CDF Run II Analysis System File Server

Stephan Lammel

- **Critical Components**
- **User Home Areas**
- **Network File Server**
- **Network Appliance's Filers**
- **Prototype System**
- **Performance Measurements**
- **Remaining Work**
- **Conclusions**

Critical Components
(deriviation from scaled up Run I)

The Run I analysis systems worked very well for CDF.

We run into a couple of issues though:

- **A heterogeneous approach helped in debugging our software and limited system failures to fractional outages;**
- **We reached the maximum CPU capacity on cdfsga several times limiting our quick upgrade capabilities.**
- **Replicating the software setup of the central analysis systems (on trailer/remote machines) was not convenient and the use of non-central machines limited.**

For Run II we want to have a core of the central analysis system that is extremely robust and thus probably also simple.

We would like to stay heterogeneous, so we can follow mainstream computing trends and get the most analysis power for our buck.

We would like to avoid (as much as possible) upgrade limitations.

User Home Area

The home area of the users takes a central place in the design of our Run II analysis system.

We spend most of our interactive time editing files. Switching to another machine is inconvenient but if our files are inaccessible we cannot do much work.

This means we want to have the user home areas available at any time.

We also want to have good, high speed access to our files. More than two separate home areas per user (some people may argue more than one) becomes painful to manage and does not help in the case the primary area becomes unavailable.

This means we want to put the home areas of the users onto the network where we can guarantee low latency access and sufficient bandwidth.

Options

For CDF's UNIX dominated world, three options come to mind: NFS, AFS, and DFS.

- **We had not the best experience with NFS in Run I. The trailer cluster was slow and the server not always reliable.**
- **AFS is used as clustering technology in the Fermilab general purpose UNIX cluster, FNALU. It requires significant maintenance and introduces third part software that needs to be in synchronization with OS versions. The measured performance is rather low.**
- **DFS is suppose to take over from AFS. However, it is (not yet) wide spread. It suffers from performance problems very similar to AFS.**

Network File Server

Let's analyse our Run I NFS problems in a bit more detail:

On both FNALD and the B0trailer cluster the server process (or whole machine) hung up regularly. The measured performance in the B0trailer cluster was in the order of 200 kBytes/sec. (Direct attached disk in Run I were between 4 and 8 MBytes/sec.)

The problematic component was the server.

Even with a dedicated SGI as server for the B0trailer cluster the situation is not satisfactory.

Since a few years dedicated NFS servers have appeared on the market. Those are small computers (moderate size CPU) that run a reduced operating system, that can only do NFS serving.

The advantage are two fold:

- The reduced OS and NFS oriented tuning allows for a very low latency NFS serving.
- The reduced OS allows such servers to be more robust and reliable.

Network Appliance's Filers

Network Appliance is one of the leading manufacturer of network file servers.

Models range from 1,359 I/O operations per second (with 5 msec responds time) to models with over 7,000 I/O operations per second. For a typical NFS block size of 8 kBytes that translates to around 10 to 50 MBytes/sec.

The filers run a small UNIX like kernel and use a WAFL file system. Disk can be setup in RAID configurations, providing a protection against disk failures. The machines can have several (up to 8 * 4) network interfaces.

In addition to NFS the filers can also serve files based on the HTTP and CIFS protocol (in case we have WindowsNT systems).

The filers can serve both NFS versions, v2 and v3, and using both the TCP or UDP protocol.

All standard UNIX file attribute and protections are supported. Filesystem sizes are not limited to few GBytes. Two or more filers can be clustered to provide additional redundancy. A snapshot feature is provided for quick and convenient backup.

Prototype System

We have acquired the smallest filer model from Network Appliance, a F230.

The filer has

- a 90 MHz CPU,
- 256 MByte of memory,
- 8 MBytes NVRAM,
- 3 disks of 9 GBytes each (7200 RPM), and
- 3 Ethernet 10/100 network interfaces.

The system is located in trailer 165-E (a little green box).

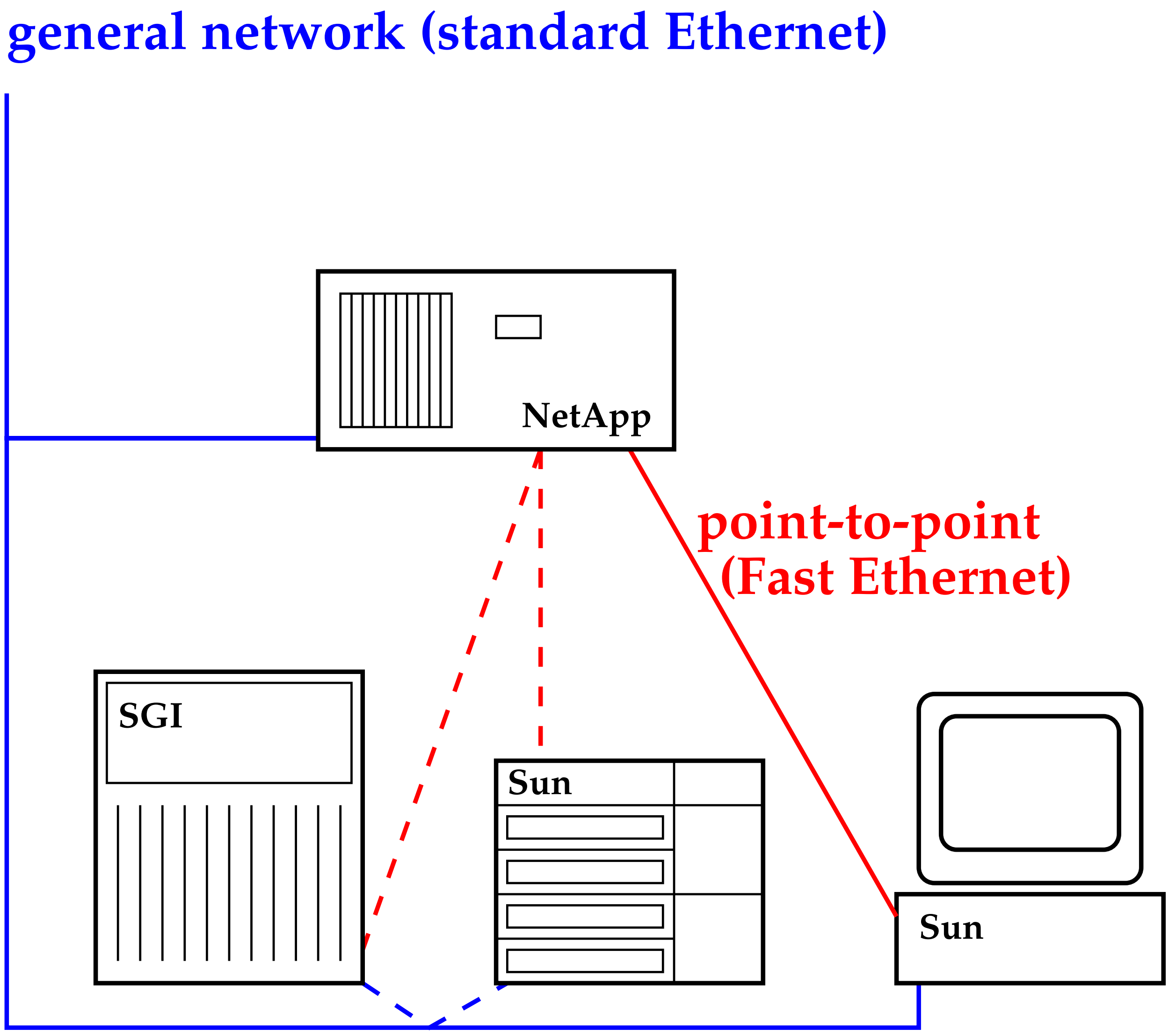
The machine was ordered middle of July and delivered beginning of August.

The basic setup of the system was very easy and straight forward. We were busy with data handling software reviews and let the machine sit idle for a while.

We picked up the evaluation middle of September. When we attempted to setup point-to-point network connection (in addition to a general purpose network connection) we ran into a problem.

We had several phone conferences with Network Appliances to resolve the issue. After several weeks and OS upgrade to the server they have convinced me to take their initial recommendation of separate subnets.

Prototype Filer Setup



Performance Measurements

We have made read and write measurements with the v2 default 8 kBytes and 32 kBytes NFS block size. We have also measured v3 TCP and UDP performance. All on a Silicon Graphics MP and Sun UP and MP.

In summary:

- The measured write performance into a single process (single synchronous write) is 3 MBytes/sec on both SGI and Sun with default NFS settings. Non-standard NFS setting bring no significant advantage but can degrade performance to 2 MBytes/sec.
- The measured read performance into a single process (single synchronous write) is 5 MBytes/sec on both SGI and Sun with default NFS settings. Non-standard NFS setting can bring the rate up to about 6.5 MBytes/sec.

Those measurements were for large sequential reads/writes (1 GByte files), i.e. always to/from disk. The statistics of the filer confirms the no cache usage.

The no cache scenario is the worst case. The filer could be configured with significant memory to cache hot/warm files. Sufficient memory for file attribute caching is very desirable, beyond that we might be better of putting the memory in the relevant analysis machine.

```
##########################################################################

rw_test.csh running on fcdfsgi1 at Fri Nov 27 18:46:51 CST 1998

 2097152 buffer of size 512 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 336.000000 sec: 3.047619 MBytes/sec
2.8u 34.8s 5:36 11% 0+0k 2+65707io 0pf+0w
 1048576 buffer of size 1024 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 339.000000 sec: 3.020649 MBytes/sec
1.4u 23.8s 5:39 7% 0+0k 2+65742io 2pf+0w
 262144 buffer of size 4096 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 342.000000 sec: 2.994152 MBytes/sec
0.4u 15.4s 5:41 4% 0+0k 3+65714io 2pf+0w
 65536 buffer of size 16384 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 340.000000 sec: 3.011765 MBytes/sec
0.1u 11.1s 5:40 3% 0+0k 3+65695io 2pf+0w
 16384 buffer of size 65536 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 340.000000 sec: 3.011765 MBytes/sec
0.0u 10.9s 5:40 3% 0+0k 3+65679io 2pf+0w
 2048 buffer of size 524288 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 341.000000 sec: 3.002933 MBytes/sec
0.0u 11.0s 5:41 3% 0+0k 3+65955io 2pf+0w
total 12571264
drwxr-xr-x    2 lammel   cdf            4096 Nov 27 19:21 ./
drwxr-xr-x    4 lammel   cdf            4096 Nov 21 16:56 ../
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 18:58 tst001k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:04 tst004k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:10 tst016k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:15 tst064k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 18:53 tst512.tmp
-rw-r--r--    1 lammel   cdf      1067769856 Nov 27 19:21 tst512k.tmp
 2097152 buffer of size 512 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 361.000000 sec: 2.836565 MBytes/sec
2.8u 34.9s 6:01 10% 0+0k 1+65665io 0pf+0w
```

```
 1048576 buffer of size 1024 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 339.000000 sec: 3.020649 MBytes/sec
1.4u 23.7s 5:38 7% 0+0k 3+65700io 2pf+0w
 262144 buffer of size 4096 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 339.000000 sec: 3.020649 MBytes/sec
0.4u 15.4s 5:39 4% 0+0k 3+65666io 2pf+0w
 65536 buffer of size 16384 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 340.000000 sec: 3.011765 MBytes/sec
0.1u 11.1s 5:39 3% 0+0k 3+65668io 2pf+0w
 16384 buffer of size 65536 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 339.000000 sec: 3.020649 MBytes/sec
0.0u 10.9s 5:39 3% 0+0k 3+65686io 2pf+0w
 2048 buffer of size 524288 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 340.000000 sec: 3.011765 MBytes/sec
0.0u 11.0s 5:40 3% 0+0k 3+65932io 2pf+0w
total 12571264
drwxr-xr-x    2 lammel   cdf            4096 Nov 27 19:55 ./
drwxr-xr-x    4 lammel   cdf            4096 Nov 21 16:56 ../
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:33 tst001k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:38 tst004k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:44 tst016k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:50 tst064k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 19:27 tst512.tmp
-rw-r--r--    1 lammel   cdf      1067769856 Nov 27 19:55 tst512k.tmp
 2097152 buffer of size 512 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 359.000000 sec: 2.852368 MBytes/sec
2.8u 34.8s 5:59 10% 0+0k 1+65686io 0pf+0w
 1048576 buffer of size 1024 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 338.000000 sec: 3.029586 MBytes/sec
1.4u 23.7s 5:38 7% 0+0k 3+65705io 2pf+0w
 262144 buffer of size 4096 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 337.000000 sec: 3.038576 MBytes/sec
0.4u 15.4s 5:38 4% 0+0k 3+65687io 2pf+0w
 65536 buffer of size 16384 written, total of 1024.000000 MBytes
```

```
 1024.000000 MBytes written in 337.000000 sec: 3.038576 MBytes/sec
0.1u 11.1s 5:37 3% 0+0k 3+65692io 2pf+0w
 16384 buffer of size 65536 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 340.000000 sec: 3.011765 MBytes/sec
0.0u 10.9s 5:39 3% 0+0k 3+65666io 2pf+0w
 2048 buffer of size 524288 written, total of 1024.000000 MBytes
 1024.000000 MBytes written in 337.000000 sec: 3.038576 MBytes/sec
0.0u 11.2s 5:39 3% 0+0k 3+65918io 2pf+0w
total 12571264
drwxr-xr-x    2 lammel   cdf            4096 Nov 27 20:30 ./
drwxr-xr-x    4 lammel   cdf            4096 Nov 21 16:56 ../
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 20:07 tst001k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 20:13 tst004k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 20:19 tst016k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 20:24 tst064k.tmp
-rw-r--r--    1 lammel   cdf      1073741824 Nov 27 20:02 tst512.tmp
-rw-r--r--    1 lammel   cdf      1067769856 Nov 27 20:30 tst512k.tmp
 2097152 buffer of size 512 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 179.000000 sec: 5.720670 MBytes/sec
2.6u 25.0s 2:59 15% 0+0k 65541+101io 0pf+0w
 1048576 buffer of size 1024 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 186.000000 sec: 5.505376 MBytes/sec
1.4u 15.7s 3:06 9% 0+0k 65541+85io 2pf+0w
 262144 buffer of size 4096 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 188.000000 sec: 5.446809 MBytes/sec
0.4u 9.4s 3:08 5% 0+0k 65541+76io 2pf+0w
 65536 buffer of size 16384 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 191.000000 sec: 5.361257 MBytes/sec
0.1u 7.7s 3:11 4% 0+0k 65541+83io 2pf+0w
 16384 buffer of size 65536 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 189.000000 sec: 5.417989 MBytes/sec
0.0u 7.6s 3:09 4% 0+0k 65541+90io 2pf+0w
 2037 buffer of size 524288 read, total of 1018.500000 MBytes
 1018.500000 MBytes read in 184.000000 sec: 5.535326 MBytes/sec
```

```
0.0u 7.7s 3:03 4% 0+0k 65177+92io 2pf+0w
 2097152 buffer of size 512 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 173.000000 sec: 5.919075 MBytes/sec
2.6u 24.7s 2:52 15% 0+0k 65539+77io 0pf+0w
 1048576 buffer of size 1024 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 190.000000 sec: 5.389474 MBytes/sec
1.4u 16.2s 3:10 9% 0+0k 65541+74io 2pf+0w
 262144 buffer of size 4096 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 191.000000 sec: 5.361257 MBytes/sec
0.4u 9.1s 3:11 5% 0+0k 65541+81io 2pf+0w
 65536 buffer of size 16384 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 192.000000 sec: 5.333333 MBytes/sec
0.1u 8.1s 3:11 4% 0+0k 65541+100io 2pf+0w
 16384 buffer of size 65536 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 185.000000 sec: 5.535135 MBytes/sec
0.0u 7.0s 3:05 3% 0+0k 65541+95io 2pf+0w
 2037 buffer of size 524288 read, total of 1018.500000 MBytes
 1018.500000 MBytes read in 184.000000 sec: 5.535326 MBytes/sec
0.0u 7.5s 3:03 4% 0+0k 65177+89io 2pf+0w
 2097152 buffer of size 512 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 173.000000 sec: 5.919075 MBytes/sec
2.6u 25.0s 2:53 15% 0+0k 65539+95io 0pf+0w
 1048576 buffer of size 1024 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 192.000000 sec: 5.333333 MBytes/sec
1.4u 15.9s 3:11 9% 0+0k 65541+74io 2pf+0w
 262144 buffer of size 4096 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 188.000000 sec: 5.446809 MBytes/sec
0.4u 9.6s 3:08 5% 0+0k 65541+106io 2pf+0w
 65536 buffer of size 16384 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 199.000000 sec: 5.145729 MBytes/sec
0.1u 7.7s 3:18 3% 0+0k 65541+97io 2pf+0w
 16384 buffer of size 65536 read, total of 1024.000000 MBytes
 1024.000000 MBytes read in 188.000000 sec: 5.446809 MBytes/sec
0.0u 7.7s 3:08 4% 0+0k 65541+84io 2pf+0w
```

```
 2037 buffer of size 524288 read, total of 1018.500000 MBytes
 1018.500000 MBytes read in 183.000000 sec: 5.565574 MBytes/sec
0.0u 7.0s 3:03 3% 0+0k 65177+82io 2pf+0w
0.0u 7.9s 2:54 4% 0+0k 65543+64io 4pf+0w
0.0u 7.2s 3:11 3% 0+0k 65539+105io 1pf+0w
0.0u 7.7s 3:10 4% 0+0k 65539+77io 1pf+0w
0.0u 7.8s 3:13 4% 0+0k 65539+82io 1pf+0w
0.0u 7.4s 3:07 3% 0+0k 65539+81io 1pf+0w
0.0u 7.7s 2:55 4% 0+0k 65175+75io 1pf+0w
```

Remaining Work

We probably want to bang onto the filer from multiple machines simultaneously and measure the total throughput.

We don't expect this to be very different to the vendor quoted NFS spec numbers. Just to make sure...

We are missing a network connection in order to hook up the SGI and Sun at the same time. Should be available within the next weeks.

We still need to decide where to put the CDF software. If we want to put it on a filer, i.e. one copy per flavour, we have to do some link performance degradation measurements.

We want to deploy the evaluation unit to get some more operational experience. Current thinking is to move users of a well used but less critical system, for instance cdfaix users.

Conclusions

- A network file server for the user home areas seems to be a very feasible approach.

- A dedicated NFS file server looks like a good solution.

- Setup and configuration of the prototype machine was very simple and quick.
- The measured read and write performance into a single process meets our Run II needs.
- The total I/O bandwidth remains to be measured but should not be a concern (specs are above our needs).

- The little prototype machine we have is ready for a real task to prove its reliability (no crash/problem so far).
- Other installations (i.e. Jefferson Lab) report high reliability.

- Need to spec out in detail what we need for Run II. If there is no price penalty there is no reason not to take few and large models.