



Go further, faster™

Understanding and Managing Performance





Class Outline and Timeline

- Day 1: Little's Law, What is Performance
Domain and Domain Utilization
The WAFL Filesystem
- Day 2: What about Reallocate?
The Importance of Case Framing
Action Plans
Tools
- Day 3: Looking Inside a filer
Case studies



Module 1:

Little's Law, What is Performance?

- Latency
- Concurrency
- Throughput



Performance Manifesto

- **Technical support is the act of providing a fix to something that is broken.** For example, a hard drive that no longer responds to commands is "fixed" by replacing the defective unit.
- **The resolution of performance issues** are often more difficult, in part due to abstract definitions of "performance".
- The goal of any performance analysis is to understand the **limiting factors of a given workload.**



Latency, Throughput and Concurrency

- **Latency:** The amount of time to complete a unit of work. This is also known as *response time* or *round trip time*.
- **Throughput:** A measure of the quantity of service performed per unit time. For instance, the number of CIFS operations performed per second.
- **Concurrency:** A value measuring outstanding work, both queued, in flight and in service.



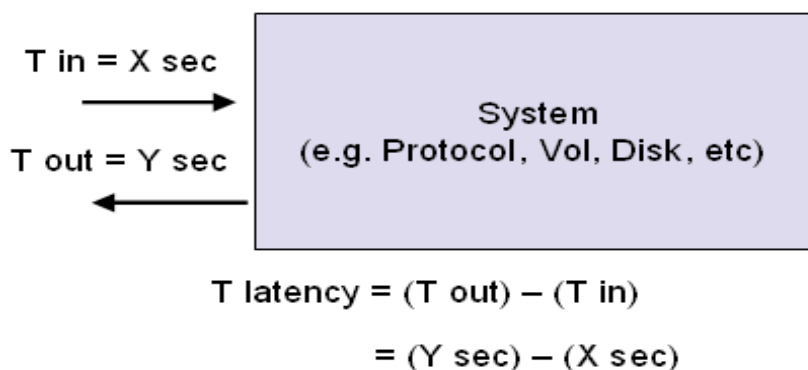
What is Concurrency?

- Concurrency is the number of outstanding requests waiting to be processed at any one point.
- Calculated as: $\text{Op/s} \times \text{Latency (s)}$
- Concurrency values < 1.0 indicates that the client-side application performing the I/O is probably single-threaded.
- Concurrency values < 1.0 is not an efficient way to do I/O.



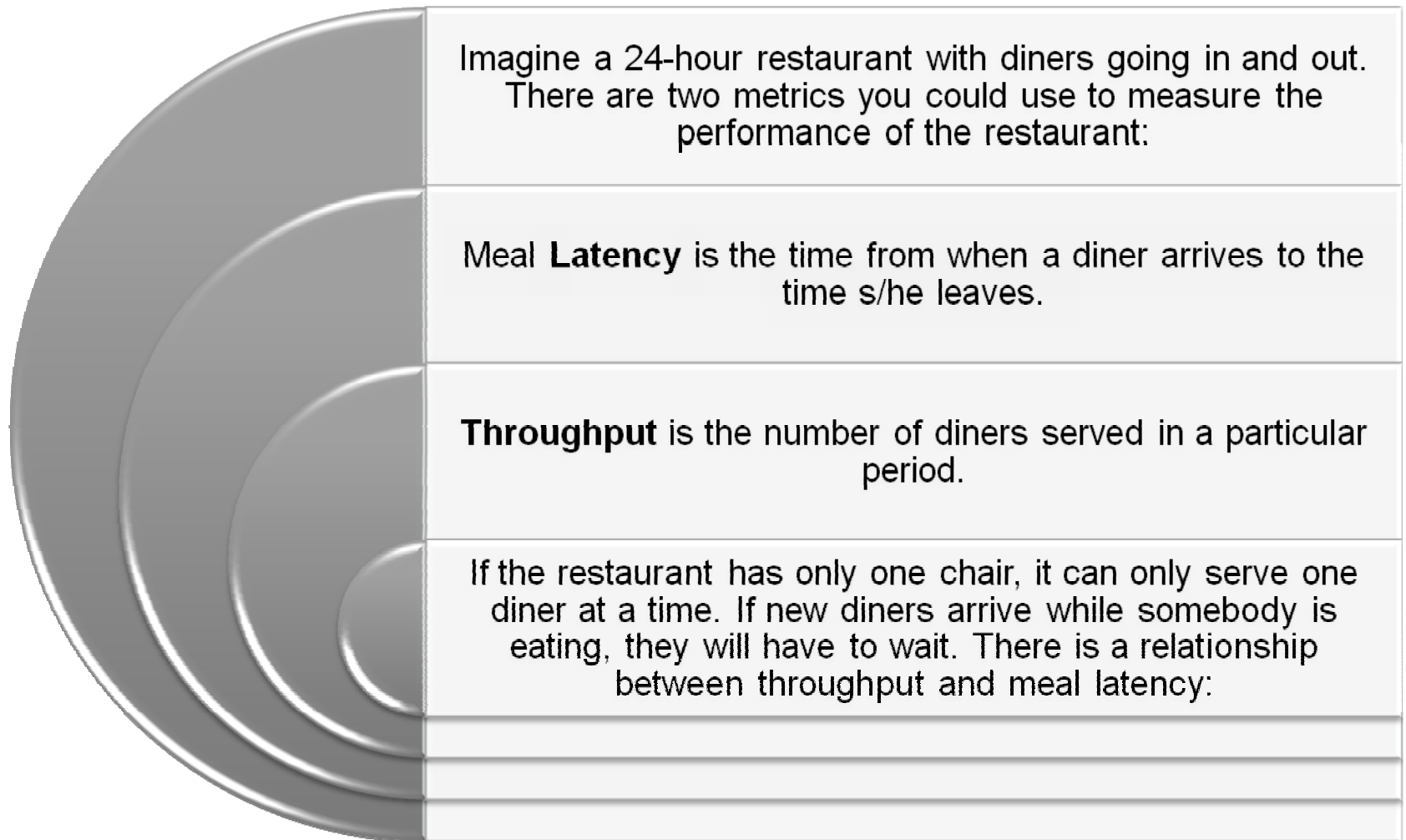
What is Latency

- Latency is the amount of time it takes to perform a unit of work. This is also known as *response time* or *round trip time*.
- Composed of wait (or queueing) time plus service time.
- Latency is only relevant when work is being performed.





Analogy - Measuring Restaurant Performance





The Real Thing - Measuring Filer Performance

Now imagine a highly-available filer handling NFS requests from a bunch of clients. There are two metrics you could use to measure the performance of the filer:



Response Time is the time from when an NFS request is made to the time the response is received by the client.



Throughput is the number of NFS requests completed during a particular period.



What is an *Acceptable* Latency?

- Acceptable Latencies are defined by the customer and their applications.
- However, one often should use the application vendors recommends as a reference.
 - Oracle's latency threshold (i.e., 20ms)
- Different applications may have different latency thresholds.



What is an *Acceptable* Latency? (cont.)

- So with the previous statements made, are the latencies shown below “acceptable?”

vol	Iteration	Volume Name	Read Ops(/s)	Read Data(b/s)	Read Latency(ms)
vol	2	r3_data_prd_mirror	1,052	20,962,896	112.35
vol	4	r3_data_prd_mirror	1,144	15,329,629	98.87
vol	1	r3_data_prd_mirror	837	22,917,022	84.58
vol	3	r3_data_prd_mirror	687	18,887,840	72.24
vol	5	r3_data_prd_mirror	942	17,758,017	60.47
vol	1	exchange_data	811	3,513,462	7.22
vol	2	exchange_data	630	2,792,909	6.87
vol	3	exchange_data	780	3,467,621	6.83
vol	5	exchange_data	665	2,974,939	6.49
vol	4	exchange_data	928	4,115,081	5.92

- Exchange_data is serving luns via FCP to windows server to exchange.
- The customer is also running database tests over FCP to other clients from this same filer.



What is an *Acceptable* Latency? (cont.)

- It is important not to take latencies at face value. This customer is used to seeing lower latencies.

vol	Iteration	Volume Name	Read Ops(/s)	Read Latency	Read Data(b/s)
vol	4	exchange_data	1211	2.21	5705192
vol	3	exchange_data	1086	1.94	4982149
vol	2	exchange_data	1121	1.55	4942796
vol	1	exchange_data	1199	1.51	5299950
vol	1	r3_data_prd_mirror	0	0	0
vol	2	r3_data_prd_mirror	0	0	0
vol	3	r3_data_prd_mirror	0	0	0
vol	4	r3_data_prd_mirror	0	0	0

- With the database test workload removed, the latencies normalized at approximately 2ms.
- The customer was actually complaining about the performance of their exchange environment ***with the added database workload.***



Common Causes of Latencies.

- The single largest cause of high latency is workload!!!
 - In some way the workload has changed, and changed does not necessarily mean grown (i.e., type vs. volume).
 - The filer was not sized properly and the workload either in quantity or makeup was never correct.
 - The filer has outgrown it's original sizing
 - Or the client(s) have not been tuned properly and are having an impact.
 - Network is not tuned correctly or has outgrown traffic
- Filer configuration changes
 - Adding disks/expanding aggregates and/or tradvols without running 'reallocate' afterwards.
 - Improper 'priority' schedule (where available).
- Environmental changes
 - Network configuration, changes, or upgrades.
 - SAN fabric configuration, changes, or upgrades.
 - Client-side configuration, changes, or upgrades.





Start from the top

- Start with volume level latencies, then narrow down by protocol and operation
- In general, if the latencies are low, the perfstat is from the wrong time or the issue is external to the filer
- What is tracked by stats stop –l perfstat vol?
 - Cifs, nfs, san [iSCSI & fcp], flexcache and volume stats
 - Snapmirror/snapvault and backup performance are not



Volume level latencies:

Stats stop -l perfstat vol (How to Read)

- Search for volumes by <protocol>_write_ops or <protocol>_read_ops.
- **Skip volumes with no read or write activity**

volume:h15:nfs_read_data:5825085b/s
volume:h15:nfs_read_latency:0.84ms
volume:h15:nfs_read_ops:178/s
volume:h15:nfs_write_data:6104538b/s
volume:h15:nfs_write_latency:155.47ms
volume:h15:nfs_write_ops:186/s
volume:h15:nfs_other_latency:26.42ms

- Read Concurrency = $178\text{ops/s} * (0.84\text{ms} * 1\text{s}/1000\text{ms})$
 - Read Concurrency = 0.15ops
- Write Concurrency = $186\text{ops/s} * (155.47\text{ms} * 1\text{s}/1000\text{ms})$
 - Write Concurrency = 29ops



Conclusion and Key Concepts

- **Latency:** The amount of time to perform a unit of work. This is also known as *response time* or *round trip time*.
- **Throughput:** A measure of the quantity of service performed per unit time. For instance, the number of CIFS operations performed per second.
- **Concurrency:** A value measuring outstanding work, both queued, in flight and in service



Break

“I am a flightless Bird. Who needs wings when you're cool?”





Module 2: Domain and Domain Utilization

- Domain and Domain Utilization
 - How they Work
 - Understanding Utilization



What is an ONTAP “Domain”

- ONTAP is divided into several domains:
 - Network
 - Storage
 - Kahuna (the default)
 - and more.

- Domains are used to perform work across multiple CPU's in parallel.

- Use sysstat
 - The cpu column calculation varies by release.



Domains and Processing

Objectives:

- Define a domain
- Determine if processes are domain bound using appropriate tools (e.g., `syststat -M` and `ps`)



But what if my CPU is at 100%?

- CPU is just a resource, like memory and disk
 - Utilization is an OBSERVATION not a CONCLUSION
- *CPU Utilization* is the percentage of time a CPU is not executing the idle thread or “the percentage of time a CPU is busy”

```
filer*> ps -c 1
```

```
Process statistics over 194150.079 seconds...
```

ID	State	Domain	%CPU	StackUsed	%StackUsed	Name
1	RR	i	98%	808	19%	idle_thread0



But what if my CPU is at 100%?

- High CPU utilization is simply an indication of high CPU utilization
 - Does not indicate a bottleneck by itself!
- The Data ONTAP scheduler prioritizes user work and system work on demand
- Fine, how do I tell if CPU utilization is a problem then?
 - In short, look for increases in measured operational latency



Summary

You should now be able to:

- Define a domain
- Determine if processes are domain bound using appropriate tools (e.g., `sysstat`)



Break

“I am a flightless Bird. Who needs wings when you’re cool?”





Module 3: The WAFL Filesystem

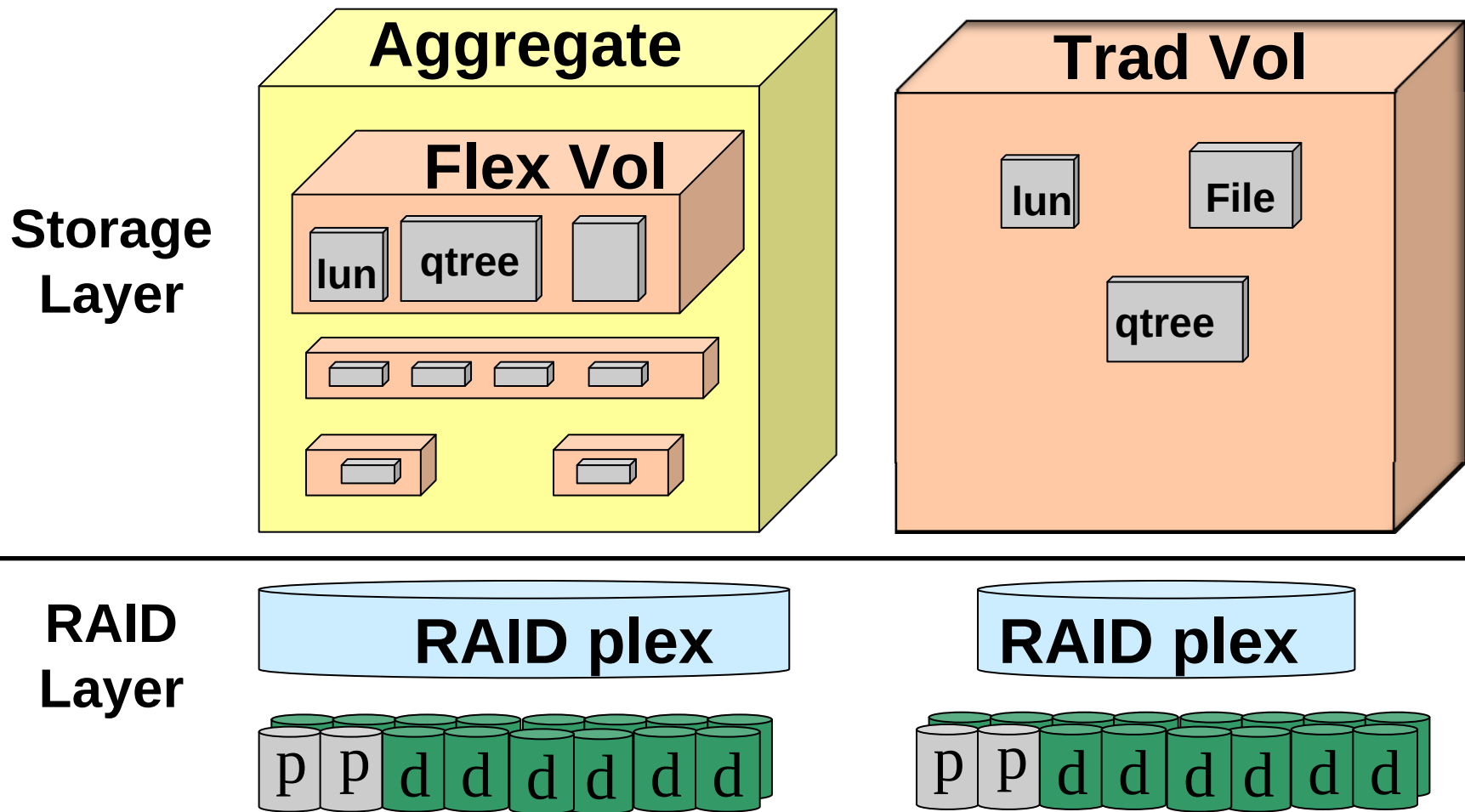
- Tradvols versus Flexvols
- Inodes
- Swizzling and deswizzler
- Maxfiles
- Directories



WAFL - Write Anywhere File Layout

- WAFL stores data in 4 KB blocks with no fragments.
- WAFL uses inodes to describe its files.
- Directories are specially formatted files.
- WAFL uses files to store meta-data.

Aggregates + FlexVols vs. TradVols





Volumes and Aggregates

- Aggregates
 - Container for FlexVols
 - Drive spindles shared by all FlexVols
 - Size is currently limited to 16TB



Volumes and Aggregates

■ Volumes

- A WAFL file system image, plus snapshots
- Volume size is currently limited to 16TB
- Number of volumes is limited by release platform
- Two flavors: flexible and traditional



Inodes

- An inode is a data structure that is used represent file system objects (files and directories)



Inodes

- An inode consists of the following:
 - Type of file (regular file, directory, link, etc)
 - Size
 - Owner, group, permissions
 - can store complete file data if less than 64 bytes
 - head of the file's data



Inode File

- All inodes are stored in a special file
- Each 4K inode block references up to 21 inodes
- `maxfiles` limits the size of this file
 - may be grown, never shrinks



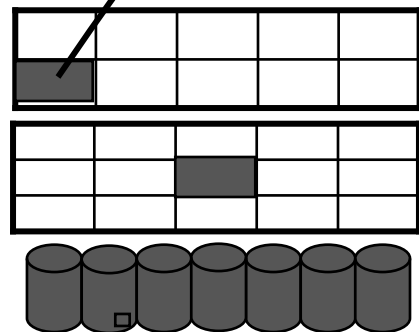
Swizzling

- Virtual blocks are mapped in the Flexvol to the aggregate physical block.
- During a *vol copy* and *Volume SnapMirror* the virtual to physical mapping is invalidated. This is known as swizzling.
- De-Swizzling restores the virtual to physical relationship after the copy has completed.



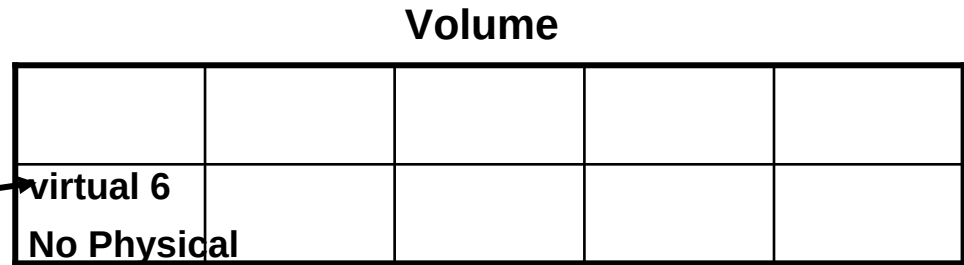
How the volume gets swizzled

SnapMirror process sends the block to Destination Filer. Destination Filer puts the block in same virtual location, but the destination physical location is unknown to the source.

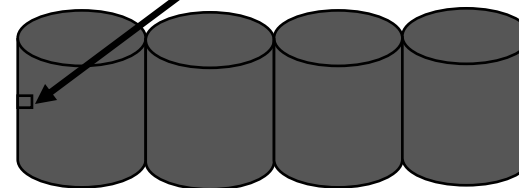
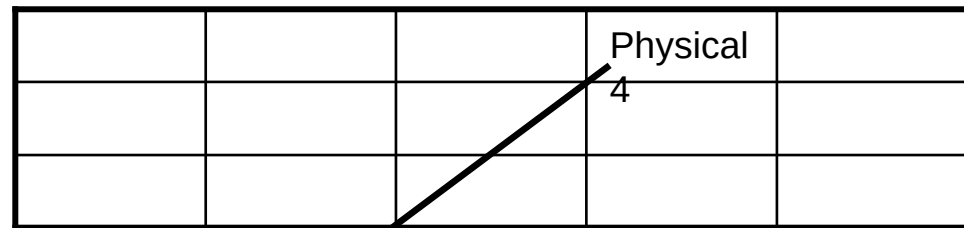


Source Filer

Aggregate



The source send the block to the destination which writes it to disk. The Flex volume file is not updated with the physical disk location.



Destination Filer



Maxfiles, The man page definition

- maxfiles increases the number of files that a volume can hold, as close as possible to *max*. File inodes are stored in blocks, and the filer may round the requested *max* number of files to the nearest block. Once increased, the value of *max* can never be lowered, so the new value must be larger than the current value.
- If no argument is specified, maxfiles displays the current value of *max* for all volumes in the system. If just the *vol_name* argument is given, the current value of *max* for the specified volume is displayed.
- Because each allowable file consumes disk space, and because the value of *max* can never be reduced, increasing *max* consumes disk space permanently.



- maxfiles *[vol_name]* *[very large number]*"

- ```
filer> maxfiles vol0 90000000000000000000000000000000000000
```

```
filer> maxfiles
```

Volume vol0: maximum number of files is currently 3,097,061 (3766 used).



# Maxfiles: Reference commands

## ■ df -i

- The following example shows file system inode usage for a specified volume:

```
filer> df -i /vol/vol0
```

| Filesystem | iused | ifree     | %iused | Mounted on |
|------------|-------|-----------|--------|------------|
| /vol/vol0  | 3766  | 3,093,295 | 1%     | /vol/vol0  |



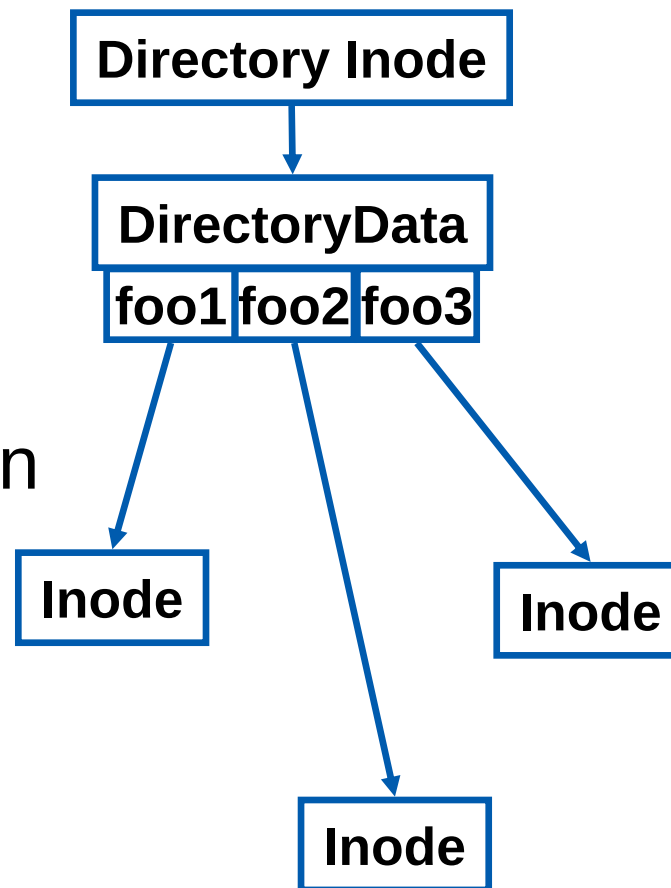
## What does *maxfiles* do?

- Controls the number of inodes that can be allocated on the volume.
- It is recommended that this is not set any larger than needed.



# Directories Are Just Files

- Files that have directory data
- Maps names to inodes
- At most 99,998 subdirectories
- More than one directory can point at an inode
- Hard link - the directory entry for a file





# Break

“I am a flightless Bird. Who needs wings when you're cool?”





## Module 4: What About Reallocate?

- Spatial Locality?
- The benefits of Reallocate
- Reallocate “the command”



# Spatial Locality and REALLOCATE

- By the time you finish this module, you should:
  - The potential affect of random write and sequential reads
  - What is Reallocate and when is it useful?
  - Be able to identify when a reallocate may be called for and when it is not
  - Understand that a number on its own is simply an indicator, Latencies and Concurrency are guiding factor.

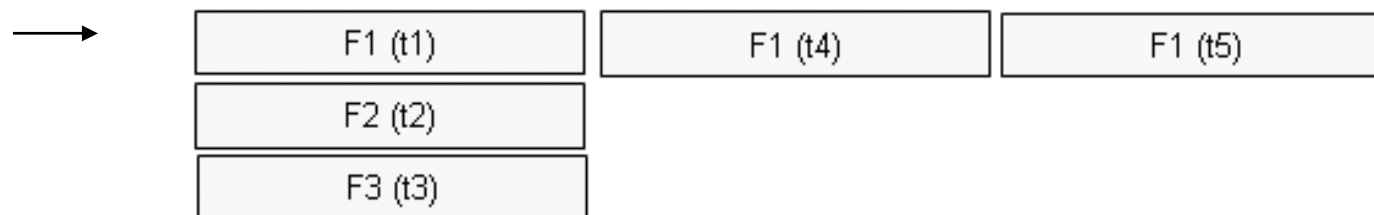
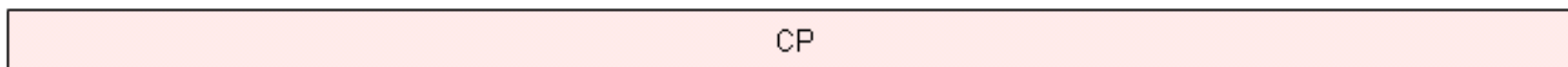
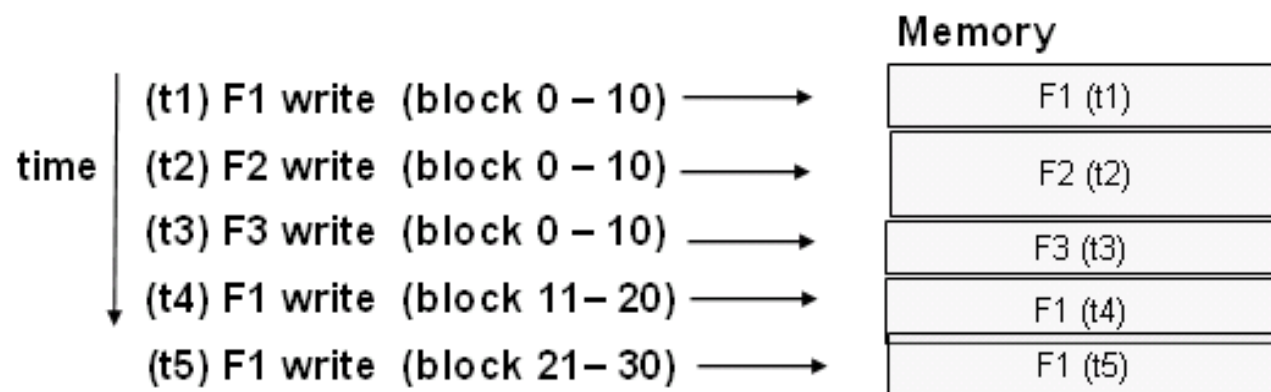


# What is spatial locality

- Blocks that come in together as part of the same file are located close to each other when written to disk during each CP.
  - Block migration does occur overtime: i.e data blocks naturally migrate away from each other as file systems age and as files are modified over time.
- However, some applications that perform random updates also make extensive use of sequential access patterns. For these applications, sequential data layout may deliver superior performance. This optimization comes at no cost to random read or write performance....
  - Ipso Facto: Reallocate was born



# An example Spatial Locality





# Why then Reallocate?

- Reallocate was designed with a very specific goal: to improve sequential read access to data not written sequentially.
- Reallocate will not benefit write performance or random read performance



# What workloads may benefit?

- Write workloads that are randomly written but sequentially retrieved: (Such As)
  - Databases that combine OLTP transactions with large table scans, such as in business analytics; or OLTP environments with large-scale reporting
  - E-mail systems that store updates in a database style format and perform verifications of the data in off-production hours.
  - Host-side backup of an entire LUN or structured data set



## What else can benefit from Reallocate

- After adding disks to a raidgroup
  - When adding disks to an aggregate, after the disks are added, the load is rebalanced between the old and new disks; however, the process can take a long time. Use of `reallocate -f` can hasten this process.
  
- Following client side defrag of luns:
  - If a host-based defragment utility is run on a NetApp LUN, `reallocate` can be run afterward to restore the sequentiality of the LUN data.



## How do we gauge “spatial locality”

- How do we gauge the degree of spatial locality of the data blocks.
  - WAFL scan measure\_layout
  - Or
  - reallocate measure\_layout (7.2.4+)
- The measure\_layout scanner actually measures the 'spatiality' of each block within a container



# What about Aggregate level Reallocate?

- The reallocate -A command does aggregate level freespace reallocation. This is different from the other forms of the reallocate command.
- If your intention was to reallocate all of the files in all of the volumes on an aggregate, then you need to run the reallocate command on each volume in the aggregate.
- This command is rarely used, and is not a substitute for the reallocate command at the volume or file level.
- This command is not to be used following the addition of disks to a raidgroup or the otherwise expansion of an aggregate or tradvol.
  - If you are doing this after adding an additional raid group to the aggregate or after adding disks to raidgroups, be sure to use the -f option.



## Concept Repeated

- Remember:
  - Performance is all about response times.
  - YOU defines acceptable and un-acceptable response times.
  - The `measure_layout` value is meaningful only for sequential workload.



# Break

“I am a flightless Bird. Who needs wings when you're cool?”





## **Module 5: Framing a Case OR I have a client complaint ... now what?**

- Case Framing, or understanding the customers complaint is often critical for success in resolving performance issues.
  
- It is important to understand:
  - What is the meaning of “SLOW”
  - How “SLOWness” is felt
  - Etc...



# Frame the case with the Client.

- I have a client complaint ... now what?
  - Accurately define the issue – what is “slow”?
  - Customers define performance expectations, what's slow to them “may” be acceptable to others. We need to understand what the customer means by “slow”
    - How does the customer describe the issue?
      - Was performance acceptable at one time?
      - degradation over time?
      - new install and/or performance was never good?
    - What is the business impact?
      - production system? testing? Number of users affected?
    - What is slow?
      - Are reads slow / writes slow / browsing of file or directories /file creations/mounts?
      - How are you judging “slow”?
    - Who is it (define it) “slow” for?
      - How many clients and which clients by name and ip and OS?
      - Which protocols are clients reporting a problem for? NFS|CIFS|Backups|Snapmirror|FCP|ISCSI
      - Which volumes and or luns are affected?
    - What is the configuration (hosts and filer)?
      - mount options, automounter, network
      - Is the filer using nis (nis+) or AD.
      - What security style is being used on the affected volume/qtree?
      - what applications are involved and what are the applications doing?
    - What is the purpose of the filer?
      - Is the filer primary storage or a dr filer?
      - Is the filer a test and productions filer? This happens often.
    - When is performance a problem?
      - Only during the mornings when users are logging in?
      - Only when backups are run?
      - Only when snapmirror transfers are occurring?
      - Etc...
    - What has changed recently? (Don't forget to ask this)
      - Just because they say nothing doesn't mean they are correct.



# Validate Configuration and ID changes

## Reviewing Asups

- Start by reviewing the Asups for the filer in question. Issues such as the following can be addressed by Asups: [Asup Premium](https://now.netapp.com/willows/login.do) (<https://now.netapp.com/willows/login.do>)
- Looking at /etc/messages, are there any indications of hardware or any issue that could be related / contribute to the customers reported problem? **There is no reason to expect good performance from a sick filer.**
  - Configuration changes:
    - Has there been a recent Ontap upgrade.
    - Have licenses been added (additional licenses is an indicator that workload may have been added)?
    - Were new volumes added or current expanded?
    - Were disks added to raidgroups and/or raidgroups added to tradvols/aggregates? If this is the case was reallocate ran against the volumes in the aggregate or the the volume itself be it a tradvol? (refer to this document's index concerning Reallocate best practices)
    - Do you see space changes according to df -h?
    - Do you see inode count increasing according to df -i?
    - Does the filer have minra enabled on the volume(s)? With Ontap 6.5+ minra should be off.
      - Enabling minra (minra on) disables intelligent readahead!!!.



# General Filer Utilization “ops/hour”

- Historical data is valuable:
  - Use a simple tool track the delta between hourly protocol stats as reported by /etc/messages.
  - <PATH>/asupTrend.pl – A PERL script that allows the user to extract per-hour protocol ops from the messages file by manually copying the contents of the /etc/messages.” The output can then be loaded into MS Excel and graphed for visual trends.

Syntax:

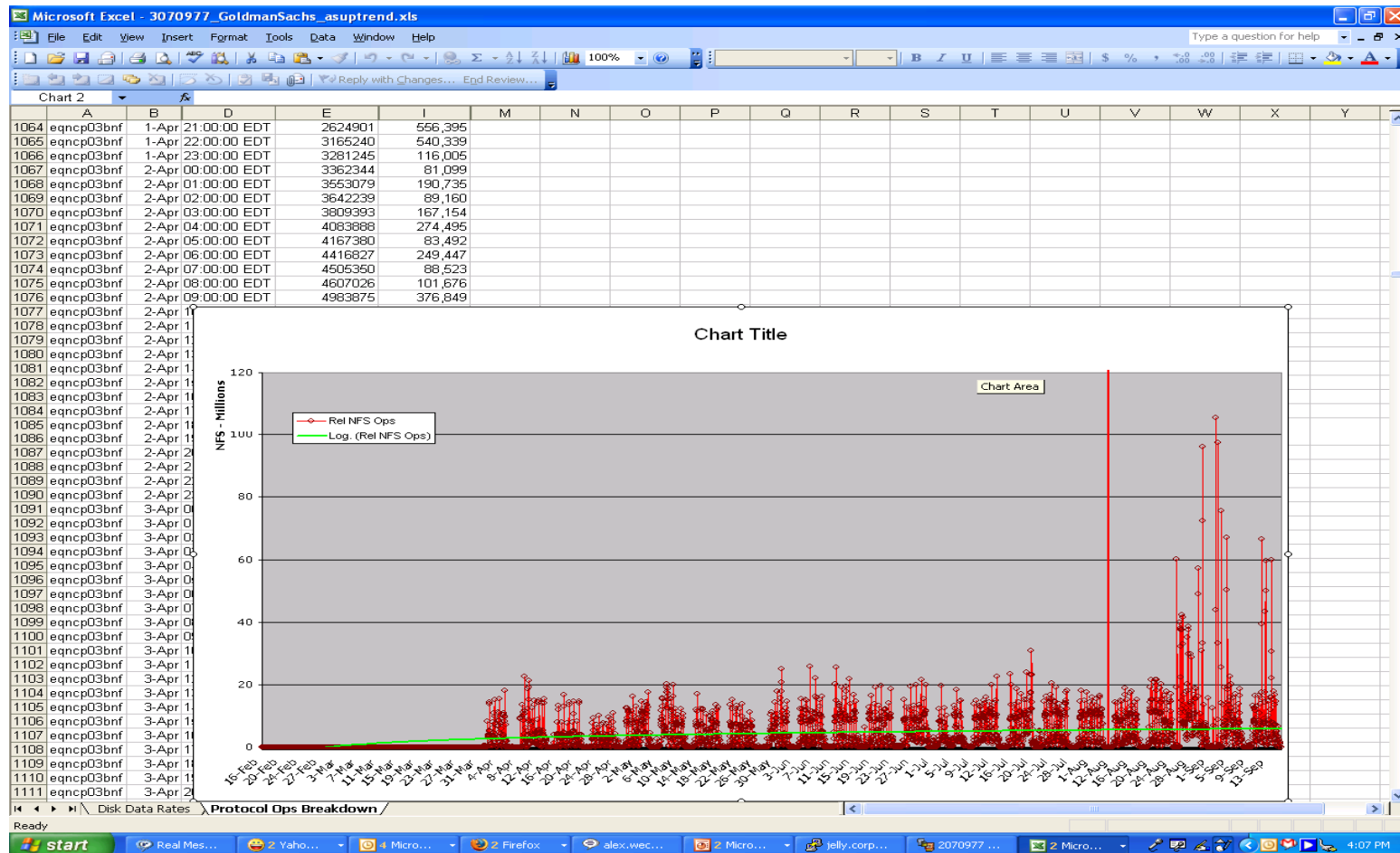
```
asupTrend.pl -f <source file> > <outputfile.xls>
```

Example:

```
asupTrend.pl -f messages.txt > sample_trend.xls
```



# The results of asupTrend.pl





## Module 6: Action Plans

- Agenda:
- Identify action plans for:
  - CPU
  - NFS
  - CIFS
  - Backups
  - SAN
  - Snapmirror



## Besides reviewing asups, Working High Cpu Issues (are they or are they not Perf related?)

- High CPU alone does not indicate a problem. It is not unusual in ontap 7.2+ to see kahuna domain at 95% utilization when the filer is not doing many user ops (its efficient that way) With that said, run the following commands in the order shown:
  - `priv set diag`
  - `ps -z`
  - `sysstat -x -c 30 -s 1`
  - `sysstat -M -c 30 -s 1` (run this for multi-cpu system; collect this at the same time as the previous command)
  - At the end of sysstat collection interval (30 seconds) m
    - `ps -c 1`
    - `ps -cs 1.`
  - `waf scan status`
  - `priv set`
- The filer is displaying high cpu because it is doing work. We run these commands above to see the work the filer is actually doing. High cpu without observed slowness is not a performance problem.
  - Some internal tasks, such as WAFL scanners, consume all available CPU, provided that no data protocols are demanding resources.
    - In other words, on a not so busy filer, it's not unusual for the CPU to report 100% ... if say, a deswizzling scan is running on a volume.
    - If a user request is made, the filer will give that request priority
- If the filer has recently been upgraded to 7.2.X (from 7.0.X or 7.1.X), you can expect cpu utilization to be high for some period of time, the amount of time depends on many factors. This affects the smaller filers such as the fas270, fas920/fas940 and fas3020 more so it is advisable to have a baseline perfstat of the filer at its loaded period prior to upgrade.



# Besides reviewing asups, Working NFS Perf Cases

- Have the customer tell you whether the problem is happening now or not and then collect the following
  - Enable nfs.per\_client\_stats: options nfs.per\_client\_stats.enable on (make sure to turn it off afterwards)
  - Zero out the nfsstat : nfsstat -z
  - Wait five minutes
  - Generate an asup: options autosupport.doit casenumber
  - Collect the nfs top talkers data: nfsstat -l and nfsstat -h (See the script on the next slide).
- If the problem is not occurring, just collect the asup
- Review asups for :
  - configuration issues, configuration changes, hardware problems
  - any messages similar to “nfs operation took longer than 60 seconds.”
  - Look for nis or dns issues which suggest that the problem may not be an nfs perf problem but a name resolution perf problem
    - nis or name resolution errors in messages, look for trends identifying when they occur.
    - Nis\_info and dns\_info looking specifically at how long the last three look ups took suggestive of slow nis or dns servers.
- Collect the mount options using the mount command from at least one client reporting performance issues.
  - If the issue is related to Oracle, check the following KB article for the proper mount options appropriate the client platform.  
[Oracle Mount Options](#)
  - Glaring mount options to be aware of, non-oracle:
    - Netapp recommends tcp for nfs
    - Always use hard mounts (where the application allows).
    - noac (on Solaris) and/or sync makes writes synchronous which slows writes down dramatically.
    - actimeo=0 disables attribute caching, this results in more metadata operations that would not be necessary otherwise.
    - If there are a large number of gettats, say 50%, consider increasing actimeo to a value larger than 60 seconds which is the default max of acreg and acdirmax.
    - Any nfs rsize or wsize less than 32K.
- If the customer is using netgroups (have them tell you if they are using nis or nis+)
  - make sure that an @ sign is placed before every netgroup within the exports file.
  - Exportfs -r to reread the exports file into memory
  - Enable the option nfs.netgroup.strict on
- Useful tools: netdiag, ping -s (hostname), getXXbyYY



# Sample nfsstat collection scripts

## ■ Collection script for client side nfsstat stats:

```
date > /tmp/data.nfsstat
for ip in `rsh filer_ip "nfsstat -l" | awk '{print $1}'` ; do
 rsh filer_ip " nfsstat -h $ip" >> /tmp/data.nfsstat
done
```

## ■ Parser for client side nfsstat stats

– tool\_used\_to\_parse\_nfsstat\_client\_data.pl

|    | Name Box                     | A         | B       | C      | D    | E       | F       | G      | H      | I        | J    | K     | L      | M     | N       | O     | P      | Q     |
|----|------------------------------|-----------|---------|--------|------|---------|---------|--------|--------|----------|------|-------|--------|-------|---------|-------|--------|-------|
| 1  | date                         |           | ipaddr  | client | null | getattr | setattr | lookup | access | readlink | read | write | create | mkdir | symlink | mknod | remove | rmdir |
| 2  | Fri Apr 18 22:09:25 UTC 2008 | 169.z.z.z | zzz.com |        | 0    | 31226   | 16      | 690    | 13354  | 2        | 245  | 83    | 2      | 0     | 0       | 0     | 10188  | 1087  |
| 3  | Sat Apr 19 22:01:47 UTC 2008 | 169.y.y.y | yyy.com |        | 0    | 1529    | 0       | 0      | 230    | 0        | 0    | 0     | 0      | 0     | 0       | 0     | 3047   | 118   |
| 4  | Sat Apr 19 21:56:38 UTC 2008 | 169.y.y.y | yyy.com |        | 2    | 289     | 0       | 13     | 11     | 0        | 0    | 0     | 0      | 0     | 0       | 0     | 1298   | 0     |
| 5  | Fri Apr 18 06:29:36 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 28874   | 3692    | 4150   | 7994   | 0        | 214  | 27976 | 1961   | 0     | 0       | 0     | 801    | 0     |
| 6  | Fri Apr 18 06:24:35 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 14840   | 1726    | 2564   | 6931   | 0        | 263  | 14235 | 1753   | 0     | 0       | 0     | 761    | 0     |
| 7  | Fri Apr 18 06:19:32 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 19720   | 2108    | 4162   | 10210  | 0        | 521  | 20004 | 1992   | 0     | 0       | 0     | 697    | 0     |
| 8  | Fri Apr 18 06:14:31 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 28386   | 3814    | 4581   | 9936   | 0        | 691  | 27896 | 2132   | 0     | 0       | 0     | 687    | 0     |
| 9  | Sat Apr 19 06:30:15 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 16963   | 1848    | 2542   | 7188   | 0        | 153  | 20837 | 1881   | 0     | 0       | 0     | 659    | 0     |
| 10 | Fri Apr 18 17:46:15 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 15304   | 2705    | 2434   | 3509   | 0        | 86   | 14408 | 1115   | 0     | 0       | 0     | 649    | 0     |
| 11 | Fri Apr 18 06:09:22 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 12173   | 1835    | 3137   | 6493   | 0        | 499  | 11870 | 1845   | 0     | 0       | 0     | 643    | 0     |
| 12 | Sat Apr 19 16:59:13 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 6795    | 1294    | 5437   | 4414   | 0        | 273  | 7405  | 1315   | 0     | 0       | 0     | 631    | 0     |
| 13 | Fri Apr 18 08:05:34 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 12288   | 1407    | 3407   | 4467   | 0        | 246  | 11240 | 1410   | 0     | 0       | 0     | 628    | 0     |
| 14 | Fri Apr 18 05:54:17 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 13215   | 1984    | 4425   | 5994   | 0        | ###  | 12016 | 1852   | 0     | 0       | 0     | 618    | 0     |
| 15 | Fri Apr 18 06:04:21 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 15622   | 1993    | 4542   | 6209   | 0        | 895  | 15159 | 1976   | 0     | 0       | 0     | 613    | 0     |
| 16 | Fri Apr 18 08:56:07 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 11046   | 1763    | 2805   | 5387   | 0        | 455  | 10820 | 1576   | 0     | 0       | 0     | 612    | 0     |
| 17 | Sat Apr 19 06:15:04 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 18485   | 1892    | 3915   | 7428   | 0        | 488  | 21885 | 1916   | 0     | 0       | 0     | 596    | 0     |
| 18 | Fri Apr 18 18:01:28 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 16019   | 2692    | 3632   | 3172   | 0        | 85   | 14514 | 1067   | 0     | 0       | 0     | 582    | 0     |
| 19 | Sat Apr 19 16:49:04 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 7300    | 1250    | 3849   | 3875   | 0        | 255  | 5546  | 1265   | 0     | 0       | 0     | 580    | 0     |
| 20 | Sat Apr 19 06:20:06 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 18251   | 2183    | 2694   | 8526   | 0        | 178  | 22450 | 2114   | 0     | 0       | 0     | 579    | 0     |
| 21 | Fri Apr 18 09:11:21 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 13211   | 1450    | 3876   | 6288   | 0        | 435  | 12690 | 1380   | 0     | 0       | 0     | 574    | 0     |
| 22 | Fri Apr 18 00:40:32 UTC 2008 | 169.x.x.x | xxx.com |        | 0    | 8856    | 1310    | 1879   | 4473   | 0        | 358  | 8109  | 1293   | 0     | 0       | 0     | 572    | 0     |



# Nfsstat -d

- Nfsstat -d : nfs reply cache statistics: All entries should be misses, any entries found "in progress" are retransmissions.

NFS reply cache statistics:

TCP:

| In progress | Delay hits | Misses | Idempotent | Non-idempotent |
|-------------|------------|--------|------------|----------------|
| 15          | 0          | 896306 | 0          | 44             |

UDP:

| In progress | Delay hits | Misses | Idempotent | Non-idempotent |
|-------------|------------|--------|------------|----------------|
| 0           | 0          | 0      | 0          | 0              |

- Tcp flow control:
  - If the number of outstanding acks for nfs requests passes a high water mark, the filer will begin throttling nfs requests on a socket by socket basis. Out reflects that the filer has come out of tcp flowcontrol.

```
tcp input flowcontrol receive=9, xmit=5
tcp input flowcontrol out, receive=45, xmit=2
```

- Nfs and the assist queue:
  - Any nfs message that requires pre-processing ( name lookups, netgroup expansions, access verification, mount requests) will end up on the assist queue.

```
nfs msgs counts: on assist queue=0, max on assist queue = 0, cut off for assist queue=2096
```

- On assist queue= current number of nfs messages waiting for pre-processing
- Max on assist queue = highest number recorded following the last reset (look for values == or above cut off value.
- Cut off for assist queue = Any nfs operation received by the filer past the cut off value will be dropped.



# Besides reviewing asups, Working Cifs Perf Cases

- Have the customer tell you whether the problem is happening now or not and then collect the following
  - Enable cifs.per\_client\_stats: options cifs.per\_client\_stats.enable on
  - Zero out the cifs stats : cifs stat -z
  - Wait five minutes
  - Generate an asup: options asup.doit casenumber
  - Run the cifs top command: cifs top -s ops -a now and have the customer send this to you to see what the 20 most active cifs clients are doing.
  
- If cifs clients are timing out and the customer is using nis, this can effect current connections and new connections:
  - Check asup for the option nis.group\_update.enable on? If its not enabled not, enable it.
    - [CIFS access problems apparently due to NIS access](#)
  - In asup look at nis info to see how long the last three nis lookups took.
  
- If there is slow access or lack of responsiveness to the client:
  - Compare the values in the asup for:
    - "Cifs stat: Max Multiplex" against the options value "options cifs.max\_mpx".
      - cifs stat : Max multiplex is a high water mark showing the highest number of concurrent in flight ops seen by any client since the value was last reset.
      - options cifs.max\_mpx is the maximum number of concurrent in flight operations allowed per client by the filer
    - If the number of max\_mpx and max multiplex are at or near one for one, increase the max\_mpx number.
      - The possible values for max\_mpx are: 50, 126, 253, 1124.
      - Microsoft recommends maxing out max\_mpx to 1124.
      - [Increase maximum number of simultaneous windows requests](#)
  - Check for Authentication back log which may be indicative of the DC's falling behind
    - "Cifs stat: Max gAuthQueue depth = X" is the current number of Auth requests queued.
    - If the gAuthQueue is at 25 or greater, consider this a potential performance bottleneck.
      - In the asup, look at cifs domaininfo: Look for Broken PDC
      - Look in messages for attempt to improve dc connections
      - Look at historical asups to see if the current connected dc has changed or been changing
      - Perform a ping -s of the current and leading dc's to make sure that the current dc is still the best dc.
      - Perform a traceroute of the current and leading dc's to make sure that the current dc's are logically close to the filer.
    - This could also be indicitive of name service issues
      - Check the messages file for any obvious indicators of name resolution problems
      - Check dns\_info in the asup for the last three dns lookup times.
    - Ask the customer to verify that the DC's and name resolution (DNS/WINS) are full functional
  
- Are new cifs connections taking a long time connect or timing out, but existing connections are fine?
  - Check cifs stats in asup for pBlk exhaustion. One pBlk is required to track each cifs request, once the filer runs out of pBlks no new connections will be allowed until pBlks are made available. :
    - Max pBlk Exhaust = X where X = the number of times the filer has run out of pblks
    - Max pBlks = 4000 Current pBlks = 4000 Where current pBlks is the number of pblks currently free.
  - Virus scanners have been known to cause pBlk exhaustion. If pBlks are exhausted, check for vscan errors in the messages log, turn vscan off to see if the issue clear.s
  - If mandatory vscanning is turned on, check to see that there are vscanners registered.



# Beside Besides reviewing asups, Working San Perf Cases

- If the case is a SAN case, it is expected that :
  - In addition to asup, collect client (host\_info) and switch (switch\_info) logs
  - SAN Data collection tools are available in the host attach kits
  - Have the customer perform a physical inspection;
    - LEDs normal?
    - Loose cables or dirty connection?
  - If the problem is intermitant, collect a perfstat the next time that the issue occurs. See latter section to understand perfstat collection.



# San Path Misconfiguration

- Have the customer tell you whether the problem is happening now or not and then collect the following
  - Zero out the lun stats : `lun stats -z`
  - Wait five minutes
  - Generate an asup: `options asup.doit casenumber`

## ■ Approach all San Related cases as San Configuration cases:

- The LUN STATISTICS section of the Autosupport will display the read and write operations for both local and partner paths per LUN. This output may also be obtained using the command `lun stats -o`. the LUNs that are being accessed by a non-primary path may reside on the partner node.

- LUN STATS

### ■ ===== LUN STATISTICS =====

|   |                                                            |                |          |           |           |        |                        |
|---|------------------------------------------------------------|----------------|----------|-----------|-----------|--------|------------------------|
| – | /vol/frs/lun03 (11 days, 22 hours, 56 minutes, 49 seconds) |                |          |           |           |        |                        |
| – | Read (kbytes)<br>Kbytes                                    | Write (kbytes) | Read Ops | Write Ops | Other Ops | QFulls | Partner Ops<br>Partner |
| – | 22646080                                                   | 28541652       | 285102   | 728443    | 418572    | 0      | 311918<br>54565        |
| – | /vol/frs/lun02 (11 days, 22 hours, 56 minutes, 49 seconds) |                |          |           |           |        |                        |
| – | Read (kbytes)<br>Kbytes                                    | Write (kbytes) | Read Ops | Write Ops | Other Ops | QFulls | Partner Ops<br>Partner |
| – | 22714266                                                   | 28585423       | 284805   | 729941    | 418572    | 0      | 311919<br>54569        |

- See KB [How to troubleshoot FCP Partner Path Misconfigured errors](#)



# Besides reviewing asups, working backup cases

- If the case is a backup case:
  - First you will need to see that there is an issue:
    - Check asup: messages for hardware problems
    - With the proper logging setup on the filer, have the customer run a level 0 backup “full” at the normal time.
      - Options backup.log.enable debug
      - Ndmpd debug 50
    - To isolate the problem to the filer, initiate a level 0 dump to null of the affected volume.
      - rsh “filer ip” dump 0f null /vol/volume
    - Have the customer send all /etc/log/backup\* and /etc/log/ndmpd\* logs for review
      - Compare the differences between the level 0 backups against the dump null.
      - Look for differences in speed between the ndmp level0 and the dump level 0.
      - This will help you to isolate the issue to the filer or an issue external to the filer.
  - Note which phases are taking a long time.
    - Note that Phase 3 is retrieving metadata and this doesn’t go to tape
    - Note that Phase 4 is retrieving actual data and this does go to tape



# Besides reviewing asups, working backup cases (cont.)

- Once you have identified if the issue is on filer or off, follow the following checklist. If you make any changes, make only one change at a time and then test collecting the logs.
- Check asup for minra=on for affected volumes, minra=on can substantially affect backup speed.
- Understand the configuration of the backup environment.
  - The path that the metadata takes is just as important as the path taken by the actual data. Physical writes of data will wait if the metadata writes get backed up.
  - Make sure that the filer is writing to the tape drives connected to the filer/backup server that the client thinks they are.
  - If the filer being backed up is a vsm destination, are the volumes which are slow vsm destination volumes? Note this.
    - Why this matters
      - (layman's terms) Flexvol VSM data is transferred quickly because the data is layed out in a disorganized manner, reads will be slower as a result. There is a background process (scanner) call deswizzler which organizes the data logically recovering access time.
      - This scanner runs upon completion of each vsm transfer.
      - If this scanner does not complete, it aborts at the begging of the next transfer.
      - Each run will start at the oldest snapshot working upward towards the active filesystem which it does last.
      - If the VSM schedule is too aggressive, it will never organize the active filesystem. This means that the backup of the volume will be noticeably slower.
- How to detect that the scanner is not completing
  - Let the scheduled transfer occur
  - Every five minutes run the command: priv set diag; WAFL scan status
  - Look for deswizzler to run on the volume
  - Run "priv set diag; WAFL scan status" until the next update.
  - If the deswizzler never completes, the schedule is to aggressive and is affecting backups.
- What can be done:
  - Adjust the vsm update schedule to allow deswizzler to complete
  - Consider backing up the vsm primary rather than the secondary.
  - Consider decreasing the number of snapshots on the source volume.
  - Headway on this issue was made in 7.2.4, consider upgrading.
  - Have the account team look at architecture options with the customer.



## Besides reviewing asups, Working snapmirror Perf Cases

- When working snapmirror, always collect data from both the source and destination filers.

*This applies to both asups and perfstats.*

- Collect the snapmirror.log from both sides as well.
- Collect a traceroute from both sides of the relationship looking for latencies.



# Break

“I am a flightless Bird. Who needs wings when you're cool?”





## Module 7: Tools

### LatX.pl, getPerfSection.pl, Perfstats,

- What tools are available for reading Perfstats?
  - LatX.pl (NOT SUPPORTED & NOT SUPPORTABLE)
  - getPerfSection.pl (NOT SUPPORTED & NOT SUPPORTABLE)
- Perfstat output is written to an out file and is “difficult” to read in its raw format.
- LatX.pl and getPerfSection.pl have been designed to assist in this effort.



# First a note on Tools: Latency Extractor

First: LatX.pl

## Purpose:

- To assist in quickly identifying filer workload,
  - To assist in quickly identifying latencies at:
    - Protocol : NFS, CIFS, ISCSI, FCP
    - Internal: Volume (WAFL) and LUN
- And to do so across a multiple iteration perfstat (and we should always collect perfstats using multiple iterations)
- The TEA region developed LatX.pl (Latency Extractor)
- This Script is unsupported.



## LatX.pl: a deeper explanation

- LatX.pl is a perl script written to extract workload and latency information from protocol, volume, and lun sections of the perfstat files.
- With the output of this tool, you will be able to pinpoint specific iterations to investigate further within a multi-iteration perfstat file.
- The output provides summaries of workload and related latencies per perfstat iteration.



# LatX.pl Syntax for Run

## ■ How do I use LatX.pl

Syntax: `unix% ./LatX.pl -f <perf_filename> [-P <protocol>] [-F <out_filename>]`

Sample: `unix% ./LatX.pl -f cmcfile3_ti48_perfstat1_1700.out -P nfs -F _Latx_cmcfile3_ti48_perfstat1_1700.out.xls`

```
LatX.pl -f <perf_filename> [-P <protocol>] [-F <out_filename>]
```

where:

<perf\_filename> - path to Perfstat file to parse for latencies. Required parameter.

<protocol> - protocol(s) to filer for. Defaults to 'none'. Valid values are 'nfs', 'cifs', 'fcip', and 'iscsi'. Use comma-separated list to specify multiple protocols.

<out\_filename> - path to the filtered output file. Defaults to '\_LatX\_<perf\_filename>'.



# LatX.pl A sample output

- A sample output of LatX.pl taken from case 2792609 (JPMC: Slow performance following application change)
- First by volume / Then by protocol

|    | A   | B        | C         | F           | G              | H             | I               | J            | K             | L               | M                | N             | O             | P              | Q                | R                 | S              | T             | U                | V                 |
|----|-----|----------|-----------|-------------|----------------|---------------|-----------------|--------------|---------------|-----------------|------------------|---------------|---------------|----------------|------------------|-------------------|----------------|---------------|------------------|-------------------|
| 1  | vs  | Filer    | Iterative | Volume Name | Avg latency(m) | Total Ops(/s) | Avg Concurrency | Read Ops(/s) | Read Data(kB) | Read Latency(m) | Read Concurrency | Avg Read Size | Write Ops(/s) | Write Data(kB) | Write Latency(m) | Write Concurrency | Avg Write Size | Other Ops(/s) | Other Latency(m) | Other Concurrency |
| 2  | val | cmcfiler | 27.00     | private     | 4.47           | 3595.00       | 16.02           | 1867.00      | 40291797.00   | 8.45            | 15.78            | 21559.61      | 661.00        | 3982332.00     | 0.16             | 0.11              | 6024.71        | 1056.00       | 0.13             | 0.14              |
| 3  | val | cmcfiler | 30.00     | private     | 3.87           | 3801.00       | 14.71           | 1739.00      | 36993993.00   | 8.31            | 14.45            | 21273.14      | 873.00        | 3974961.00     | 0.15             | 0.13              | 4553.22        | 1188.00       | 0.10             | 0.12              |
| 4  | val | cmcfiler | 29.00     | private     | 3.57           | 4063.00       | 14.50           | 1855.00      | 38432456.00   | 7.63            | 14.15            | 20718.31      | 988.00        | 4020060.00     | 0.16             | 0.16              | 4068.89        | 1219.00       | 0.16             | 0.20              |
| 5  | val | cmcfiler | 31.00     | private     | 3.38           | 4010.00       | 13.55           | 1442.00      | 30835111.00   | 9.12            | 13.16            | 21368.75      | 970.00        | 3696107.00     | 0.13             | 0.13              | 3810.42        | 1596.00       | 0.16             | 0.26              |
| 6  | val | cmcfiler | 32.00     | private     | 2.88           | 3776.00       | 10.87           | 1623.00      | 33226373.00   | 6.52            | 10.58            | 20472.20      | 925.00        | 4214350.00     | 0.15             | 0.14              | 4556.05        | 1227.00       | 0.12             | 0.15              |
| 7  | val | cmcfiler | 33.00     | private     | 2.37           | 3873.00       | 9.18            | 1543.00      | 31477432.00   | 5.78            | 8.92             | 20400.15      | 1009.00       | 4017747.00     | 0.14             | 0.14              | 3981.91        | 1320.00       | 0.09             | 0.12              |
| 8  | val | cmcfiler | 28.00     | private     | 2.28           | 4563.00       | 10.40           | 1659.00      | 35726254.00   | 6.08            | 10.09            | 21534.81      | 1155.00       | 3897075.00     | 0.14             | 0.16              | 3374.09        | 1749.00       | 0.10             | 0.17              |
| 9  | val | cmcfiler | 17.00     | private     | 0.94           | 2744.00       | 2.58            | 444.00       | 11336637.00   | 1.92            | 0.85             | 25532.97      | 810.00        | 5283751.00     | 0.15             | 0.12              | 6523.15        | 1490.00       | 1.09             | 1.62              |
| 10 | val | cmcfiler | 26.00     | private     | 0.84           | 3457.00       | 2.90            | 964.00       | 22409509.00   | 2.78            | 2.68             | 23246.38      | 1155.00       | 5327119.00     | 0.13             | 0.15              | 4612.22        | 1336.00       | 0.05             | 0.07              |
| 11 | val | cmcfiler | 21.00     | private     | 0.67           | 3242.00       | 2.17            | 544.00       | 12544719.00   | 3.38            | 1.84             | 23096.91      | 1116.00       | 4838906.00     | 0.14             | 0.16              | 4335.94        | 1580.00       | 0.11             | 0.17              |
| 12 | val | cmcfiler | 1.00      | private     | 0.49           | 3770.00       | 1.85            | 496.00       | 10254125.00   | 3.00            | 1.49             | 20673.64      | 1568.00       | 6344835.00     | 0.15             | 0.24              | 4046.45        | 1706.00       | 0.07             | 0.12              |
| 13 | val | cmcfiler | 2.00      | private     | 0.41           | 2654.00       | 1.09            | 439.00       | 10731732.00   | 1.93            | 0.85             | 24445.86      | 910.00        | 5370412.00     | 0.15             | 0.14              | 5901.55        | 1305.00       | 0.08             | 0.10              |
| 14 | val | cmcfiler | 4.00      | private     | 0.40           | 2378.00       | 0.95            | 233.00       | 4952639.00    | 3.12            | 0.73             | 21255.96      | 1018.00       | 5546602.00     | 0.14             | 0.14              | 5448.53        | 1126.00       | 0.08             | 0.09              |
| 15 | val | cmcfiler | 20.00     | private     | 0.39           | 2977.00       | 1.16            | 500.00       | 12111253.00   | 1.93            | 0.97             | 24222.51      | 1062.00       | 5065403.00     | 0.13             | 0.14              | 4769.68        | 1414.00       | 0.04             | 0.06              |

|    | A   | B         | C         | D          | E             | F               | G            | H                | I                    | J             | K                 | L                     | M          |
|----|-----|-----------|-----------|------------|---------------|-----------------|--------------|------------------|----------------------|---------------|-------------------|-----------------------|------------|
| 1  | nfs | Filer     | Iteration | Time Stamp | Time Delta(s) | Avg NFS Ops(/s) | Read Ops(/s) | Read Latency(us) | Read Concurrency(op) | Write Ops(/s) | Write Latency(us) | Write Concurrency(op) | null Ops r |
| 2  | nfs | cmcfiler3 | 31.00     | 0.86       | (313s)        | 4008.00         | 1443.00      | 9185.75          | 13.26                | 970.00        | 174.99            | 0.17                  | 21.00      |
| 3  | nfs | cmcfiler3 | 27.00     | 0.84       | (313s)        | 3583.00         | 1867.00      | 8524.47          | 15.92                | 661.00        | 214.95            | 0.14                  | 21.00      |
| 4  | nfs | cmcfiler3 | 30.00     | 0.86       | (337s)        | 3798.00         | 1739.00      | 8372.79          | 14.56                | 873.00        | 189.57            | 0.17                  | 23.00      |
| 5  | nfs | cmcfiler3 | 29.00     | 0.85       | (314s)        | 4064.00         | 1856.00      | 7702.08          | 14.30                | 988.00        | 201.90            | 0.20                  | 21.00      |
| 6  | nfs | cmcfiler3 | 32.00     | 0.87       | (314s)        | 5400.00         | 1623.00      | 6576.60          | 10.67                | 925.00        | 184.05            | 0.17                  | 21.00      |
| 7  | nfs | cmcfiler3 | 28.00     | 0.85       | (315s)        | 4561.00         | 1659.00      | 6148.73          | 10.20                | 1155.00       | 184.01            | 0.21                  | 21.00      |
| 8  | nfs | cmcfiler3 | 33.00     | 0.87       | (313s)        | 7750.00         | 1543.00      | 5839.80          | 9.01                 | 1009.00       | 181.88            | 0.18                  | 21.00      |
| 9  | nfs | cmcfiler3 | 21.00     | 0.81       | (315s)        | 3238.00         | 544.00       | 3458.00          | 1.88                 | 1116.00       | 176.65            | 0.20                  | 21.00      |
| 10 | nfs | cmcfiler3 | 4.00      | 0.73       | (316s)        | 2553.00         | 233.00       | 3188.37          | 0.74                 | 1193.00       | 231.69            | 0.28                  | 20.00      |
| 11 | nfs | cmcfiler3 | 1.00      | 0.71       | (321s)        | 3896.00         | 496.00       | 3080.68          | 1.53                 | 1695.00       | 234.98            | 0.40                  | 21.00      |
| 12 | nfs | cmcfiler3 | 26.00     | 0.84       | (312s)        | 3455.00         | 964.00       | 2843.69          | 2.74                 | 1155.00       | 166.83            | 0.19                  | 21.00      |
| 13 | nfs | cmcfiler3 | 5.00      | 0.73       | (333s)        | 2987.00         | 286.00       | 2772.88          | 0.79                 | 1290.00       | 206.07            | 0.27                  | 22.00      |
| 14 | nfs | cmcfiler3 | 11.00     | 0.76       | (315s)        | 2881.00         | 205.00       | 2212.95          | 0.45                 | 1091.00       | 149.78            | 0.16                  | 20.00      |
| 15 | nfs | cmcfiler3 | 12.00     | 0.77       | (315s)        | 2278.00         | 212.00       | 2086.62          | 0.44                 | 936.00        | 169.23            | 0.16                  | 21.00      |



# Tools: Perfstat Section Retrieval

## getPerfSection.pl

- Also: getPerfSection.pl
- Purpose:
  - This perl script will allow the user to quickly filter on an individual section of the perfstat and view all iterations therein.
  - This script will retrieve the single desired section of the perfstat in question across iterations.
- Format:
  - Syntax: `./getPerfSection.pl -f <perfstat file> -s <section>`
  - Example: `./getPerfSection.pl -f cmcfiler3_ti48_perfstat1_1700.out -s`



# Perfstat Sections

- ➡ aggr show\_space
- ➡ backup status
- ➡ cf monitor all
- ➡ cf partner
- ➡ cf status
- ➡ cifs sessions
- ➡ cifs sessions -t -c
- ➡ cifs stat -c
- ➡ cifs top
- ➡ df
- ➡ df -A -h
- ➡ df -i
- ➡ df -r
- ➡ flexcache stats -S -c
- ➡ httpstat -t
- ➡ ic stats performance
- ➡ ifstat -a -v
- ➡ iscsi stats
- ➡ iswt connection show -v iswta
- ➡ iswt connection show -v iswtb
- ➡ iswt session show -v iswta
- ➡ iswt session show -v iswtb
- ➡ license.out

- ➡ lun hist
- ➡ lun\_stats -o
- ➡ lun\_stats.out
- ➡ mbstat
- ➡ mem\_stats
- ➡ mem\_stats -e
- ➡ ndmpd probe
- ➡ ndmpd status
- ➡ netstat -i
- ➡ netstat -na
- ➡ netstat -s
- ➡ nfsstat -d
- ➡ nfs\_hist
- ➡ options.out
- ➡ protocol\_stats.out
- ➡ ps
- ➡ ps -s
- ➡ quota
- ➡ raid\_config -r
- ➡ raid\_config -x
- ➡ /etc/log/backup
- ➡ /etc/log/snapmirror
- ➡ /etc/messages
- ➡ smb\_hist

- ➡ snapmirror status -l
- ➡ statit.out
- ➡ stats perfstat\_aggregate
- ➡ stats perfstat\_cifs
- ➡ stats perfstat\_cifsdomain
- ➡ stats perfstat\_cifs\_ops
- ➡ stats perfstat\_cifs\_stats
- ➡ stats perfstat\_disk
- ➡ stats perfstat\_ext\_cache
- ➡ stats perfstat\_ifnet
- ➡ stats perfstat\_iscsi
- ➡ stats perfstat\_ldap
- ➡ stats perfstat\_lmem
- ➡ stats perfstat\_nfs
- ➡ stats perfstat\_nfsv4
- ➡ stats perfstat\_nvram
- ➡ stats perfstat\_perf
- ➡ stats perfstat\_processor
- ➡ stats perfstat\_qtree
- ➡ stats perfstat\_raid
- ➡ stats perfstat\_raid\_stripe
- ➡ stats perfstat\_readahead
- ➡ stats perfstat\_scsi\_partner
- ➡ stats perfstat\_system

- ➡ stats perfstat\_target
- ➡ stats perfstat\_vol
- ➡ stats perfstat\_vsm\_destination
- ➡ stats perfstat\_wafl
- ➡ stats.out
- ➡ storage show disk -a
- ➡ storage stats tape
- ➡ sysconfig.out
- ➡ sysstat.out
- ➡ sysstatM.out
- ➡ sysstat\_1sec.out
- ➡ version.out
- ➡ vif.list
- ➡ vifstat\_vif1.out
- ➡ vol media\_scrub status -v
- ➡ vol scrub status -v
- ➡ vscan
- ➡ wafl cpstats
- ➡ wafl scan status
- ➡ wafl scan status -A
- ➡ wafl\_susp -r
- ➡ wafl\_susp -w



# Perfstat Sections

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>aggr status -v</li><li>backup status</li><li>cf monitor all</li><li>cf partner</li><li>cf status</li><li>cifs sessions</li><li>cifs sessions -t -c</li><li>cifs shares</li><li>cifs stat</li><li>date</li><li>df</li><li>df -A -h</li><li>df -h</li><li>df -i</li><li>df -r</li><li>dns info</li><li>environment status</li><li>exports</li><li>fcstat device_map</li><li>fcstat fcal_stats</li><li>fcstat link_stats</li><li>hostname</li><li>httpstat</li><li>ic stats performance</li><li>ifconfig -a</li><li>ifstat -a</li><li>ifstat -a -v</li></ul> | <ul style="list-style-type: none"><li>igroup show</li><li>igroup show -v</li><li>iscsi config</li><li>iscsi isns show</li><li>iscsi nodename</li><li>iscsi security show</li><li>iscsi show adapter</li><li>iscsi show initiator</li><li>iscsi stats</li><li>iscsi status</li><li>iswt connection show -v iswta</li><li>iswt connection show -v iswtb</li><li>iswt interface show</li><li>iswt session show -v iswta</li><li>iswt session show -v iswtb</li><li>license</li><li>lun config_check -v</li><li>lun hist</li><li>lun show -m</li><li>lun show -v all</li><li>lun stats -o</li><li>mbstat</li><li>ndmpd probe</li><li>ndmpd status</li><li>netdiag -v -b</li><li>netstat -i</li><li>netstat -m</li><li>netstat -na</li></ul> | <ul style="list-style-type: none"><li>netstat -rn</li><li>netstat -s</li><li>netstat -x</li><li>nfsstat -d</li><li>nfs_hist</li><li>nis info</li><li>options</li><li>options iscsi</li><li>options lun</li><li>printflag</li><li>qtree status</li><li>quota</li><li>/etc/cifsconfig.cfg</li><li>/etc/cifsconfig_setup.cfg</li><li>/etc/cifsconfig_share.cfg</li><li>/etc/dgateways</li><li>/etc/exports</li><li>/etc/hosts</li><li>/etc/hosts.equiv</li><li>/etc/lclgroups.cfg</li><li>/etc/nsswitch.conf</li><li>/etc/rc</li><li>/etc/registry</li><li>/etc/serialnum</li><li>/etc/snapmirror.conf</li><li>/etc/usermap.cfg</li><li>routed status</li><li>smb_hist</li></ul> | <ul style="list-style-type: none"><li>snap delta</li><li>snap reserve</li><li>snap sched</li><li>snapmirror destinations -s</li><li>snapmirror status</li><li>snapmirror status -l</li><li>snmp</li><li>storage alias</li><li>storage show</li><li>storage show disk -p</li><li>storage show mc</li><li>storage show tape</li><li>storage stats tape</li><li>storage stats tape zero</li><li>sysconfig -av</li><li>sysconfig -r</li><li>timezone</li><li>uptime</li><li>version</li><li>vfiler status -a</li><li>vif status</li><li>vlan stat</li><li>vol media_scrub status -v</li><li>vol scrub status -v</li><li>vol status -v</li><li>vscan</li><li>wall cpstats</li></ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



# getPerfSection.pl : Example output

```
$./getPerfSection.pl -f 3122366.perfstat.out -s "stats stop -I perfstat_wafl" | more
----- Perfstat v7.14 -----
APP_NAME, default, ""
BEGIN, default, "FALSE"
CONF_ONLY, default, "FALSE"
DEBUG, default, "FALSE"
END, default, "FALSE"
FILER_TARGETS, set, "b11-stglfs-100"
DO_HOST, set, "FALSE"
HOST_TARGETS, default, ""
ITERATIONS, set, "96"
ITER_INTERVAL, set, "10"
FORCE_ITERATIONS, default, "FALSE"
FILER_LOGIN, default, "root"
FORCE_OS, default, ""
SSH, default, "FALSE"
SEND_ARGS, default, ""
RAMRUN, default, "FALSE"
APP_PARAM, default, ""
PERF_ONLY, default, "FALSE"
QUIET, default, "FALSE"
ROOT_CMD, default, ""
TIME, set, "5"
PRETEND, default, "FALSE"
LOGS, default, "FALSE"
PROFILES, default, "FALSE"
EXCLUDE, default, "FALSE"
STUTTER_STATIT, default, "TRUE"
MULTI_RSH, default, "FALSE"
MONITOR, default, ""

SOURCE_HOST_NAME, stagedbiha
SOURCE_HOST_OS, SunOS
COMMAND_LINE, "-f b11-stglfs-100 -F -t 5 -i 96,10"

SAMPLES, 29
SAMPLE_INTERVAL, 10

FILEROS, b11-stglfs-100, ONTAP7.2.3

===== BEGIN Iteration 1 ===== 16:26:39

===== PERF b11-stglfs-100 POSTSTATS ===== stats stop -I perfstat_wafl
TIME: 16:32:17
TIME_DELTA: 5:2 (302s)
wafl:wafl:name_cache_hit:88/s
wafl:wafl:name_cache_miss:152/s
```



## **All of the prep work is done, and you find that a Perfstat is necessary.**

- Perfstat is used to collect iterative data. Please make sure to collect 3 or more iterations.
- We collect perfstats when we need to see a point in time holistic view of the filer.
- Collecting perfstat does not preclude the case owner from doing the preparatory work described in the earlier sections.
- Perfstat requires certain conditions:
  - Rsh or ssh to be present on the client and enabled and accessible on the filer.



# What is Perfstat?

- Perfstat is a utility that collects performance statistics in real-time from filer(s) and host(s).

Collects PRESTAT (mostly configuration stuff) and POSTSTAT (mostly the performance stuff) data for host and filer

**Must be collected during time of issue!**

However, sometimes it's beneficial to get a "baseline" perfstat for comparison.

Is a simple ASCII text file, readable by anyone (though not organized conveniently)

Has the ability to collect Oracle specific data (Statspack)

**perfstat -a oracle -t 4 -i 5 -o runas=oracle,sysdba=true > perfstat.out**



# Its Time To Collect a Perfstat: Four Rules

- First rule of collecting a perfstat:
  - Collect the perfstat when the problem is happening! If the perfstat is multiple iterations, note when the problem occurred.
- Second rule of collecting a perfstat:
  - When possible, use the latest version of perfstat. The UNIX version is preferred.
- Third rule of collecting a perfstat:
  - When collecting a baseline, make sure the load during the perfstat run is representative.
- Fourth rule of collecting a perfstat:
  - Don't collect a perfstat if rule 1 and rule 2 don't apply



# What perfstat to gather?

- If the performance problem is constant and consistent:
  - Run the perfstat from a client experiencing the problem (if possible) and collect client side config.
  - Collect at least five 4 – 7 minute iterations
    - `perfstat -f "filer" -t 4 -i 5`
- If the performance problem is spiky and very high granularity is needed to capture its occurrence:
  - Run the perfstat from a client experiencing the problem (if possible) and collect client side config
  - Collect as many iterations as needed but as few as possible with 1 minute iterations
    - `perfstat -f "filer" -t 1 -i "as few as possible but as many as needed"`
- If the performance problem involves an Oracle Database, collect statspack as well. (Follow this link, follow the rules above)
  - [How to collect statspack with perfstat](#)



# Break

“I am a flightless Bird. Who needs wings when you're cool?”





## Module 8: Looking inside the filer

- By the time that you finish this chapter, you should understand:
  - Statit
  - sysstat
  - ps
  - The basics of Consistency points



# Start with the messages file

- The first order of business is to scan the message file for any entries that could contribute to performance issue's reported by the customer during the time that the problem occurred. These include hardware-related messages that could indicate the following but are not limited to:
  - any problem(s) with the back-end FCAL loops or devices (e.g., disks, shelves, and/or shelf modules)
  - any problem(s) with the filer's memory sub-system (e.g., recurring ECC parity/correction messages)
  - any problem(s) related to the filer's CPU
  - any problem(s) with the on-board or peripheral network interfaces
  - any problem(s) with the on-board or peripheral HBA interfaces



# Hardware Configuration

- The filer's hardware configuration and OS version is neatly summarized under `sysconfig -a`. Aside from the version of Data ONTAP currently installed on the filer, of particular interest are the following:
  - type of filer
  - amount of main memory
  - number of active loops and the current loop speed
  - number of shelves and disks in each loop
  - type of disks, shelf module, and firmware
  - number of HBA's configured and active
  - number of network interfaces configured and active
  - if disks are multi-pathed (i.e., can we see the same disks on more than one FCAL channel)
  
- This can also be used to validate the filer's current configuration (i.e., card/slot placement, supported peripherals, et.al.).
  - [Card placement guide](#)



# ONTAP Configuration

- Check for optimal configuration
  - ONTAP configuration
    - `sysconfig -r`
      - raid group configuration (how disk are organized)
      - raid group status (e.g. reconstructing, ironing, syncing et.al.)
    - `vol status -v` and `aggr status -v`
      - number of volumes to aggregate
      - volume and aggr options (e.g. minra, maxdirsize, guarantee, et.al.)
      - volume and aggr status
      - `df [-h]`, `df -i [-h]`, `df -A [-h]`
- Be on the lookout for full traditional volumes or aggregates



# sysstat\_1sec.out (Workload Overview)

- 'sysstat\_1sec.out' gives a global view of what is happening on the filer. It is recommended that any form of performance analysis start here.
- The data in 'sysstat\_1sec.out' is equivalent to running the command 'sysstat -x 1' (i.e., extended format at 1-second iterations) from the filer's command-line.
- Using vi, search for 'sysstat\_1sec.out\$'

```
===== PERF usspna06 POSTSTATS ===== type ...\usspna06\sysstat_1sec.out
```

```
Wed 09/27/2006 22:13:50.01
```

| CPU | NFS | CIFS | HTTP | Total | Net   | kB/s | Disk | kB/s  | Tape | kB/s  | Cache | Cache | CP   | CP | Disk | DAFS | FCP | iSCSI | FCP  | kB/s |
|-----|-----|------|------|-------|-------|------|------|-------|------|-------|-------|-------|------|----|------|------|-----|-------|------|------|
|     |     |      |      |       | in    | out  | read | write | read | write | age   | hit   | time | ty | util |      |     |       | in   | out  |
| 67% | 0   | 115  | 0    | 398   | 5801  | 120  | 7630 | 20485 | 0    | 0     | 10    | 100%  | 100% | :f | 41%  | 0    | 283 | 0     | 2383 | 370  |
| 73% | 0   | 484  | 0    | 587   | 21132 | 424  | 3375 | 16888 | 0    | 0     | 10    | 100%  | 100% | :v | 21%  | 0    | 103 | 0     | 74   | 196  |
| 17% | 0   | 242  | 0    | 317   | 9563  | 194  | 707  | 110   | 0    | 0     | 10    | 99%   | 18%  | :  | 10%  | 0    | 75  | 0     | 37   | 171  |
| 51% | 0   | 287  | 0    | 412   | 12132 | 246  | 1003 | 0     | 0    | 0     | 10    | 100%  | 0%   | -  | 11%  | 0    | 125 | 0     | 346  | 264  |
| 66% | 0   | 199  | 0    | 385   | 9710  | 195  | 3000 | 0     | 0    | 0     | 10    | 100%  | 0%   | -  | 14%  | 0    | 186 | 0     | 153  | 795  |
| 32% | 0   | 196  | 0    | 285   | 10268 | 203  | 9328 | 11440 | 0    | 0     | 10    | 99%   | 45%  | Fn | 20%  | 0    | 89  | 0     | 198  | 211  |
| 65% | 0   | 116  | 0    | 197   | 4927  | 102  | 7313 | 20620 | 0    | 0     | 10    | 100%  | 100% | :s | 39%  | 0    | 81  | 0     | 135  | 222  |
| 68% | 0   | 203  | 0    | 309   | 8639  | 175  | 5278 | 20901 | 0    | 0     | 10    | 100%  | 100% | :f | 22%  | 0    | 106 | 0     | 262  | 244  |
| 31% | 0   | 171  | 0    | 270   | 6208  | 129  | 6839 | 19996 | 0    | 0     | 10    | 100%  | 100% | :f | 18%  | 0    | 99  | 0     | 134  | 190  |
| 27% | 0   | 222  | 0    | 342   | 10410 | 209  | 5557 | 16556 | 0    | 0     | 10    | 96%   | 100% | :f | 19%  | 0    | 120 | 0     | 130  | 556  |
| 24% | 0   | 136  | 0    | 203   | 6902  | 138  | 6835 | 21968 | 0    | 0     | 10    | 97%   | 100% | :f | 26%  | 0    | 67  | 0     | 45   | 189  |
| 26% | 0   | 197  | 0    | 244   | 9893  | 197  | 5737 | 20653 | 0    | 0     | 10    | 99%   | 100% | :f | 22%  | 0    | 47  | 0     | 134  | 50   |
| 28% | 0   | 288  | 0    | 323   | 12263 | 247  | 5784 | 18272 | 0    | 0     | 10    | 99%   | 100% | :f | 20%  | 0    | 35  | 0     | 21   | 1    |



# Understanding sysstat\_1sec.out

- CPU
  - for multi-CPU systems 100% does not necessarily reflect a CPU bottleneck.
- NFS/CIFS/HTTP/DAFS/FCP/iSCSI
  - number of OPS/sec per protocol
- Total (total number of ops)
  - NFS+CIFS+HTTP+DAFS+FCP+iSCSI



## Understanding sysstat\_1sec.out (cont.)

- Cache age
  - Age of the most recently ejected cache entry expressed in minutes ... it will have an 's' appended for seconds: 55s (bigger is better)
- Cache hit
  - Percentage of reads satisfied via read cache (bigger may better)
    - Cache hit matters less for random workloads.
- Disk util
  - How busy are the disks (potentially should be less than 85%)
    - The more work requested the higher the latencies past a certain point, which is why we state a generic value of 85%.
  - The granularity of sysstat (one sample per second) is very useful in detecting disk bottlenecks which may be too “bursty” for statit to characterize.



## Understanding sysstat\_1sec.out (cont.)

- CP time :
  - percentage of sample period spent processing the CP
- CP type
  - Type of CP (Consistency Point) operation and processing phase
  - Full list of CP types
  - KB Article: [fairly comprehensive list of cp types](#)
- NOTE: A Consistency Point (CP) is the point when we write to disk.



# Understanding statit

- While 'sysstat\_1sec.out' provides a per-second average of disk utilization, 'statit' provides interval-averaged data for individual disks.
- Using vi, search for 'statit\$'

```
disk ut% xfers ureads--chain-usecs writes--chain-usecs cpreads-chain-usecs
/aggr1/plex0/rg0:
2a.16 4 6.82 0.06 1.00 9778 3.66 12.82 1133 3.10 5.50 1911
2a.32 4 6.88 0.06 1.00 36278 3.73 12.61 1132 3.09 5.45 2065
2a.48 44 84.49 76.94 3.57 3557 3.72 7.24 3025 3.83 3.31 4345
2a.17 46 86.47 78.78 3.49 3653 3.87 6.78 3323 3.82 3.34 4196
2a.33 45 87.14 79.02 3.51 3559 4.04 6.64 3341 4.09 3.14 4451
2a.49 45 84.96 77.47 3.53 3594 3.85 7.19 3133 3.64 3.08 4735
2a.18 44 84.65 76.92 3.55 3429 3.89 7.05 3179 3.84 3.24 4213
2a.34 44 84.85 77.43 3.55 3462 3.80 7.38 2993 3.62 3.30 4512
```

- The above snippet leaves out another grouping ... greads:  
greads--chain-usecs gwrites-chain-usecs  
“greads” and “gwrites” are seen when RAID reconstructs are active.



## Understanding statit (cont.)

disk – “name” of the disk

ut % – Disk utilization as an average percentage. Use sysstat for general per second utilization

xfers – total number of disk ops

| disk              | ut% | xfers | ureads | --chain- | usecs | writes | --chain- | usecs | cpreads | --chain- | usecs |
|-------------------|-----|-------|--------|----------|-------|--------|----------|-------|---------|----------|-------|
| /aggr1/plex0/rg0: |     |       |        |          |       |        |          |       |         |          |       |
| 2a.16             | 4   | 6.82  | 0.06   | 1        | 9778  | 3.66   | 12.82    | 1133  | 3.1     | 5.5      | 1911  |
| 2a.32             | 4   | 6.88  | 0.06   | 1        | 36278 | 3.73   | 12.61    | 1132  | 3.09    | 5.45     | 2065  |
| 2a.48             | 44  | 84.49 | 76.94  | 3.57     | 3557  | 3.72   | 7.24     | 3025  | 3.83    | 3.31     | 4345  |
| 2a.17             | 46  | 86.47 | 78.78  | 3.49     | 3653  | 3.87   | 6.78     | 3323  | 3.82    | 3.34     | 4196  |
| 2a.33             | 45  | 87.14 | 79.02  | 3.51     | 3559  | 4.04   | 6.64     | 3341  | 4.09    | 3.14     | 4451  |
| 2a.49             | 45  | 84.96 | 77.47  | 3.53     | 3594  | 3.85   | 7.19     | 3133  | 3.64    | 3.08     | 4735  |
| 2a.18             | 44  | 84.65 | 76.92  | 3.55     | 3429  | 3.89   | 7.05     | 3179  | 3.84    | 3.24     | 4213  |
| 2a.34             | 44  | 84.85 | 77.43  | 3.55     | 3462  | 3.8    | 7.38     | 2993  | 3.62    | 3.3      | 4512  |



## Understanding statit (cont.)

- ureads – number of user data reads
- chain – number of 4KB blocks per disk. The length per disk
- usecs – disk latency measured in microseconds: 1000 usecs = 1 ms - .001 seconds

| disk              | ut% | xfers | ureads | --chain- | usecs | writes | --chain- | usecs | cpreads | --chain- | usecs |
|-------------------|-----|-------|--------|----------|-------|--------|----------|-------|---------|----------|-------|
| /aggr1/plex0/rg0: |     |       |        |          |       |        |          |       |         |          |       |
| 2a.16             | 4   | 6.82  | 0.06   | 1        | 9778  | 3.66   | 12.82    | 1133  | 3.1     | 5.5      | 1911  |
| 2a.32             | 4   | 6.88  | 0.06   | 1        | 36278 | 3.73   | 12.61    | 1132  | 3.09    | 5.45     | 2065  |
| 2a.48             | 44  | 84.49 | 76.94  | 3.57     | 3557  | 3.72   | 7.24     | 3025  | 3.83    | 3.31     | 4345  |
| 2a.17             | 46  | 86.47 | 78.78  | 3.49     | 3653  | 3.87   | 6.78     | 3323  | 3.82    | 3.34     | 4196  |
| 2a.33             | 45  | 87.14 | 79.02  | 3.51     | 3559  | 4.04   | 6.64     | 3341  | 4.09    | 3.14     | 4451  |
| 2a.49             | 45  | 84.96 | 77.47  | 3.53     | 3594  | 3.85   | 7.19     | 3133  | 3.64    | 3.08     | 4735  |
| 2a.18             | 44  | 84.65 | 76.92  | 3.55     | 3429  | 3.89   | 7.05     | 3179  | 3.84    | 3.24     | 4213  |
| 2a.34             | 44  | 84.85 | 77.43  | 3.55     | 3462  | 3.8    | 7.38     | 2993  | 3.62    | 3.3      | 4512  |



## Understanding statit (cont.)

- writes – number of user data writes
- Average round-trip latency = usecs \* chain

| disk              | ut% | xfers | ureads | --chain- | usecs | writes | --chain- | usecs | cpreads | --chain- | usecs |
|-------------------|-----|-------|--------|----------|-------|--------|----------|-------|---------|----------|-------|
| /aggr1/plex0/rg0: |     |       |        |          |       |        |          |       |         |          |       |
| 2a.16             | 4   | 6.82  | 0.06   | 1        | 9778  | 3.66   | 12.82    | 1133  | 3.1     | 5.5      | 1911  |
| 2a.32             | 4   | 6.88  | 0.06   | 1        | 36278 | 3.73   | 12.61    | 1132  | 3.09    | 5.45     | 2065  |
| 2a.48             | 44  | 84.49 | 76.94  | 3.57     | 3557  | 3.72   | 7.24     | 3025  | 3.83    | 3.31     | 4345  |
| 2a.17             | 46  | 86.47 | 78.78  | 3.49     | 3653  | 3.87   | 6.78     | 3323  | 3.82    | 3.34     | 4196  |
| 2a.33             | 45  | 87.14 | 79.02  | 3.51     | 3559  | 4.04   | 6.64     | 3341  | 4.09    | 3.14     | 4451  |
| 2a.49             | 45  | 84.96 | 77.47  | 3.53     | 3594  | 3.85   | 7.19     | 3133  | 3.64    | 3.08     | 4735  |
| 2a.18             | 44  | 84.65 | 76.92  | 3.55     | 3429  | 3.89   | 7.05     | 3179  | 3.84    | 3.24     | 4213  |
| 2a.34             | 44  | 84.85 | 77.43  | 3.55     | 3462  | 3.8    | 7.38     | 2993  | 3.62    | 3.3      | 4512  |



## Understanding statit (cont.)

- cpreads – number of compute parity reads (unrelated to consistency points)
- greads – reads due to RAID reconstruct or media scrubs
- gwrites – writes due to RAID reconstruct

| disk              | ut% | xfers | ... | cpreads | --chain- | usecs | greads | --chain- | usecs | gwrites | --chain- | usecs |
|-------------------|-----|-------|-----|---------|----------|-------|--------|----------|-------|---------|----------|-------|
| /aggr0/plex0/rg0: |     |       | ... |         |          |       |        |          |       |         |          |       |
| 2a.16             | 14  | 26.22 | ... | 4.49    | 4.95     | 848   | 0 .... | .        |       | 0 ....  | .        |       |
| 2a.32             | 17  | 28.14 | ... | 4.55    | 5.07     | 1209  | 0 .... | .        |       | 0 ....  | .        |       |
| 2a.17             | 47  | 64.51 | ... | 3.68    | 4.96     | 1905  | 0 .... | .        |       | 0 ....  | .        |       |
| 2a.33             | 42  | 59.06 | ... | 3.85    | 4.41     | 2239  | 0 .... | .        |       | 0 ....  | .        |       |
| 2a.18             | 42  | 60.6  | ... | 3.56    | 4.4      | 1997  | 0 .... | .        |       | 0 ....  | .        |       |
| 2a.34             | 42  | 57.66 | ... | 3.25    | 4.8      | 1765  | 0 .... | .        |       | 0 ....  | .        |       |
| 2a.19             | 41  | 57.99 | ... | 3.15    | 4.8      | 1764  | 0 .... | .        |       | 0 ....  | .        |       |



# Understanding statit (cont.)

- Example of saturated disks

| disk              | ut% | xfers  | ureads -- | chain - | usecs | writes -- | chain - | usecs | cpreads - | chain - | usecs |
|-------------------|-----|--------|-----------|---------|-------|-----------|---------|-------|-----------|---------|-------|
| /aggr0/plex0/rg0: |     |        |           |         |       |           |         |       |           |         |       |
| 0a.77             | 1   | 1.03   | 0.1       | 1       | 12500 | 0.52      | 9.3     | 914   | 0.41      | 4.25    | 2176  |
| 0a.64             | 1   | 1.14   | 0.1       | 1       | 44500 | 0.62      | 8.08    | 887   | 0.41      | 4.25    | 2500  |
| 0a.32             | 100 | 233.53 | 232.75    | 1.82    | 18569 | 0.26      | 3.4     | 3471  | 0.52      | 7       | 2543  |
| 0b.49             | 100 | 220.68 | 219.96    | 2.1     | 16485 | 0.26      | 3.4     | 19529 | 0.46      | 7.33    | 4409  |
| 0b.65             | 100 | 210.41 | 209.54    | 1.78    | 20622 | 0.41      | 8       | 4000  | 0.46      | 2.78    | 15680 |
| 0b.45             | 100 | 231.36 | 230.69    | 1.83    | 18582 | 0.41      | 9.38    | 7547  | 0.26      | 4.4     | 6000  |



# Break

“I am a flightless Bird. Who needs wings when you're cool?”





## Module 9: Network

- `'ifconfig -a'` (CONFIG PRESTATS)
  - provides interface configuration information: IP address, MAC address, netmask, mtu, etc.
- `'netstat -s'` (PERF PRESTATS|POSTSTATS)
  - provides all protocol level counters for TCP, UDP, IP, ICMP, and IGMP
- `'netstat -a -n'`
  - provides information about active TCP connections as well as what TCP/UDP ports we are listening on.



## ifstat -a -v

- ifstat counters are zeroed out in PERF/PRESTATS.
  - We do capture ifstat -a -v in PERF/PRESTATS ... so this provides some trending data from “PRESTATS” to “POSTSTATS” in each iteration.
- Definitions about these counters are available in appendix A of the Network Management Guide:
  - [Network Interface Statistics](#)



# Sample ifstat

-- interface e11b (139 days, 2 hours, 3 minutes, 38 seconds) --

## RECEIVE

|                  |        |                  |        |                   |       |
|------------------|--------|------------------|--------|-------------------|-------|
| Frames/second:   | 135    | Bytes/second:    | 9553   | Errors/minute:    | 0     |
| Total frames:    | 33235m | Total bytes:     | 48627g | Total errors:     | 34709 |
| Multi/broadcast: | 0      | No buffers:      | 0      | Non-primary u/c:  | 0     |
| Tag drop:        | 0      | Vlan tag drop:   | 0      | Vlan untag drop:  | 0     |
| CRC errors:      | 21928  | Runt frames:     | 0      | Fragment:         | 1693  |
| Long frames:     | 0      | Jabber:          | 0      | Alignment errors: | 11088 |
| Bus overruns:    | 0      | Queue overflows: | 0      | Xon:              | 0     |
| Xoff:            | 0      | Jumbo:           | 0      | Reset:            | 0     |
| Reset1:          | 0      | Reset2:          | 0      |                   |       |

## TRANSMIT

|                  |        |                   |       |                   |      |
|------------------|--------|-------------------|-------|-------------------|------|
| Frames/second:   | 211    | Bytes/second:     | 291k  | Errors/minute:    | 0    |
| Total frames:    | 16090m | Total bytes:      | 2316g | Total errors:     | 7    |
| Multi/broadcast: | 26889  | Queue overflows:  | 24    | No buffers:       | 0    |
| Max collisions:  | 7      | Single collision: | 1607k | Multi collisions: | 122m |
| Late collisions: | 1959k  | Timeout:          | 0     | Xon:              | 0    |
| Xoff:            | 0      | Jumbo:            | 0     |                   |      |

## LINK\_INFO

|                |      |              |      |              |      |
|----------------|------|--------------|------|--------------|------|
| Current state: | up   | Up to downs: | 16   | Auto:        | off  |
| Speed:         | 100m | Duplex:      | half | Flowcontrol: | none |



# Gigabit and Flowcontrol

- Flowcontrol in the Gig environment is accomplished through the use of pause packets:
  - Xoff: delay sending data
    - Interface is almost out of receive buffers
  - Xon: resume sending data
- Xon/Xoff packets on the “RECEIVE” queue are coming from the switch or a directly connected host.
- Xon/Xoff packets on the “TRANSMIT” queue are coming from the filer.
- NetApp Best Practices recommends Flowcontrol be set to “Full”



## Uh oh...I've got questions

- Well framed issues can be submitted to NetApp
- Include:
  - A framed problem description
  - All relevant data files (i.e., manually-collected datapoints, logs, perfstats, et.al.)
  - Thoughts and observations



# Module 10: Case Studies



# Module 11: Index and Sundry Items

- Various Resources
- NFS Troubleshooting Guide
- Questions?



# Link References

## ■ Perfstat

- <http://now.netapp.com/NOW/download/tools/perfstat/index.shtml>

## ■ Useful Technical Reports

- TR-3537 [High File Count Env Best Practices](#)
- TR-3780 [Reallocate best practices](#)
- TR-3322 [Oracle: Database Performance with NAS: Optimizing Oracle on NFS](#)
- TR-3459 [Flexshare Design and implementation Guide](#)
- TR-3550 [NFSv3 Enhancements in Data OnTap 7.2.1](#)
- TR-3399 [An Introduction to Flexcache Volumes](#)

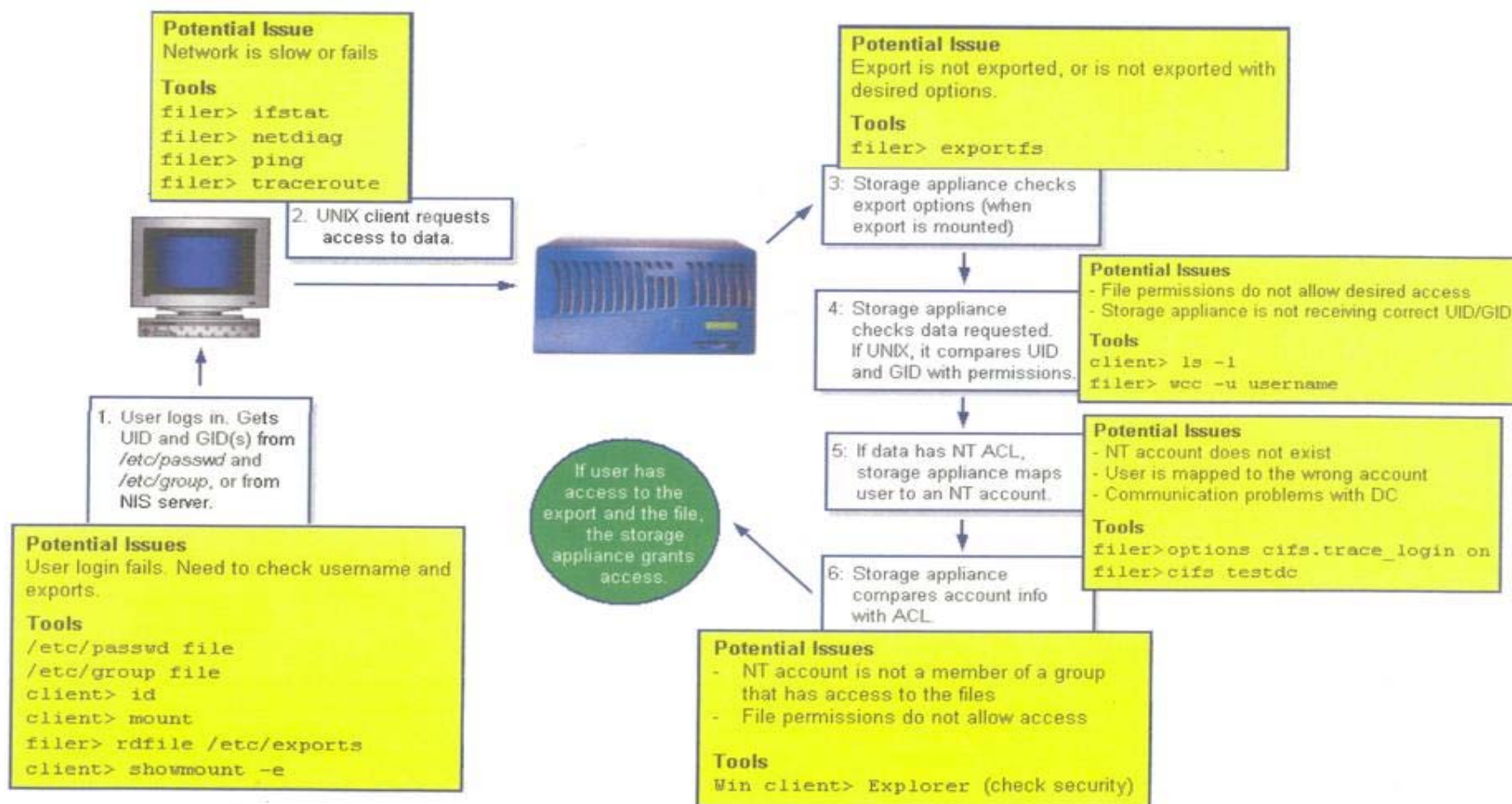
## ■ Useful KBs:

- [How to collect statspack with perfstat](#)
- [Oracle Mount Options](#)
- [Increase maximum number of simultaneous windows requests](#)
- [CIFS access problems apparently due to NIS access](#)
- [fairly comprehensive list of cp types](#)
- [Card placement guide](#)

## ■ Other Guides:

- [Network Interface Statistics](#)

# Troubleshooting NFS Clients



© Copyright Network Appliance, Inc. 2004



# Questions ??

