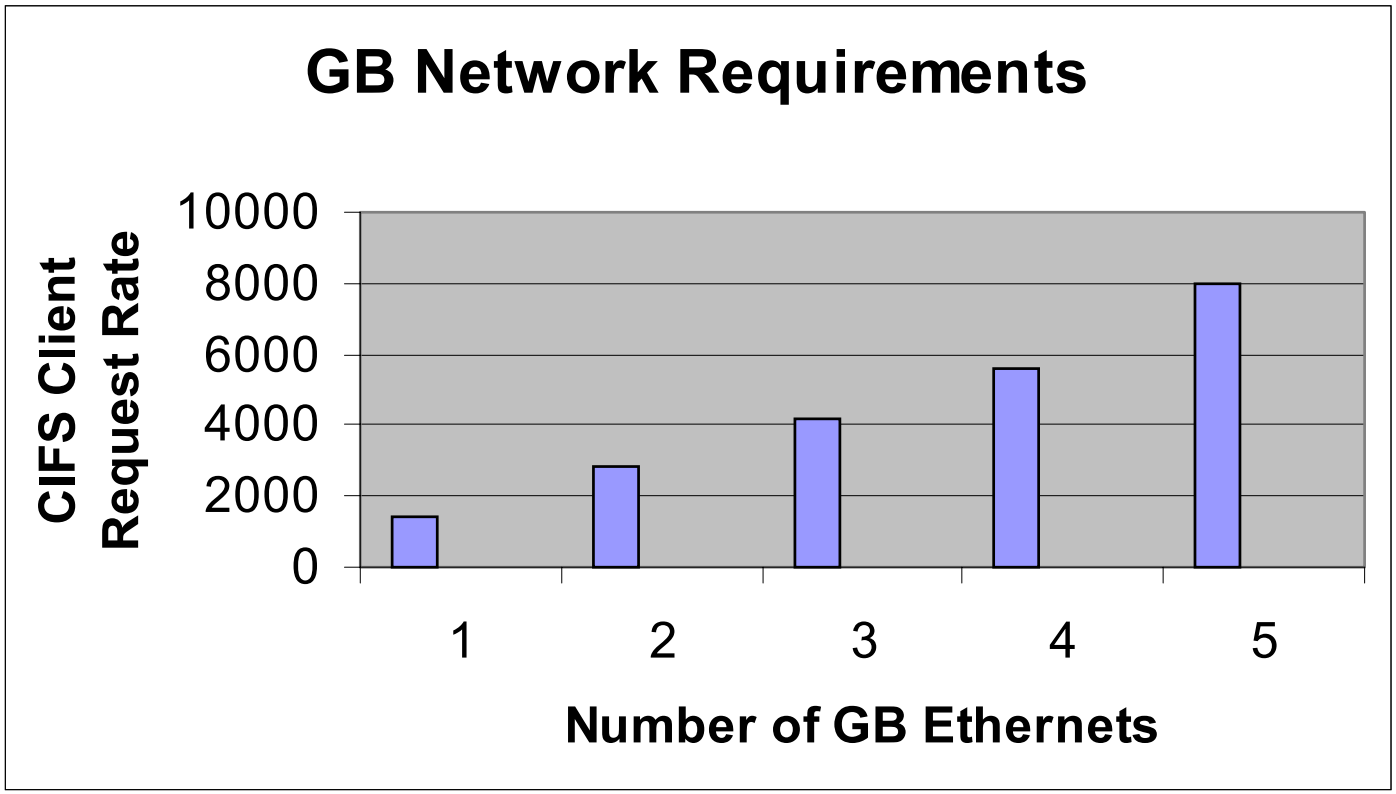
System Hardware

- CPU Scaling



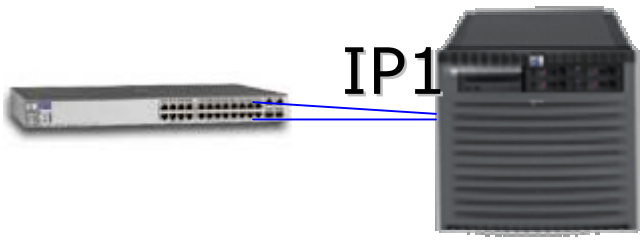
System Hardware

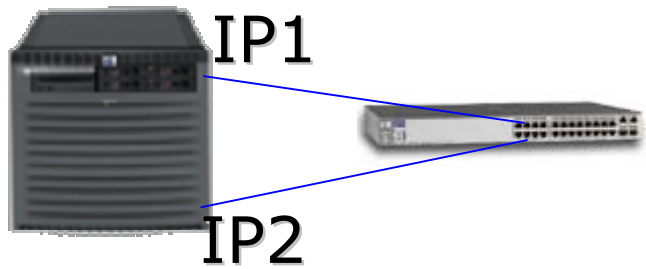
- NIC Scaling



LAN Cards

- APA (Automatic Port Aggregation) versus Samba interfaces

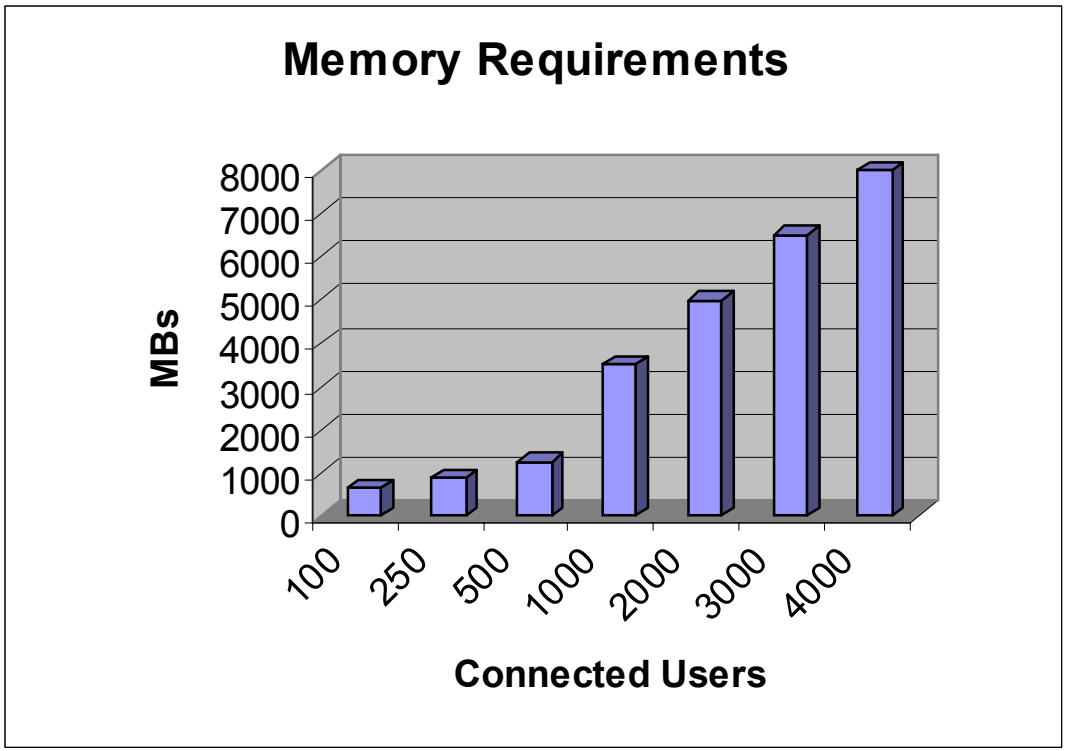




- APA
  - multiple GbE
  - Single IP
  - Auto load balance
- Samba Interfaces
  - multiple GbE
  - multiple IP
  - manual load balance
  - see “interfaces” in smb.conf
- Samba Interfaces about 10% faster than APA

System Hardware

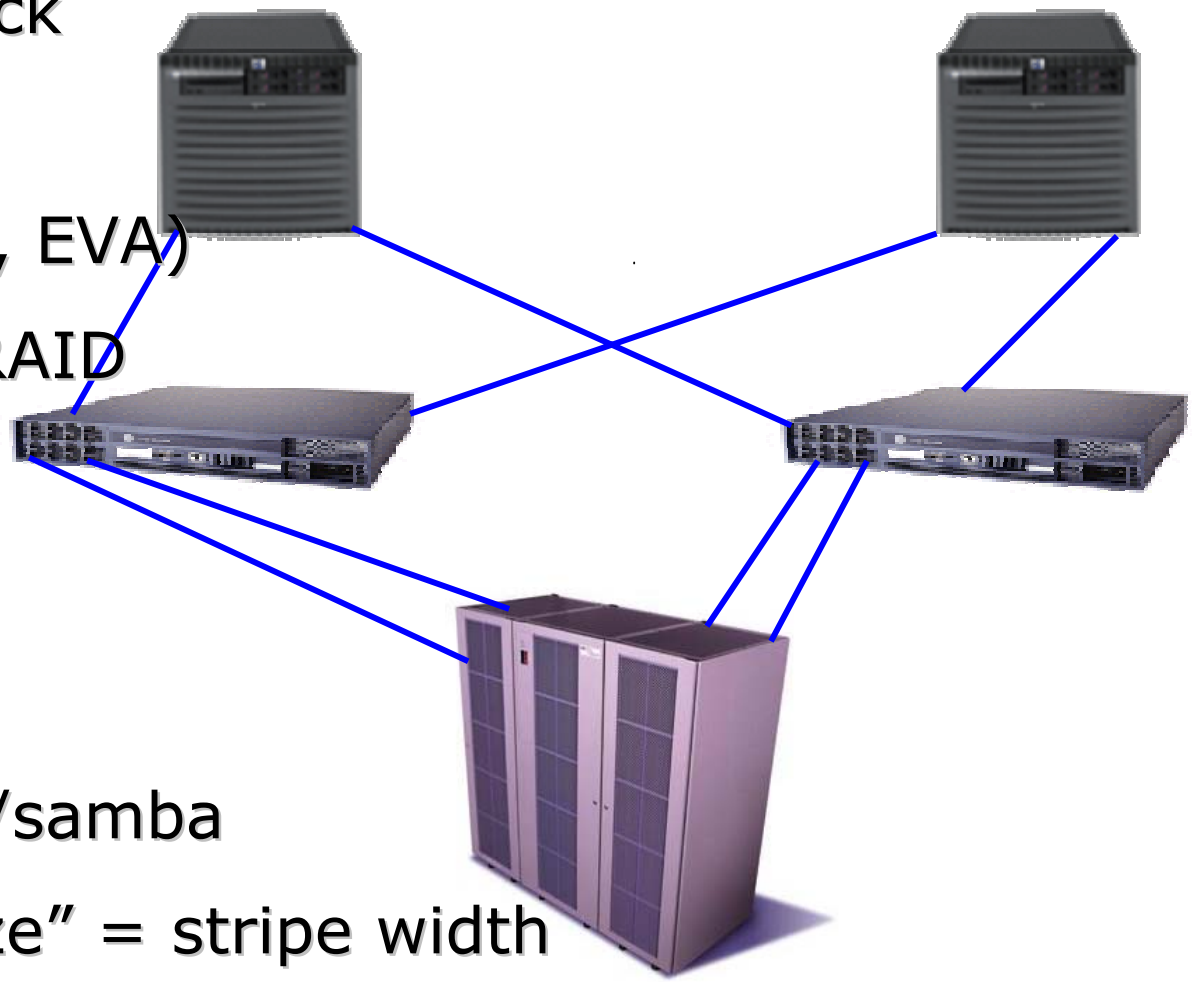
- Memory Utilization



- 1.5MB per client
  - connected
  - Windows client
  - Adjust for non-Windows
- Buffer Cache
  - <500 clients 500MB
  - >500 clients 2GB
  - test
  - adjust for load
- System-wide
  - adjust for NFS
  - adjust for other apps

Disc Access

- Eliminate bottleneck
- Dual FC cards
- Storage Array (XP, EVA)
- RAID 0,1,5: AutoRAID



- Samba
  - Stripe data
  - Stripe /var/opt/samba
  - "write cache size" = stripe width
    - likely 32768 or 65535

Agenda

- CIFS/9000 Server Tuning Strategy
- SMB Architecture
- Samba smb.conf Tuning
- HP-UX for CIFS/9000 Server Tuning
- **Microsoft Client Effects**
- Windows Infrastructure Effects
- Summary

Roaming Profiles

- Roaming Profiles: user profile stored on file server
  - copied to client at every logon
  - copied back to server at every logoff
- Bigger profile = slower logon
- Bigger profile = slower logoff
- Items on Desktop stored in profile
  - Folders are BAD
  - Documents are BAD
  - .pst (Outlook folders) are VERY BAD
- Look for stealth apps
  - Outlook Express stores viewed attachments on profile

Roaming Profiles

- Typical profile folders
  - My Documents (yikes!)
  - Application Data
  - Desktop
  - Start Menu
  - Templates
  - NetHood
  - Favorites
- Note: Outlook .pst and/or .pab on Samba works fine
  - But not on a roaming profile

Client Applications

- Client Multi-User Database Applications
  - Access, Clipper, Fox Pro, etc...
- Require:
  - oplocks = no
  - level2 oplocks = no (to make testparm look okay)
- Note: locking is enabled
- Things to try:
  - lock spin time = 15 (default is 10)
    - interval to retry failed lock
  - lock spin count = 50 (default is 2)
    - retry attempt limit

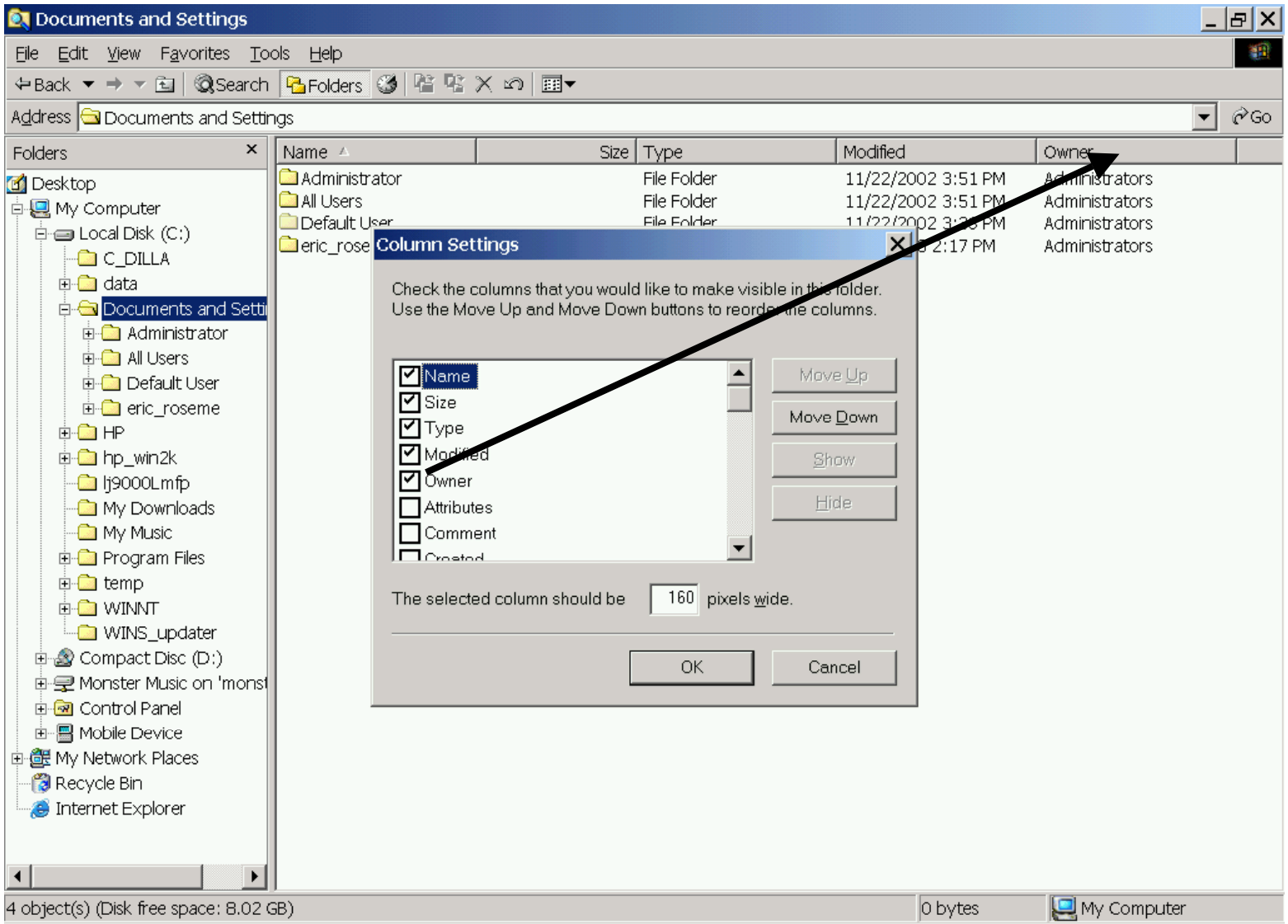
Office Applications

- Word, PowerPoint, Excel, ...
- Oplocks = yes (read only, exclusive access)
  - These apps copy served source file to local work file cache
  - Write changes at “Save” “Save As”
  - Typically exclusive write access
- Word default cache size is 64MB
- .doc files larger than 64MB must be read from network
- Maximum cache size is 3.5MB
  - Decreases network file access
  - Registry setting Q212242

File Explorer (not IE)

- Use default Windows options
- Do not requesting additional metadata fields
  - severe system loading due to increase of stat64 calls
- Case of selecting “owner”
  - with very large directory
- see next slide

File Explorer (not IE)



KeepConn

- KeepConn Windows client registry parm
  - Keeps client connection active
  - Not the same as Samba keepalive
- Maintains client connection to server
- Custom client apps that disconnect and reconnect
  - Connection establishment is very expensive
  - See later topic
- KeepConn discontinued by Microsoft at W2000
  - hotfix for W2000 provided
  - MS Knowledge Base Article 818704

Agenda

- CIFS/9000 Server Tuning Strategy
- SMB Architecture
- Samba smb.conf Tuning
- HP-UX for CIFS/9000 Server Tuning
- Microsoft Client Effects
- **Connection Establishment**
- Summary

Connection Establishment

- Discovery
  - locate domain controller(s)
- Name resolution
- Authentication
- Authorization
- Session Setup

Connection Establishment

- Connection Establishment can be a significant bottleneck
  - Client boots and/or logs on network
    - DNS: discover domain controller
    - Authenticate to DC
  - Client maps share
    - DNS/WINS/Files: resolve file server
    - Start session setup on file server
    - File server authorizes user
  - File Server authorizes user
    - WINS/Files: resolve Domain Controller
    - Connect to DC
    - Authorize user
  - Process user service request

Connection Establishment

- Potential bottleneck
- Multiple systems and networks
- Peak usage periods: everybody does it at once

- Typical scenario at a University
  - Class starts
  - everybody logs on – across entire campus
  - Operate for 45 minutes
  - everybody logs off
  - Repeat

Connection Establishment

- Samba portion of connection establishment
  1. negprot (negotiate the protocol we will use)
  2. sesssetupX (authenticate the user)
  3. tconX (identify/provide the “service” – share/printer)
- map \\emonster\eroseme
  - name resolution to emonster
  - negotiate protocol (negprot) to \\emonster
  - authenticate user (sesssetupX)
  - name resolution to authentication server
  - negotiate protocol to authentication server
  - finish authentication
  - provide the requested data (tconx)
  - authorization for resources

Connection Establishment Name Resolution

- NetBIOS
  - Do not disable NetBIOS (W2000, W2003)
  - NetBIOS itself is not a performance inhibitor
  - NetBIOS “chatter” is a specious argument
    - Windows on SMB floods the network with superfluous traffic
    - NetBIOS is not a significant contributor by comparison
  - Microsoft recommends *against* disabling NetBIOS for native
- nmbd
  - Samba name resolution service
  - Always listening
  - Therefore, exhibits CPU utilization even on idle system

Connection Establishment Name Resolution

- nmbd
  - monitors IP traffic for name queries
  - When it receives unrecognizable traffic, logs it
    - /var/opt/samba/locks.unexpected.tdb
  - On a dirty LAN can drive CPU utilization to 98%
- Fixed in CIFS/9000 Server version A.01.09.05
  - does a much more efficient process of logging
- Upgrade to current version: A.01.10

- Example environment
  - 15,000 clients on network
  - 4,500 clients on one CIFS/9000 Server

Connection Establishment Name Resolution

- nmbd
  - tell nmbd where to look for names
  - smb.conf
- name resolve order =
  - default: name resolve order = lmhost host wins bcast
  - best to place your name resolution device first
  - best to have a working backup – place it second
  - if using WINS, do not leave this at default
    - optimize efficiency
    - wins lmhosts host bcast
- Optimize for thousands of connects
- Connect flooding

Connection Establishment Name Resolution

- WINS
  - Windows Internet Name Service
  - Replaced by W2000 DDNS
  - Still commonly utilized for NetBIOS (Samba)
- Current Samba limitation of single WINS server config
  - default: wins server = no
  - example: wins server = servername
  - Samba enhancement: secondary server
  - example: wins server = servername1,servername2
    - coming at Samba 3.0
    - Your priority for this enahncement?

Connection Establishment Name Resolution

- WINS
- Single WINS server exposure
  - If WINS is unavailable
  - Example:
    - Windows admins take WINS server down for maintenance
    - Assume that all clients/systems have Windows backup config
    - CIFS/9000 Server now unavailable
- Single WINS server bottleneck exposure
  - requires network access
  - requires WINS database lookup
  - often WINS server is old and/or loaded
  - thus bottleneck, slow connection establishment

Connection Establishment Name Resolution

- lmhosts
- flat file holding NetBIOS name to IP mappings
- resides on CIFS/9000 Server
- simple, fast, effective
- Usually only requires domain name and DC name
  - often, these rarely change
- Do not overlook this option!!
- At least use as WINS backup
- Must add DC mapping and Domain Name mapping
  - 192.168.1.1 DCname
  - 192.168.1.1 domainname#1c

Connection Establishment Authentication

- password server =
  - *
  - server1 server2
  - IP1 IP2
- password server = *
  - broadcasts DOMAIN<0x1c> to scan for domain controllers
  - most versatile, flexible config
  - not a significant network load, but is often eschewed for....
- password server = server1 server2 (or IP1 IP2)
  - these indicate direct DC lookups
  - but will be tried in descending order
  - could take longer
  - often the cause of rejected sessions

Connection Establishment Authentication

- NIS
- NIS user authentication requires sequential user lookup
- Larger list = longer lookups
- Excessively long maps create significant delays
- Example:
  - 1.8MB map – severe slowdown
- Update timing
  - Do not push new map to NIS slaves during work hours
  - Blocks user logons

Connection Establishment Authentication

- add user script
- script for adding UNIX accounts at initial client logon
  - Existing Windows domain user
  - UNIX account created
  - Home directory and share create
  - Quota created
- Incoming connections blocked
  - while new smb.conf is being written
  - all other user logon is blocked
  - can delay connection establishment
  - especially for large smb.conf, peak logon period

Agenda

- CIFS/9000 Server Tuning Strategy
- SMB Architecture
- Samba smb.conf Tuning
- HP-UX for CIFS/9000 Server Tuning
- Microsoft Client Effects
- Connection Establishment

- **Summary**

TEASERS!

- LDAP
- Samba 3.0
- Kerberos
- CXFS

HP-UX 11i release roadmap

- Binary compatible with HP-UX 11.0
- Introduction of OEs
- Performance leadership

- #1 rated OS by DH Brown
- VARBusiness OS of the year
- Performance improvements

- Open source application bundle
- Security boost

HP-UX11i | #1 UNIX® in the world | Internet Express

HP-UX 11i version 3
Combined PA-RISC and Itanium®-based release

- Full Enterprise Release
- Cell OL*
- TruCluster technology
- AdvFS
- 128-way support
- Source and Binary compatible

PA-RISC ONLY
Itanium®-based servers ONLY
PA-RISC and Itanium®-based servers

HP-UX 11i version 2

- Full Enterprise Release
- All OEs
- 64-way scaling
- Cell local memory
- Binary compatibility with HP-UX11i v1.6
- Dynamic kernel tunables
- HP-UX for Itanium® 2-based and beyond releases
- nPars

HP-UX 11i version 1.6

- Mission critical
- ISV friendly
- Binary compatible with HP-UX 11i v1.5
- HP-UX for Itanium® 2-based servers

HP-UX 11i version 1.5

- 1st in the industry
- HP-UX for Itanium®-based servers

Current as of July 9, 2003 –HP-UX future plans subject to change without notice HP-UX 11i roadmap – Customer Viewable

Interex, Encompass and HP bring you a powerful new HP World.