NFS Performance

for

System Administrators

John Corbin

corbin@sun.com

Network Resources

Sun Microsystems Computer Corp.

December 8, 1992

Overview

- NFS Components
- Measuring NFS Performance
- Client Mount Options
- Identifying Client Bottlenecks
- Identifying Network Bottlenecks
- Identifying Server Bottlenecks
- Async Server Mount Options
- NFS Over WANs
- Futures

Definitions

- Network File System (NFS)

  A simple remote procedure-based protocol that allows remote access to file data and file attributes.

- File Data

  Data contained within a file.

  Reading/Writing file data is a relatively slow operation which includes file attribute operations.

- File attributes

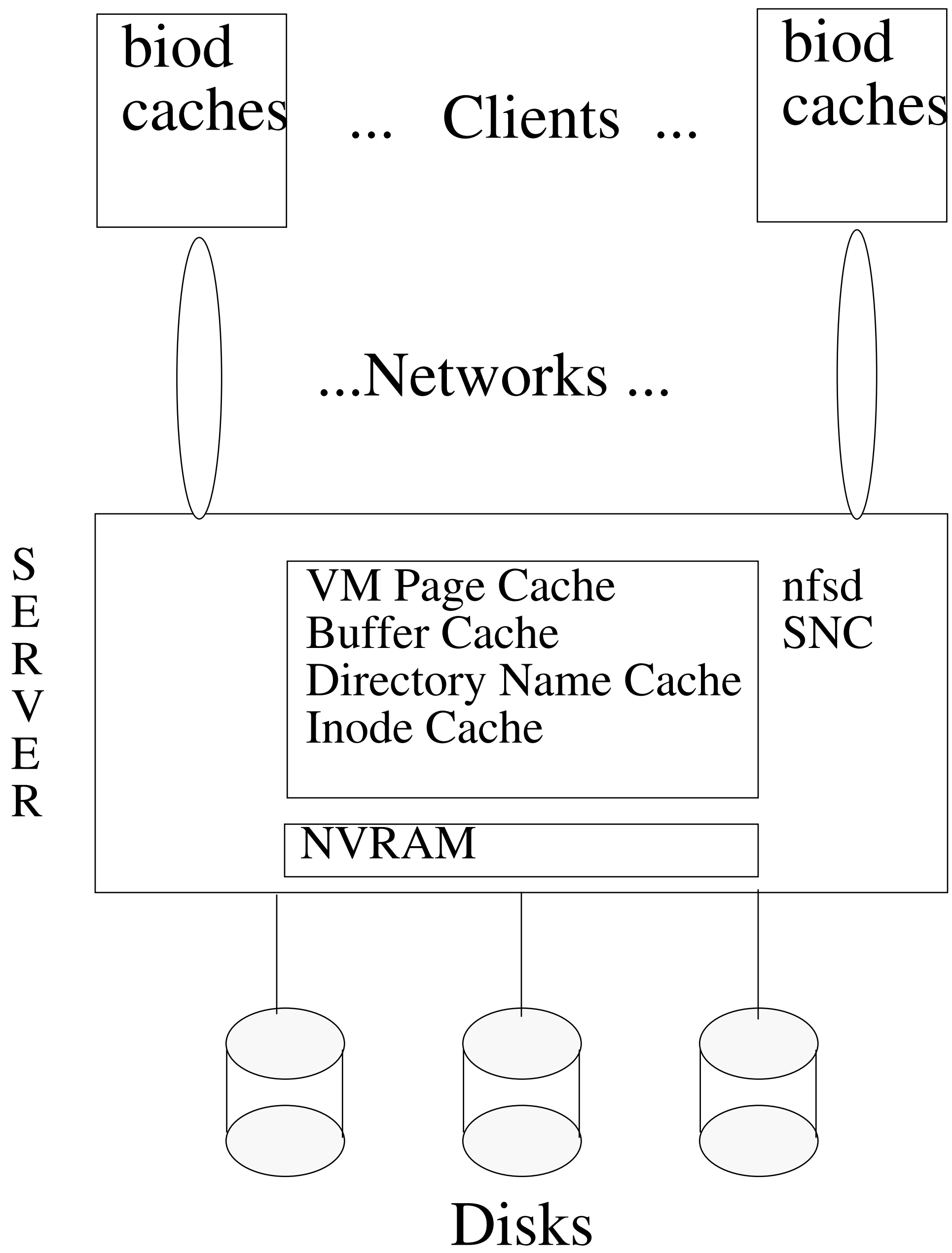
  Also referred to as file meta-data.

  EX:

  file size, file owner, read/write/execute permissions

  Reading/Writing file attributes is a relatively fast operation.

NFS Components



Measuring NFS Performance

- User's Perception

    Most important measure of NFS performance.

- NFS Performance Metrics

- Benchmarks

    nhfsstones

    SPEC LADDIS

        Simulated NFS Work Loads - Not your work load!

Measuring NFS Performance (cont.)

- What really matters?
  - Response Time
    - The server's responsiveness to a client's NFS request.
    - Getting a server with faster response time is like putting a faster disk on a Unix Workstation.
  - $/NFSOPS
    - Server cost / Server Capacity
  - Capacity
    - Response time is a function of Capacity.
    - Server's operating in the saturation region typically have poor response times.

- Using NFS benchmarks to support purchasing decisions
  - Don't use the default benchmark parameters!
    - Use your work load parameters.
  - Don't rely on SPEC published results!
  - Watch out for sales people overselling capacity.
    - Right sizing is important!
  - Expect server performance breakthroughs ~ every 2 years

Measuring NFS Performance (cont)

- Nhfsstone 1.X (from Legato)

  Uses Client OS file operations to generate NFS requests.

  Metric: NFSOPS @ Average Response Time

  Availability: anonymous ftp from legato.com

- Advantages

  NFSOPS generated by real a NFS client implementation

  N processes per client to try to simulate a full subnet load

- Disadvantages

  To generate correct load, must defeat NFS client caches.

    Would you do that on your NFS Client? Of course not!

  Read/Write operations throttled by NFS client

    Fails to simulate R/W load in excess of number of workstations used. Possibly tune number of biods to increase load. Side effects unknown at this time.

  Simulated work load

    It is not your work load!

Measuring NFS Performance (cont.)

- Nhfsstone 2.X (from Legato)

  Uses user-level RPC calls to generate NFS requests.

  Metric: NFSOPS @ Average Response Time

  Availability: Not really available.

- Advantages:

  Really can simulate a full subnet load of NFS traffic using just 2 workstations.

- Disadvantages:

  Read/Writes done in 4KB blocks instead of 8KB.

  Does illegal NFS operations during initialization phase

    Ex: Remove Directory using RFS_REMOVE instead of RFS_RMDIR.

  Simulated work load

    It is not your work load!

Measuring NFS Performance (cont.)

- LADDIS - Future SPEC Benchmark

  Uses user-level RPC calls to generate NFS requests.

  Metric: NFSOPS @ 50 msec

  Availability: Hopefully in 1993, missed 1992.

- Advantages:

  Really can simulate a full subnet load of NFS traffic using just 2 workstations.

  Attempts to emulate BIOD behavior.

  Lots of parameters available - good benchmark tool.

- Disadvantages:

  Default parameter selection lacks justification.

  Current version (0.1.16) too disk I/O intensive.

    Will sell a lot of hardware. You may not need it though.

  PRELADDIS 0.1.3 != LADDIS

    Not even close!

  Simulated work load

    It is not your work load!

Measuring NFS Performance (cont.)

- Read/Write tests

    Metric: KB/sec

    Make sure you fsync the file before getting the final time stamp.

    Use multiple readers/writers to get a good view of the servers ability to handle this type of load.

    How many? Depends on your environment.

Client Mount Options

- noquota

  Prevent ***quota*** from checking if user is over quota on this file system. Default is noquota.

  Quotas will still be checked if enabled on the server.

- actimeo=N seconds

  Set amount of time to time-out cached file attributes.

  Use large value for read-only file systems.

- ncto

  Suppress getting fresh attributes when opening a file.

  Use for read-only file systems.

- noac

  Suppress attribute and name caching.

  Never set unless working around a bug.

Client Mount Options (cont)

- rsize = n bytes, wsize = n bytes

  Set read or write buffer sizes.

  Sometimes necessary to adjust when there are several routers between the server and the client. If adjusted improperly, it will hurt NFS performance.

- timeo = N tenth of seconds

  Set the NFS request time-out. Sometimes necessary to adjust when there are several routers between the server and the client or when you have an consistently over saturated server.

Finding The Bottlenecks

- Where is the Bottleneck?

    Client

    Network

    Server

- Tools

    iostat

    netstat

    nfsstat

    vmstat

    sharpshooter (from AIM technologies)

- Simple, identify and fix.

    Your ability to identify a problem is only as good as your tools.

    Take lots of samples

    Don’t get into high frequency tuning.

Potential Client Bottlenecks

- User's Path Variable

  Directories on local file systems first

  Critical directories on remote file systems next

  Then rest of remote file systems

- Dynamic Retransmission?

  SunOS 4.1.X, Solaris 2.X

  Try nfsstat -m

  Flags: hard int dynamic read size=512, write size=512, count = 5

  Note that read and write size are only 512 instead of 8192. If client and server are not on the same subnet then check for network problems between client and server. If they are on the same subnet then try turning off dynamic retransmission.

  Patch kernel nfs_dynamic=0 to turn it off.

Potential Client Bottlenecks (cont)

- Monitor Caches

  Use ***netstat, nfsstat, vmstat***

  Discussed in more detail on the server side.

  More memory can help.

- Running executable from an NFS mounted file system

  SunOS uses demand paged executable.

- Number of **biod**s (SunOS 4.X only)

  If doing a lot of reads and writes, increase from 4 to 6.

      Watch out for server network buffer overflows.

  Except on an SS2, IPX, ELC -- keep it at 4.

  If NFS requests are going through a store & forward device (like a router), then keep it at 4.

Potential Client Bottlenecks (cont)

- nfsstat -rc

  Watch out for badxid and time-outs.

  badxid indicates the number of times a client didn't wait long enough for server reply.

Client rpc:

calls	badcalls	retrans	badxid	timeout	wait	newcred	timers
1826	0	0	475	500	0	0	41

  If number of badxids is about the same as the number of time-outs and if more than about ~5% of total calls then you may need to do something.

  Identify all the clients that are seeing large # of time-outs when mounting file system. Do they share a common server? Common exported file system? Do a bottleneck check on server?

  If you cannot do anything to improve the servers response time then try the following time-out series on the clients- 25, 50, 100, 200 (tenths of seconds). Wait a full day between modifications to see if the number of time-outs is decreasing.

  Eliminating client retransmissions can indirectly improve server performance.

NFS Server Not Responding

- ping server

  No Response

    Server is down

    Network to server is down

  Response

    Server could be in a panicky state

    Server is over saturated - find server bottlenecks

    Network could be slow - find network bottlenecks

Network Bottlenecks

- Layout Important

  Routing NFS requests increases response time.

  Try to keep the clients on the subnet that is directly connected to the server.

- Is the wire hot?

  High Collision Rate = # Collis / # Opkts > 0.05

  Try:

  ```
  netstat -i; rand_sleep 3600; netstat -i
  ```

Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Collis	Queue
le0	1500	ZZ-net	ZZ	70207	1	12687	0	831	0
lo0	1536	loopback	localhost	702	0	702	0	0	0

Potential Server Bottlenecks

- Directory Name Cache

    % vmstat -s

    ...

    41656 total name lookups (cache hits 83%)

    too long 1161

    If hit rate is low (< 85%), increase MAXUSERS (max is ~128)

- Page Cache

% vmstat 60

procs	memory		page	disk	faults	cpu
r b w	avm	fre	re at pi po fr de sr	s0 s1 s2 s3	in sy cs	us sy id
2 0 0	0	1108	0 10 1 10 0 16 0	0 100 0 0	49 314 45	6 74 20

Take a large sample (several hours)

Look for large CPU system time (CPU bound)

Look for large CPU user time

Look for high paging rate (more memory needed?)

Potential Server Bottlenecks (cont)

- Buffer Cache

  # ***buf_cache_stat***

  Buffer Cache Statistics Since Last Boot

37	Buffer Cache Size
68616	total buffer cache read requests
66634	hits in cache (97% hit rate)
1750	times an aged buf was allocated
813	times an lru buf was allocated
0	times had to sleep for a buf

  Might have to patch kernel to increase number of buffers if hit rate lower than 85%.

  SunOS 4.X: **nbuf**

  Solaris 2.X: **max_bufhwm**

Potential Server Bottlenecks (cont)

- Presto Cache (NVRAM)

  Use ***presto*** utility to check cache efficiency.

  Not a lot you can do to tune it.

- ***nfsd (SunOS 4.X)***

  Run 8 - 16 per Ethernet connected but I would hesitate to go higher than 64 on a 490 or 690. If you see too much context switching then try patching nfs_wakeup_one_nfsd to 1.

  Solaris 2.X doesn’t require tuning.

Potential Server Bottlenecks (cont)

- Inode Cache

  # ***ino_cache_stat***

  Inode Cache Statistics Since Last Boot

786	Inode Cache Size
21422	Cache hits
5651	Cache misses
3080	Allocated from heap
2428	Memory freed from heap
786	Largest size of cache

  Hit Rate = Cache hits / (Cache hits + Cache misses)

  If hit rate is less than 85%, then try increasing MAXUSERS.

  A tuned directory name cache will usually result in a tuned inode cache.

Potential Server Bottlenecks (cont)

- I/O Hot Spots

% iostat -D -l 10 60

	sd1			sd2			sd3	
rps	wps	util	rps	wps	util	rps	wps	util
10	0	20.0	5	60	80.0	6	12	40.0

Take a large sample before doing anything (~ 2weeks)

Balance the load across disks.

For the above example, move some of the active load from sd2 to sd1.

How to find active load

Monitor file access/modification times

Use network snooper to see which client(s) are heating up the disk.

Sharpshooter should help here.

Average disk utilization > 75% implies a busy disks

Add disks when necessary to reduce overall avg disk utilization.

Potential Server Bottlenecks (cont)

- Network Related Problems

% netstat -s

udp:	
0 incomplete headers	**Network Problems**
0 bad data length fields	**Network Problems**
0 bad checksums	**Network Problems**
0 socket overflows	**Client's Overdriving Server**
tcp:...	
ip:	
20405 total packets received	
0 bad header checksums	**Network Problem**
0 with size smaller than minimum	
0 with data size < data length	
0 with header length < data size	
0 with data length < header length	
5570 fragments received	
0 fragments dropped (dup or out of space)	
21 fragments dropped after timeout	**Network Topology**
0 packets forwarded	**Server Routing**
0 packets not forwardable	
0 redirects sent	
0 ip input queue drops	

Potential Server Bottlenecks (cont)

- Sun Network Coprocessor

  Use **nestat**

    Watch for high Maximum CPU utilization and high mbuf utilization.

    Indicates NFS requests are starting to queue on the board, this increases response time.

  Load balancing requires moving client(s) to another subnet. May not be feasible.

NFS Server Configurations

- SS2 (SunOS 4.X)

  32 - 64MB

  1 MB Presto (no presto for read-only file systems)

  1 SCSI Controller

  1 X450 Ethernet SBUS card (2 nets total)

  2 - 4 1.3 GB SCSI Drives

  Makes a great read-only file server -- /usr/local

- 600 Series (SunOS 4.X)

  64 - 128MB

  2 CPUs

  1 MB SBUS Presto [VME Presto when using IPI]

  3 SCSI/Buffered Ethernet (SBE) (for SCSI controllers)

  N 1.3 GB Drives

  3 - 8 SNC

NFS Server Configurations (cont)

- 600 Series (Solaris 2.X)

  64 - 256MB

  2 - 4 CPUs

  1 MB SBUS Presto

  2 - 3 SBE SBUS cards

  N Elites

  Will have 3 - 4 Ethernet, use SNC to get more Ethernets.

Async Write Mount Option

- Some NFS server vendors supply async write mount option.

  Gain --> Performance

  Async writes are a lot faster than sync writes.

  Lose --> Could Lose Data

  Client doesn't know about the async write.

- Fundamental Problem

  Problem exists because the client assumes that when it gets a reply to an NFS write, the data is now on the server's stable storage. The client will then throw away its copy of the data. It doesn't know that the data is not in servers stable storage and that the data could be lost.

- Correct Solution

  NFS protocol should be revised to allow async writes.

NFS Over WANs

- Dynamic Retransmission?

  Turn it OFF!

  Patch kernel nfs_dynamic=0

- Several Router Hops (Large LANs or MANs)

  Server in the next building and client has a high retransmission rate:

  Check badxid. If high, then client is not waiting long enough and packets do not appear to be dropped. Follow badxid fix.

  If badxid not a problem then perhaps packets are getting dropped. Use netstat -s to check number of dropped fragments on the server. If the number is high then cut rsize/wsize in half until problem goes away. This reduces NFS performance but so do retransmissions and the large packet size can be congesting your routers.

  Be careful! Sometimes it is difficult to trace server side statistics from one client.

- Across Country (WANs)

  Set timeo=200, rsize=1024, wsize=1024.

Future

- Decent Administration Tools

    1- 4 years

- Self Tuning Operating Systems

- Performance will go through the roof

    Need faster networks.

    Who needs 4800 NFSOPS?

        Who needed 50 MIP workstations?

    What happen to decentralization?

        Technology

        Organizational Boundaries

        System Administration Overhead

References

"Managing NFS and NIS", Hal Stern, O'Reilly & Associates, Inc. (ISBN 0-937175-75-7).

"Solaris System Administrator's Guide", Janice Winsor, Ziff-Davis Press, (ISBN 1-56276-080-7)