RZ 3578 (# 99588) 02/07/2005
Electrical Engineering 30 pages

Research Report

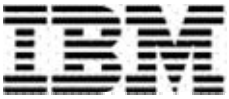
IBM TotalStorage SAN File System, Performance Report: Results of the 6th Alice Data Challenge at CERN

R. Ananthanarayanan,* P.L. Bradshaw,* J. Gomez,* R. Haas,‡ B. Henderson,* C. Silvan,#
C. Tribolet*

*IBM Research ARC

IBM Research IGS

‡IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland



IBM® TotalStorage® SAN File System

Performance Report:

Results of the Simulation Runs for the 6th Alice Data Challenge at CERN

January, 2005

Authors:

Rajagopal Ananthanarayanan,	IBM ARC,	ananthr@us.ibm.com
Paul L. Bradshaw,	IBM ARC,	paulbrad@us.ibm.com
Juan Gomez,	IBM ARC,	juang@us.ibm.com
Robert Haas,	IBM ZRL,	rha@zurich.ibm.com
Bryan Henderson,	IBM ARC,	hbryan@us.ibm.com
Charles Silvan,	IBM IGS,	csn@ch.ibm.com
Chuck Tribolet,	IBM ARC,	triblet@almaden.ibm.com

January 31, 2005.

1. Table of Contents

2. Overview

This report reviews the tested configuration and summarizes the test procedure used to stress the Storage Tank testbed at CERN. In this data challenge, the performance of parallel sequential read and writes has been tested in conditions as close as possible to the 6th ALICE data challenge[1].

Storage Tank performance results of the 5-day and 7-day tests are presented and discussed as well as performance results under various failure scenarios. Based on these results, actions for future work are proposed.

In the test workload, files of size 2GB are created and written by writer threads. Next, these files are read by reader threads and then deleted from the file system. Twelve writer clients are used with four and six writer threads each for the five-day and seven-day test, respectively. Similarly, 12 reader clients execute four reader threads each. All accesses are sequential with an application I/O request size of 1 MB.

[1] http://alice.web.cern.ch and http://aldwww.cern.ch

The storage consists of 15 iSCSI targets that export 53 iSCSI LUNs used as Storage Tank data volumes. Data volumes from each target are combined to form a Storage Pool such that there are 15 pools with a one-to-one mapping between pools and targets. Pools P1 through P8 are hosted by 200i targets. Pools P9 through P15 are hosted by iSCSI targets composed of a mixture of IBM x-series x345 and x335 machines with RAID controllers. A detailed view of the storage setup is provided in Section 4.

15 filesets are defined in the filesystem name-space. A one-to-one policy maps the 15 filesets to the 15 storage pools. As a result I/Os to a file in a particular fileset are isolated to a single storage pool and a single iSCSI target.

The Arlo simulation program (a tool written by IBM Research for the test) is used to drive the workload. Arlo consists of ArloReader, ArloWriter and ArloCop. Upon a request from an ArloWriter, the ArloCop returns the complete path of the next file to be written. The path of the file determines in which of the 15 filesets the file will be created. Upon request from an ArloReader, the ArloCop returns the complete path of the next file to be read. The ArloReader deletes the file once it has read it entirely.

Real-time performance monitoring of Storage Tank, generation of graphs and statistical analysis were done using STFS Perfmon (a tool written by IBM Research for the test).

In summary, there are 15 iSCSI targets, 24 tank clients, and two MDS servers. All machines are IA32 Linux boxes.

2.1. Reading the graphs

All graphs plot Read (R), Write (W) and Read+Write (R+W) throughput in MB/sec on the y-axis and the number of samples along the x-axis. The samples are 10 sec apart. The x-axes on many graphs are time-synchronized, but split across multiple graphs to avoid cluttering. Tags below the main title of the graphs are used to identify the graph with a particular run. Note that the y-axes on the time-synchronized graphs are not all the same because of different ranges of the data. Some graphs plot long-term (LT) averages over a past-period of 100 sec.

3. Summary of five-day and seven-day runs

Below a short summary of the two long runs is presented before complete results are analyzed in subsequent sections. The salient points to note are the following:

1. The five-day run used 130,000 files. A total of 253.9 Terabytes (TB) were written and 253.9 TB read, or a total I/O of 0.496 Petabytes (PB). The average total throughput was 1123.472 MB/sec.
2. The seven-day run used 180,000 files. A total of 351.57 TB were written and 351.57 TB were read, or a total I/O of 0.687 PB. The average total throughput was 1139.421 MB/sec.
3. Combined over a period of 306 h (12 days, 18 h), 1.183 Petabytes of I/O was performed at sustained average of slightly over 1.1 GB/sec.

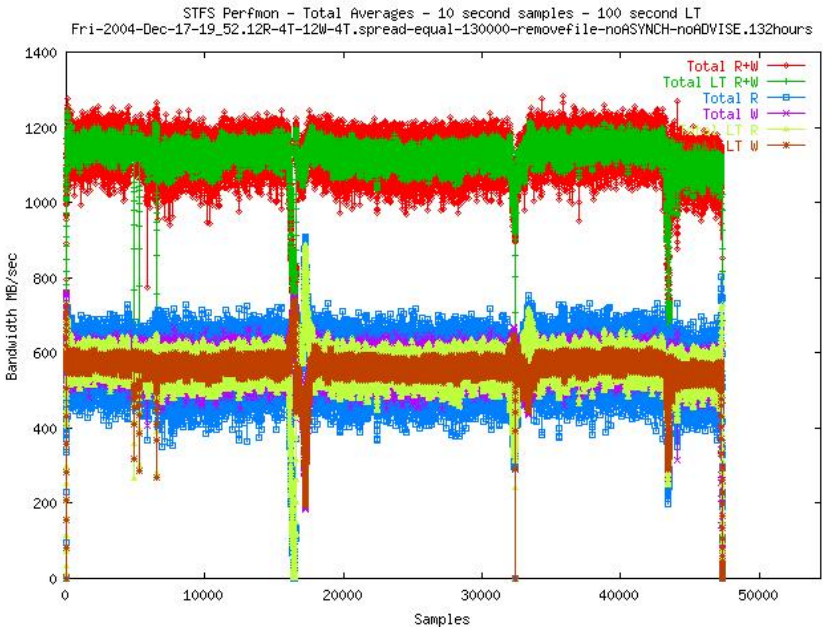


Figure 1: Total average throughput during the five-day run (132 h). The drops in throughput correspond to failures (some intentional, others due to pilot errors) as explained later.

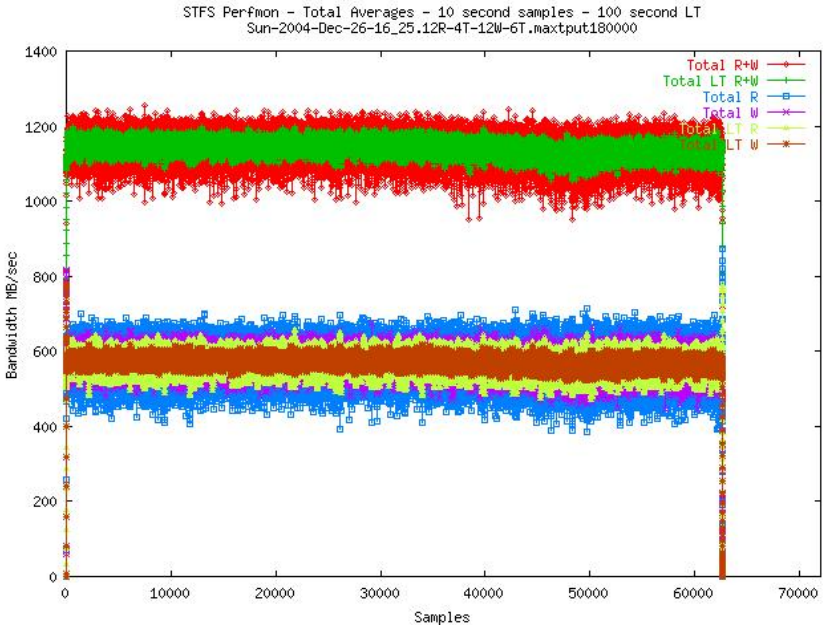


Figure 2: Total average throughput during the seven-day run (174 h). This run was entirely unattended.

4. Testbed Configuration

The system comprises 25 test clients, a management client, 15 storage servers, and a cluster of two MDSes as shown below. This section describes the configuration of all these components and their associated software.

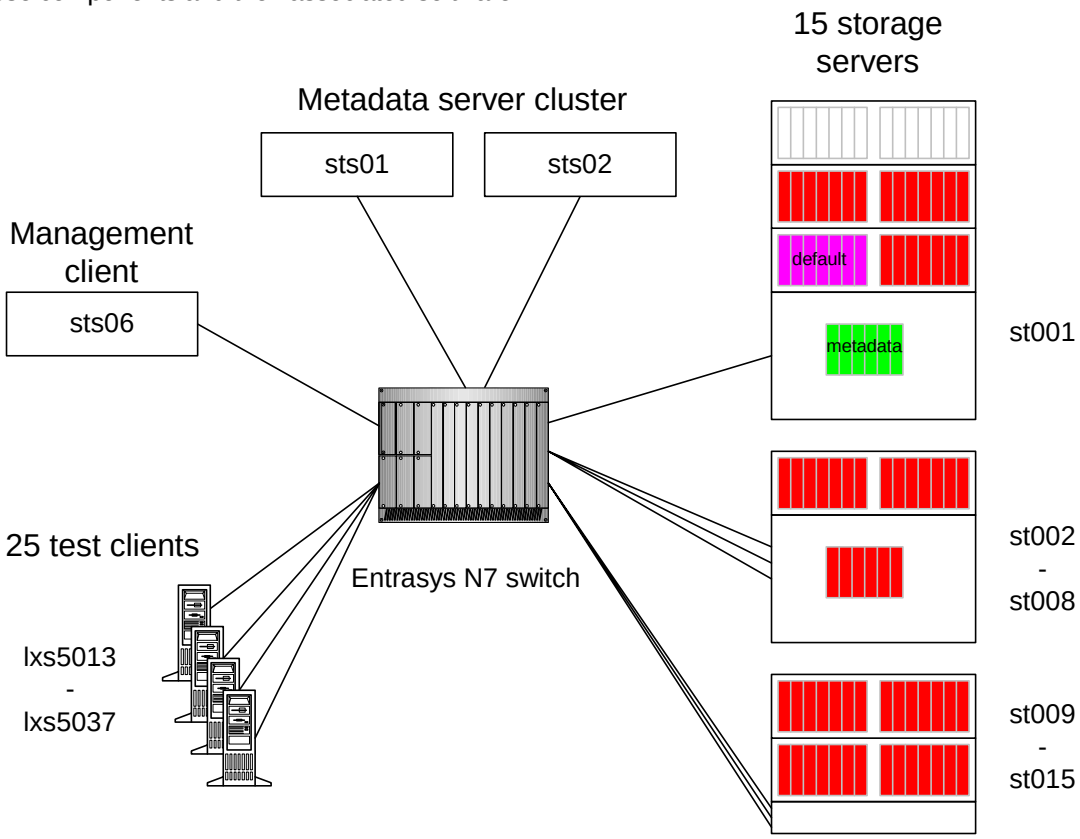


Figure 3: Overview of the Storage Tank testbed.

4.1. Storage-server configuration

The testbed originally comprised eight 200is (st001-st008), each of which was attached over a ServeRAID-4H controller (4 SCSI channels) to three EXP300 expansion boxes with 14 disks each. Seven additional storage servers have been added (st009-st015) with ServeRAID-6m controllers (2 SCSI channels), and some expansion boxes have been relocated to those servers as described in the following.

4.1.1. Target st001

Target st001 is a 200i with the original configuration (six disks in the system unit, and three expansion boxes). iSCSI VLUN 0 is Storage Tank metadata volume. iSCSI VLUN 2 is Storage Tank default data volume. iSCSI VLUNs 5 and 6 are not used. iSCSI VLUNS 2, 3, and 4 are data volumes.

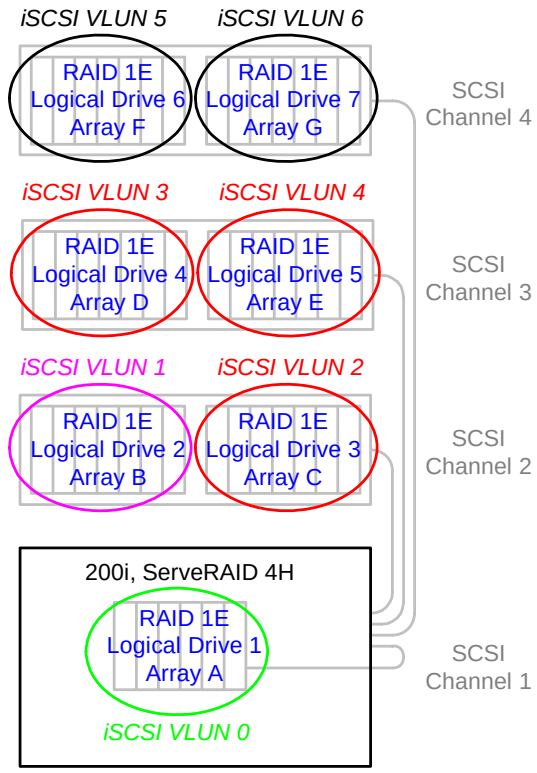


Figure 4: Target st001hardware configuration.

4.1.2. Targets st002 to st008

Targets st002 to st008 are 200is each with a single expansion box used in SCSI split-bus mode. Each target therefore provides three data volumes.

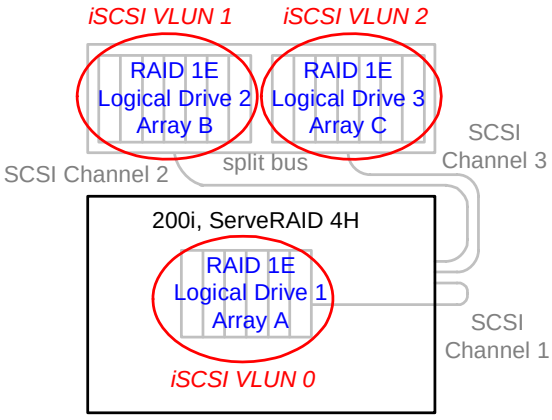


Figure 5: Targets st002 to st008 hardware configuration.

4.1.3. Targets st009 to st015

Targets st009 to st012 are x345s and st013 to st015 are x335s, each with two expansion boxes. SCSI disks in those system units are used for the operating system and are not used by Storage Tank. Each target therefore provides four data volumes.

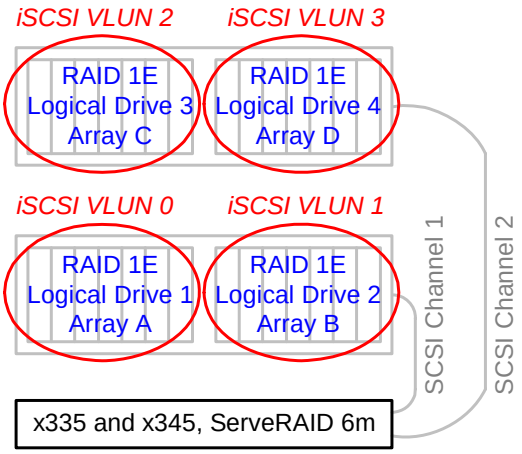


Figure 6 : Targets st009-st015 hardware configuration.

4.1.4. Targets hardware and software configuration

All targets are installed with the Scientific Linux for CERN (SLC) distribution v.3.0.3 and run kernel 2.4.21-20.EL.cernsmp. The only modifications are the enhanced ServeRAID ips driver and the rpm for the iSCSI target code. Screen dumps of the ServeRAID Manager are provided in the Appendix.

Target	CPU	RAM	ServeRAID hardware	ServeRAID driver	iSCSI target code
st001-st008	P-III 1133 MHz Dual or hyperthreaded	2 GB	ServeRAID 4H BIOS + firmware v4.84.1 Stripe-unit size 64kB	ips v 7.10.15 enhanced with *Max Active Commands* = 255 instead of 64	iscsi_tgt-cern-3.0.3-1
st009-st010	Dual Xeon 2.4 GHz	2 GB	ServeRAID 6M BIOS + firmware v6.10.24 Stripe-unit size 64kB	ips v 7.10.15 enhanced with *Max Active Commands* = 252 instead of 64	id.
st011-st012	Dual Xeon 2.4 GHz hyperthreaded	2 GB	id.	id.	id.
st013-st015	Dual Xeon 2.6 GHz hyperthreaded	2 GB	id.	id.	id.

4.2. Test clients configuration

LX-share machines lxs5013-37 have been used for the test. They are dual Xeon 2.4 GHz with 1 GB RAM. They run kernel 2.4.21-20.EL.cernstfssmp.
The following rpms are installed:

- storagetank-client-linux-cern8.7.0-0
- storagetank-cern-client-config-3.0.4-0
- storagetank-perfmon-linux-1.2-0

4.3. Metadata-servers configuration

Machines sts01 and sts02 are used as MDS v2.2. They are dual Xeon 2.6 GHz hyperthreaded with 1 GB RAM. They are installed with CERN's RedHat 7.3 and kernel 2.4.20-30.7.cern255devs (kernel recompiled to support more SCSI devices).

4.4. Management client configuration

Machine sts06 is used as a Storage Tank client (but not involved in the tests), and runs the ArloCop and Perfmon software. It is a dual Xeon 2.6 GHz hyperthreaded with 1 GB RAM. It is installed with SLC 3.0.3 and runs kernel 2.4.21-20.EL.cernstfssmp.
The following rpms are installed:

- storagetank-client-linux-cern8.9.0-0
- storagetank-cern-client-config-3.0.5-0
- storagetank-perfmon-linux-1.2-0

4.5. Storage Tank configuration

Storage pool are created with all data volumes of each target. Given the different number and size of the data volumes in each target, pools have the following sizes:

Pool	Size (in MB)
P1	733,440
P2-P8	698,624
P9-P15	977,920
Total	12,469,248

Pools are configured with the maximum partition size of 256 MB and the maximum allocation size of 256 kB. *Note: as the Storage Tank console (sfscli) does not allow the setup of pools with an allocation size of more than 128 KB, the admsim interface had to be used to create pools with an allocation size of 256 KB*.

For each pool there is a fileset and the corresponding policy rule mapping this fileset to the pool. The 15 filesets are not statically attached to any particular MDS so that the two MDSes automatically take care of seven and eight filesets, respectively.

The sequence of commands necessary to create the configuration is provided in the Appendix.

4.6. Arlo and Perfmon configuration

The ArloCop controls the activity of the ArloReaders and ArloWriters and assigns jobs to the available storage pools. The unit for reading or writing is 1 MB. Files are 2 GB large. The options for ArloReader and ArloWriter were: no ADVISE, no DIRECT_IO, no CACHED_IO, ASYNCH_IO.

As configuration perfmon requires the list of Global IDs of all data volumes and their corresponding pool in order to aggregate the I/O stats from all clients per pool. It also requires the list of clients from which to request I/O statistics.

5. Test Procedure

The logical configuration is described in Figure 7. All Storage Tank clients run a perfmon daemon. The perfmon collector queries all perfmon daemons periodically to compute real-time I/O statistics for each pool. Such statistics can be used by the ArloCop to optimize allocation of files to pools. Each writer machine runs one or more ArloWriter threads. Each reader machine runs one or more ArloReader threads. ArloWriters and ArloReaders obtain the necessary information for their job from the ArloCop. The procedure to start a test is described in the Appendix.

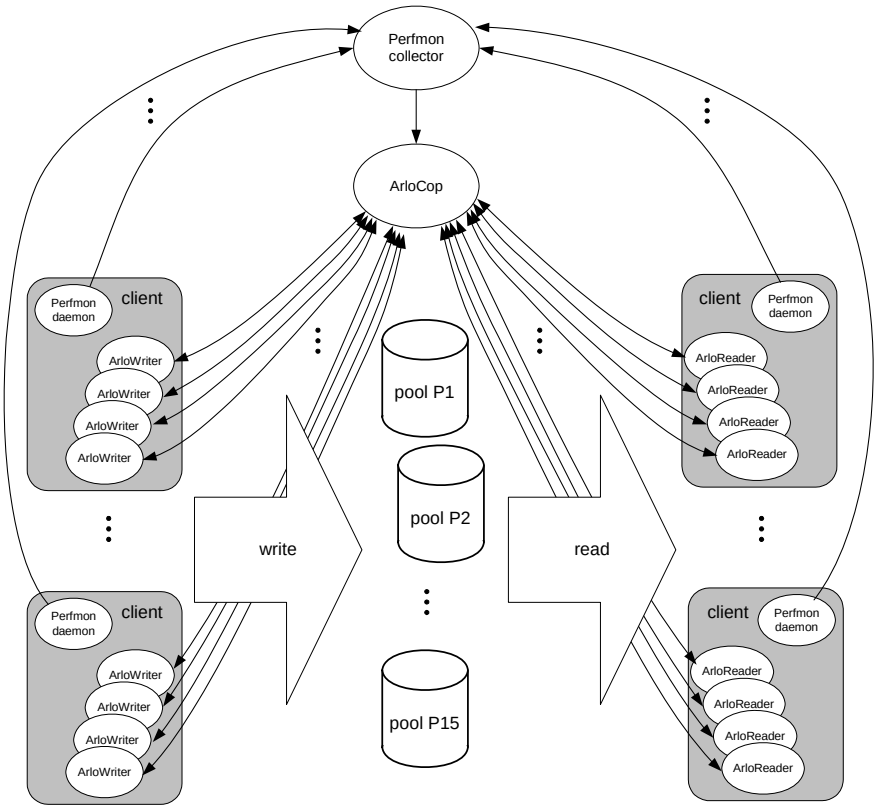


Figure 7: Logical description of the test configuration.

6. Test Results

6.1. Five-day test

The five-day test was executed with the same number of writer and reader threads. As readers tend to progress faster than writers, some reader threads remain idle waiting for new files to be read.

6.1.1. Total throughput over 132 hours

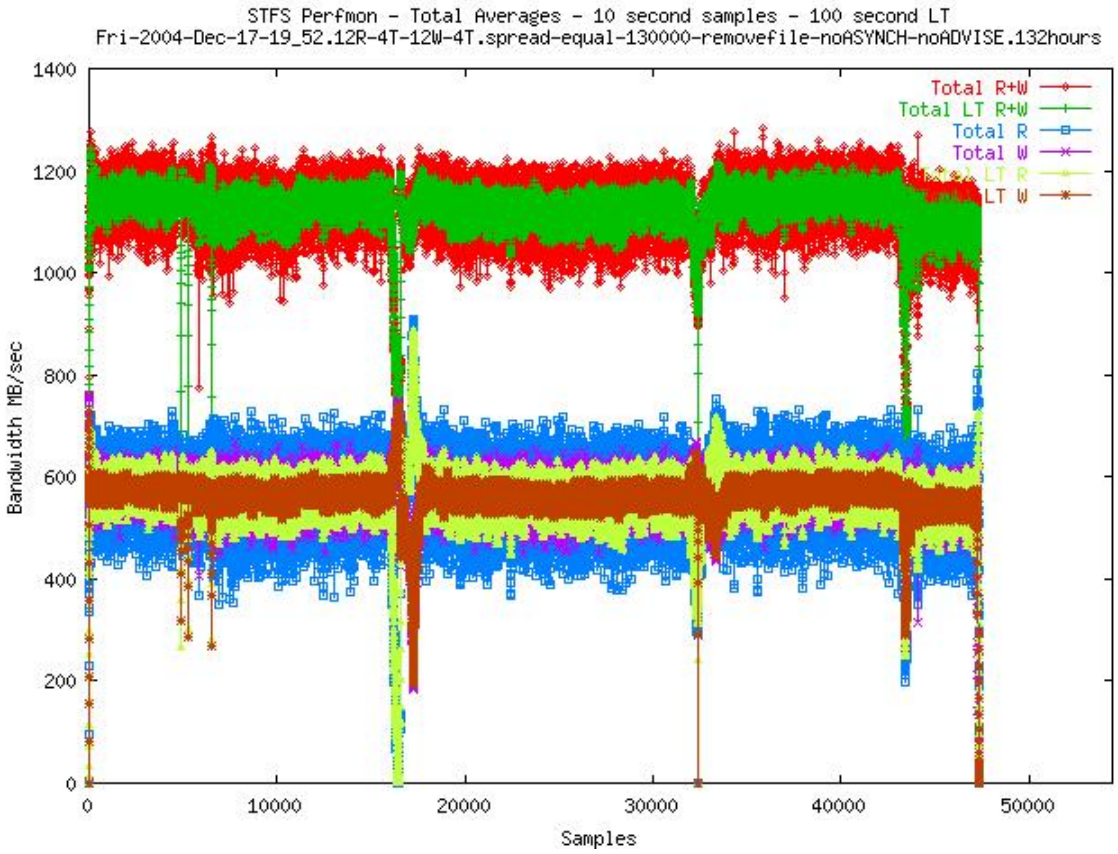


Figure 8 – Total throughput over all 15 pools during the entire 132 h period. Failures were introduced around the 43,000th sample (120 h into the run). The sustained throughput drops after the introduction of the failures, because of removal of two LUNS from service. The earlier dips at samples 16,000 and 32,000 were due to a limit in the Arlo simulation program which unintentionally stopped the reader and writer threads after the processing of only 999 files by each thread (more on this below).

6.1.2. Statistical analysis for 132 hours

Table 1 shows the analysis for the entire 132 h; this includes 120 h of failure-free run (although the stopped reader and writer threads had to be restarted), followed by approximately 12 h of run with introduced failures.

The total average throughput for 132 h was 1123.472 MB/sec, and that of the first 120 h was 1129.124 MB/sec. Part of the reason for the large deviation in individual pools is the various failures as well as the stopping and starting of the threads. These figures can be compared with similar statistics for the seven-day run presented later in Section 6.4 where no thread interruption was encountered.

Table 1: Statistical analysis for 132 h of the five-day run.

Fri-2004-Dec-17-19_52.12R-4T-12W-4T.spread-equal-130000-removefile-noASYNCH-noADVISE.132hours
*** Statistics - Sample Begin: 0, Sample End: 47344 (131.51 Hours) ***

NAME	MAX MB/SEC	MIN MB/SEC	AVG (SD, SD/AVG %) MB/SEC (MB/SEC, PERCENT)
Tot-ReadAverage	906.927	0.000	561.603 (+/- 73.85, 13.15 %)
Tot-WriteAverage	759.705	0.000	561.869 (+/- 43.43, 7.73 %)
Tot-ReadWriteAverage	**1283.614**	**0.000**	**1123.472 (+/- 66.89, 5.95 %)**
P1-256-Read	74.291	0.000	28.429 (+/- 11.73, 41.26 %)
P1-256-Write	50.428	0.000	28.460 (+/- 7.67, 26.94 %)
P1-256-ReadWrite	83.600	0.000	56.889 (+/- 4.80, 8.44 %)
P2-256-Read	76.389	0.000	28.470 (+/- 13.48, 47.34 %)
P2-256-Write	51.112	0.000	28.681 (+/- 8.49, 29.60 %)
P2-256-ReadWrite	105.489	0.000	57.151 (+/- 6.43, 11.25 %)
P3-256-Read	74.700	0.000	28.667 (+/- 12.81, 44.67 %)
P3-256-Write	51.093	0.000	28.632 (+/- 8.41, 29.36 %)
P3-256-ReadWrite	76.506	0.000	57.299 (+/- 6.99, 12.20 %)
P4-256-Read	74.600	0.000	28.820 (+/- 11.57, 40.13 %)
P4-256-Write	51.597	0.000	28.724 (+/- 7.70, 26.80 %)
P4-256-ReadWrite	92.411	0.000	57.543 (+/- 6.47, 11.25 %)
P5-256-Read	74.307	0.000	28.692 (+/- 11.01, 38.37 %)
P5-256-Write	50.500	0.000	28.750 (+/- 6.89, 23.97 %)
P5-256-ReadWrite	74.307	0.000	57.442 (+/- 5.67, 9.86 %)
P6-256-Read	74.500	0.000	28.510 (+/- 12.94, 45.39 %)
P6-256-Write	50.800	0.000	28.595 (+/- 8.15, 28.49 %)
P6-256-ReadWrite	89.617	0.000	57.105 (+/- 6.13, 10.74 %)
P7-256-Read	74.372	0.000	27.853 (+/- 14.93, 53.59 %)
P7-256-Write	50.200	0.000	27.834 (+/- 9.11, 32.73 %)
P7-256-ReadWrite	78.933	0.000	55.687 (+/- 7.49, 13.45 %)
P8-256-Read	74.400	0.000	29.008 (+/- 12.09, 41.67 %)
P8-256-Write	51.400	0.000	28.874 (+/- 7.74, 26.80 %)
P8-256-ReadWrite	74.400	0.000	57.883 (+/- 5.78, 9.99 %)
P9-256-Read	112.295	0.000	49.792 (+/- 20.83, 41.83 %)
P9-256-Write	82.900	0.000	49.900 (+/- 10.99, 22.01 %)
P9-256-ReadWrite	139.399	0.000	99.691 (+/- 15.06, 15.10 %)
P10-256-Read	112.399	0.000	50.622 (+/- 21.21, 41.91 %)
P10-256-Write	83.577	0.000	50.631 (+/- 10.85, 21.42 %)
P10-256-ReadWrite	139.462	0.000	101.253 (+/- 13.94, 13.77 %)
P11-256-Read	112.300	0.000	49.937 (+/- 23.51, 47.07 %)
P11-256-Write	85.486	0.000	50.290 (+/- 12.69, 25.24 %)
P11-256-ReadWrite	137.587	0.000	100.227 (+/- 14.38, 14.34 %)
P12-256-Read	112.400	0.000	49.482 (+/- 23.06, 46.60 %)
P12-256-Write	82.600	0.000	49.625 (+/- 12.95, 26.10 %)
P12-256-ReadWrite	138.500	0.000	99.107 (+/- 17.43, 17.59 %)
P13-256-Read	112.600	0.000	38.641 (+/- 27.06, 70.03 %)
P13-256-Write	79.599	0.000	38.572 (+/- 15.92, 41.28 %)
P13-256-ReadWrite	131.300	0.000	77.213 (+/- 22.03, 28.53 %)
P14-256-Read	124.798	0.000	50.910 (+/- 22.46, 44.12 %)
P14-256-Write	83.443	0.000	50.487 (+/- 12.26, 24.28 %)
P14-256-ReadWrite	149.699	0.000	101.397 (+/- 14.49, 14.29 %)
P15-256-Read	130.154	0.000	43.772 (+/- 30.81, 70.40 %)
P15-256-Write	86.592	0.000	43.814 (+/- 19.91, 45.44 %)
P15-256-ReadWrite	153.138	0.000	87.586 (+/- 16.40, 18.72 %)

6.2. *Failure scenarios after the five-day test*

Several failure "events" were introduced after 120 h of Arlo run, namely:

- Disconnection of iSCSI targets. This was accomplished by removing the network cable connecting the target to the GE switch. After a period of time, the network cable is reconnected. Three targets corresponding to pools P12, 13 and 14 were disconnected in sequence, with an interval of 10 min between each disconnection. All three targets were disconnected for a further 10 min. Then, all three targets were reconnected.

- Removal of LUNS from Pool 9. This was accomplished using the drain-volume administrative command to move "active" files to other LUNs in the same pool. Two LUNS were removed. The drain volume operations took 18 min and 20 min, with about 30 GB of space to be reassigned during each operation.

- Removal of an MDS. This was accomplished by administratively removing one of the two MDS from the cluster (stop server).

6.2.1. Total throughput after failures (120 – 132 hours)

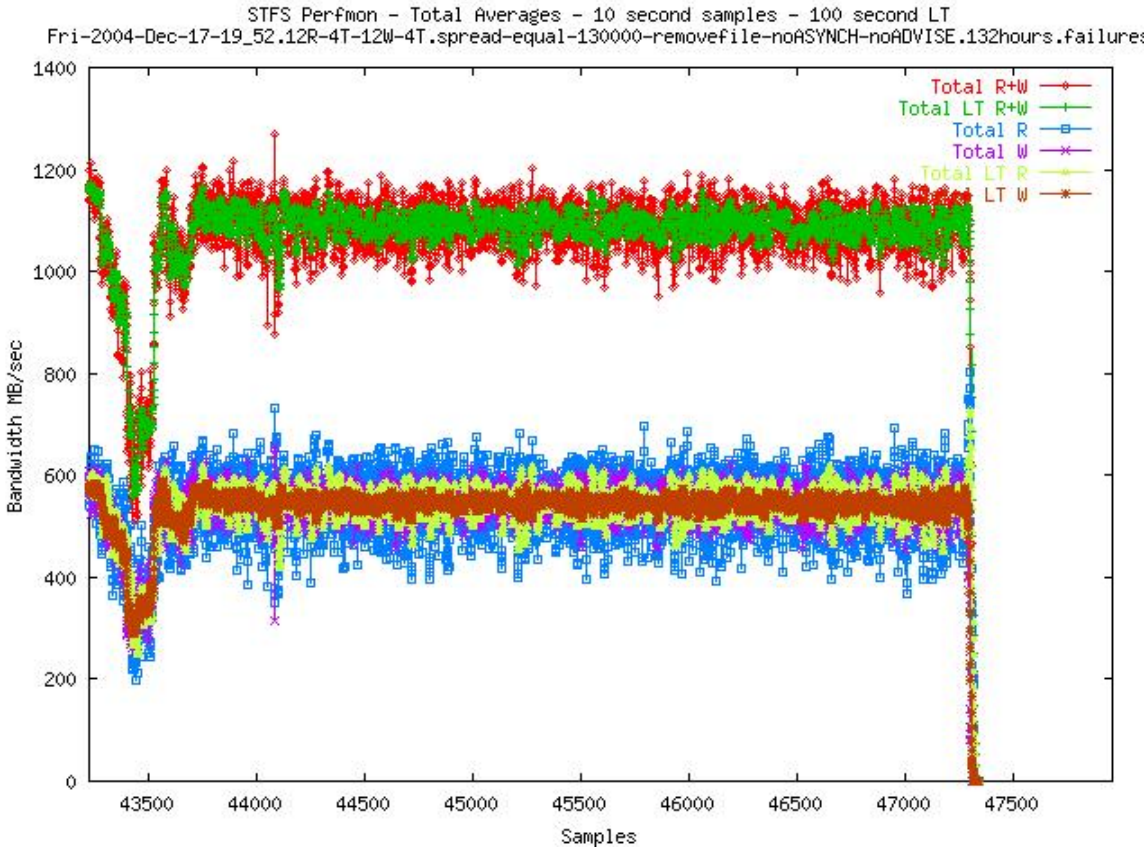


Figure 9: Plot shows total throughput over all 15 pools after failures were introduced. Note that the throughput remains steady after sample 44,500 (123.5 h) and remains stable until the end of the experiment.

6.2.2. Throughput during failure period (120 – 123.5 hours)

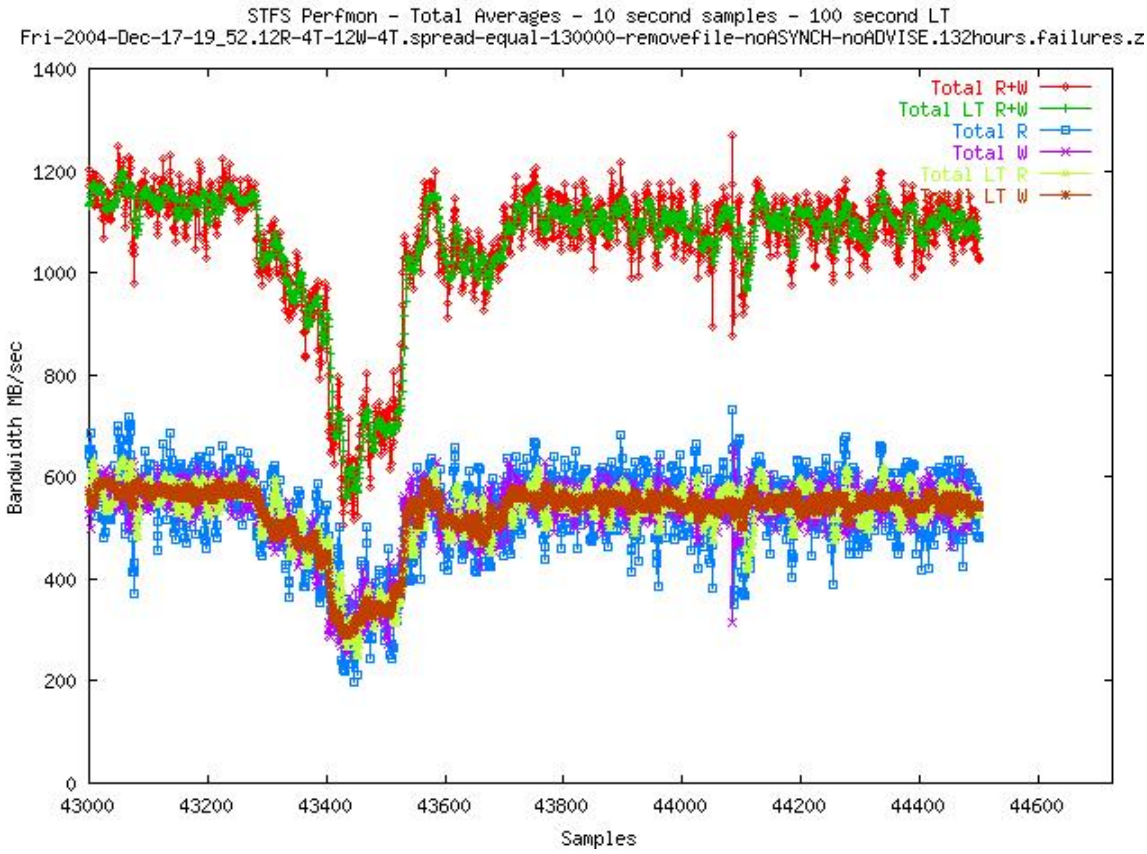


Figure 10: A close-up version of the preceding graph, focusing more on the failure period (120 h – 123.5 h). The drop in throughput between samples 43,300 and 43,600 is due to the successive removal of three different iSCSI targets from the network. The smaller drop in throughput after 43,600 is due to a faulty network cable in one of the targets, as explained in Figure 12. Finally, the fluctuation in throughput at sample 44,100 is due to the removal of one of the MDS: the throughput lowered during cluster reformation, but the effect is almost not noticeable. There were no long-term effects on operating with just a single MDS after sample 44,100 (122.5 h).

6.2.3. Disconnection of iSCSI targets

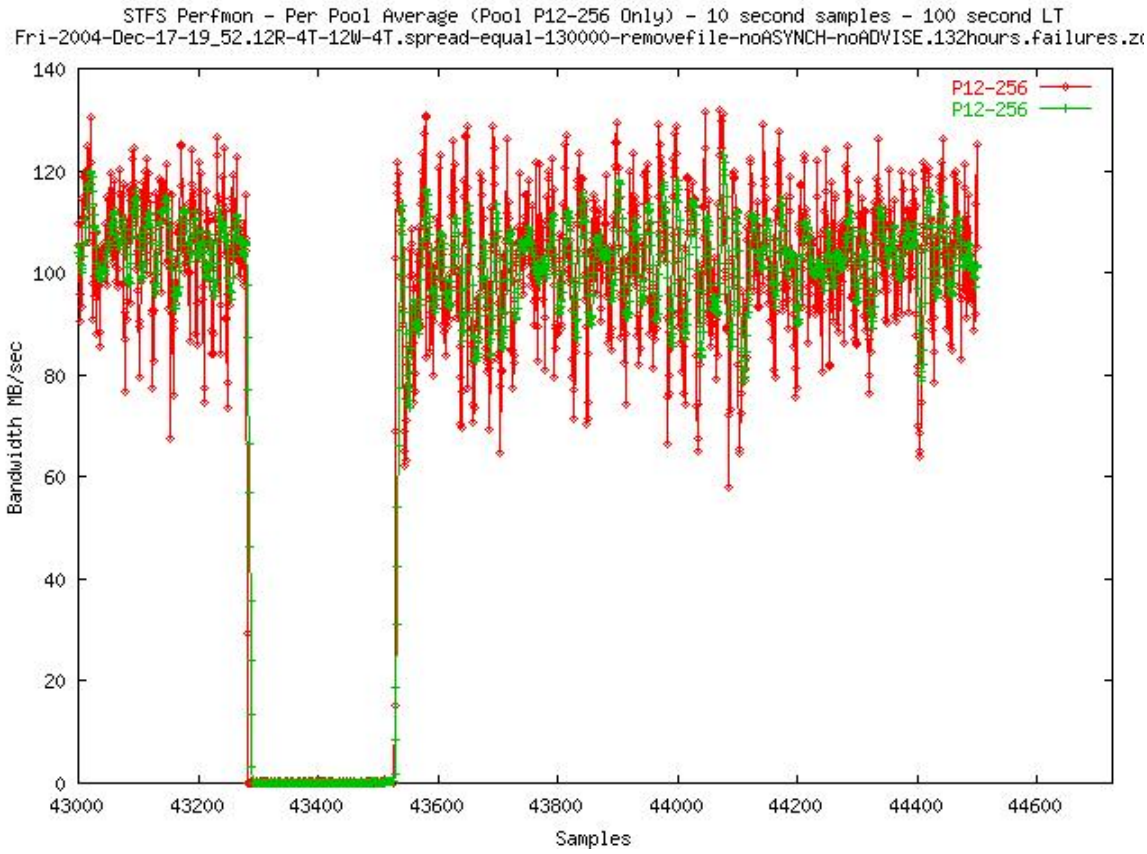


Figure 11: Pool 12 throughput before, during, and after network disconnection. Throughput dropped to zero during the disconnection, and resumed at normal level after reconnection.

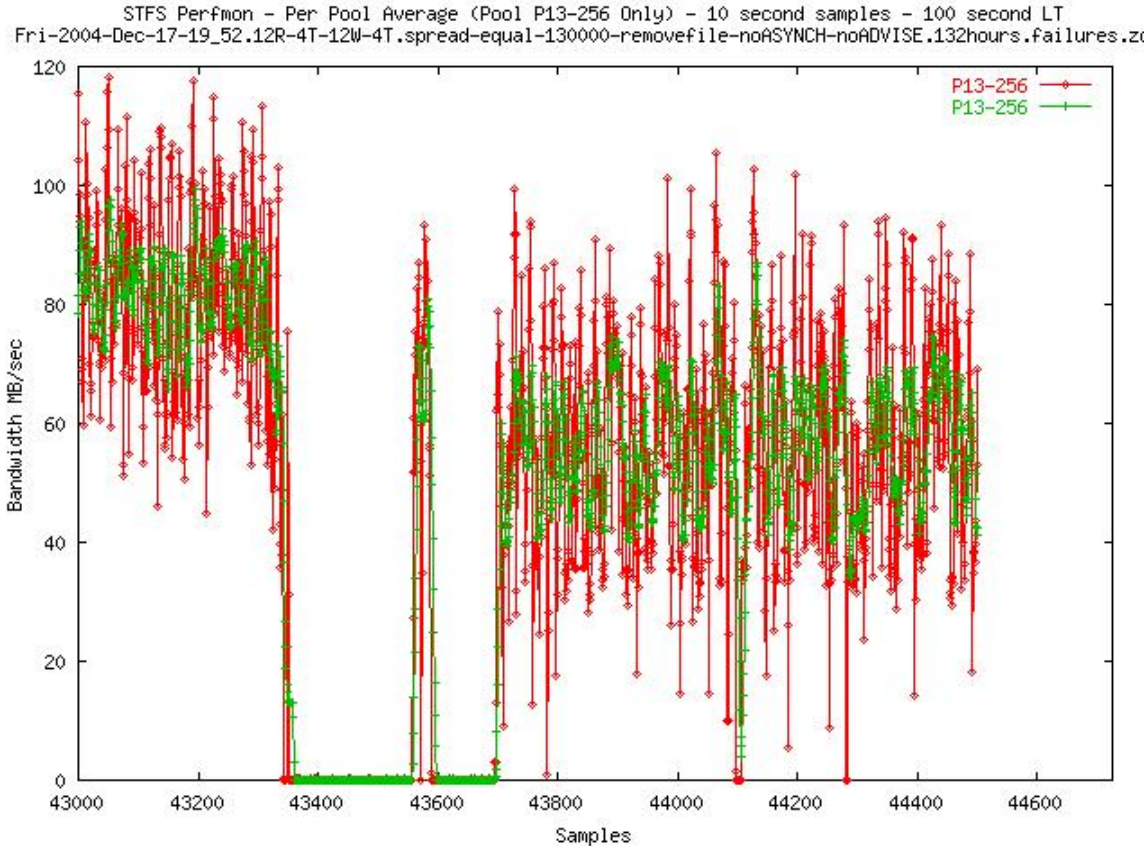


Figure 12: Performance of Pool 13 before, at, and after network disconnection. Throughput dropped to zero during the disconnection, and resumed at normal level after reconnection. In this case, the network cable was found to be faulty therefore the physical connection did not get reestablished properly. This effect is seen around the 43,600th sample when throughput had resumed at normal level, but dropped again to zero soon after. Once this problem had been corrected by the operator, throughput resumed to at a close to normal level (after sample 43,700). During this time, the operator noted a disk failure in this system as well. A raid rebuild was initiated after replacement of the failed disk. This effect can be seen in the generally lowered average throughput (roughly 60 MB/sec) after sample 43,800 compared with the throughput before sample 43,200 (roughly 80 MB/sec).

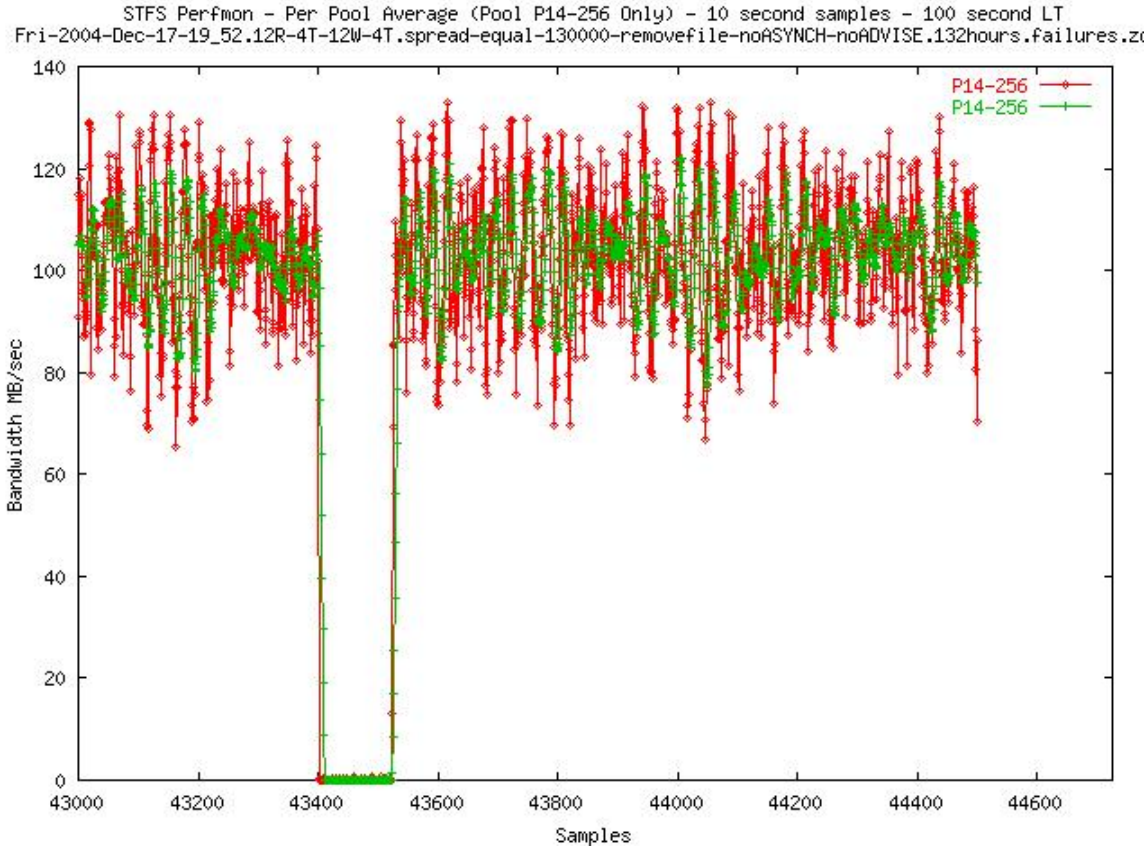


Figure 13: Pool 14 throughput before, during, and after network disconnection. Throughput dropped to zero during the disconnection, and resumed at normal level after reconnection

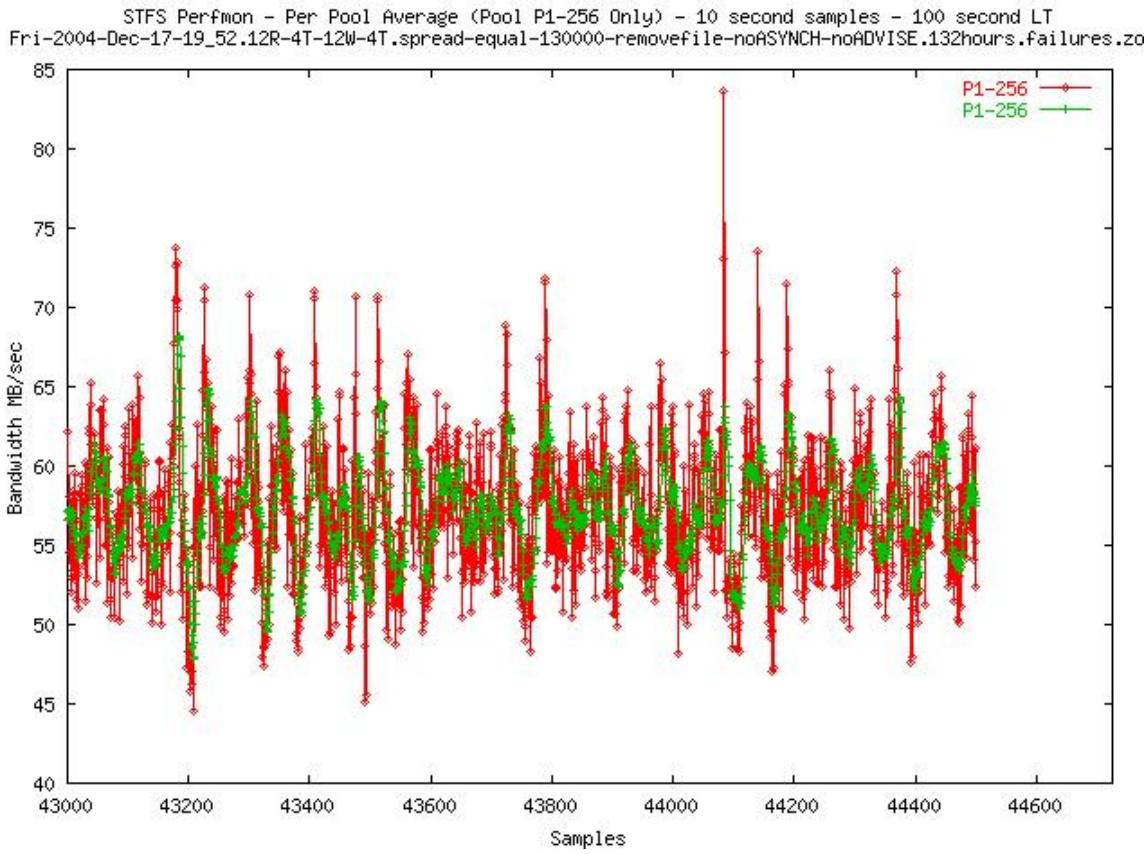


Figure 14: This graphs shows that a pool (P1 in this case) which was not a party to the failures was not affected by events elsewhere: The average throughput of the pool remains steady during the failure period (120 – 123.5 h). This, however, was not universally true. Other pools were affected in minor but noticeable manner, the reasons for which are not known.

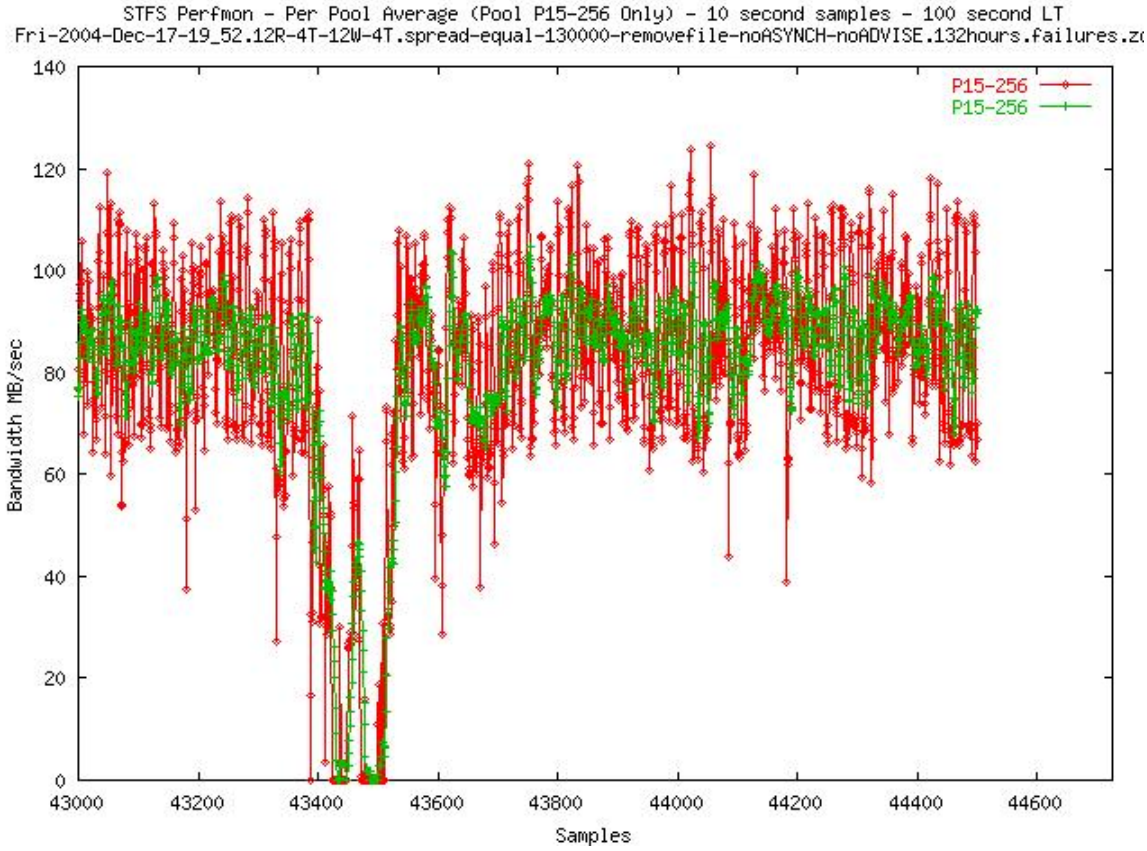


Figure 15: Behavior of Pool 15 during failures. The performance of this pool dipped although no failures were introduced to the corresponding target.

6.2.4. Removal of volumes in pool P9

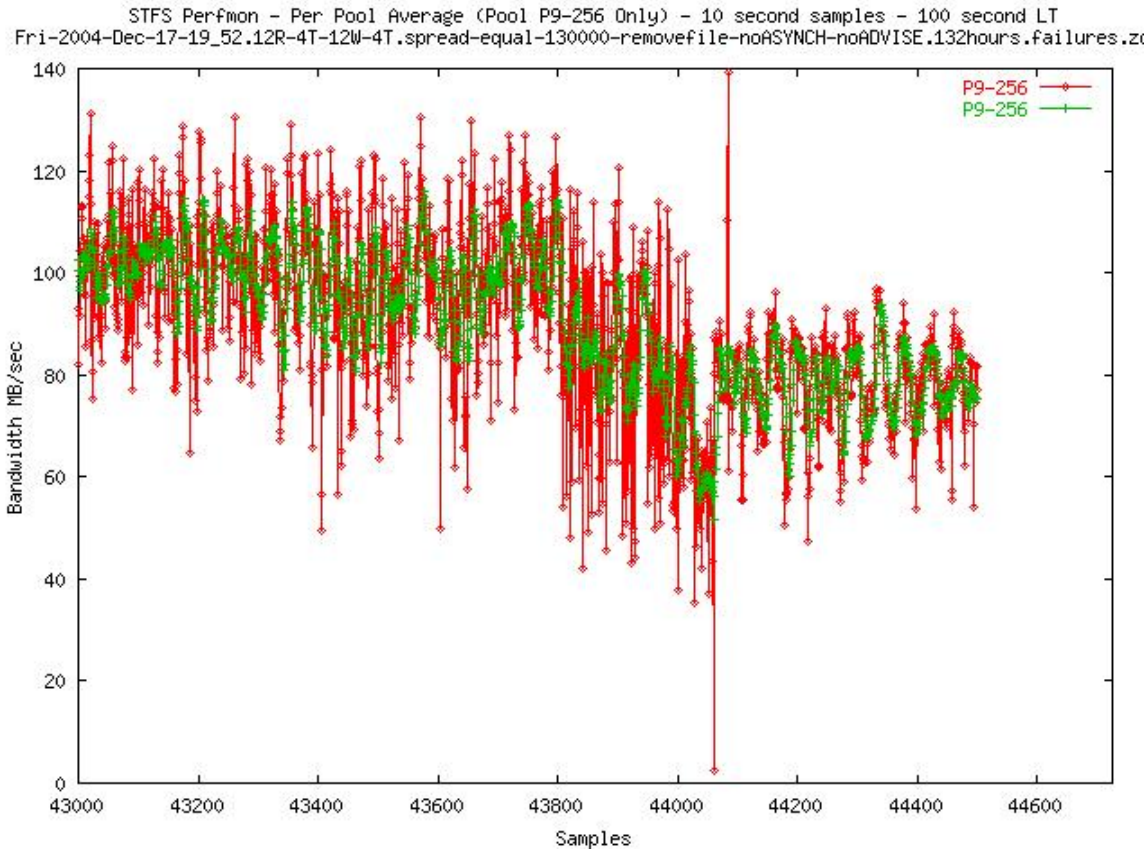


Figure 16: Performance of Pool 9 during failures. Throughput drops correspond to removal of LUNS (drain volume). The first drop at sample 43,800 is smaller when the first volume was removed. The second drop after sample 44,000 is more noticeable when the second volume was removed. Removal of the second LUN effectively made the corresponding external SCSI channel on this target to be useless as no more LUNs remained on that channel. Note also that this target did not participate in earlier failure experiments when other targets were removed from and reconnected to the network: there is no noticeable effect on the throughput of Pool 9 before sample 43,800.

6.3. Effect of premature stoppage of readers and writers during five-day test

In Figure 8, two dips can be seen at sample 16,000 and sample 32,000. These were due to an internal default setting in the Arlo program which led to the unexpected premature completion first of all the reader threads, and later also of the writer threads. The performance of the system during the first set of events at 16,000 is given in Figure 17. The events at 32,000 were due to similar reasons, except that only readers needed to be restarted at 32,000. At sample 17,000, the writers were restarted with an option overriding the default number of files because by then the cause for the stoppage of the readers at 16,000 had been identified.

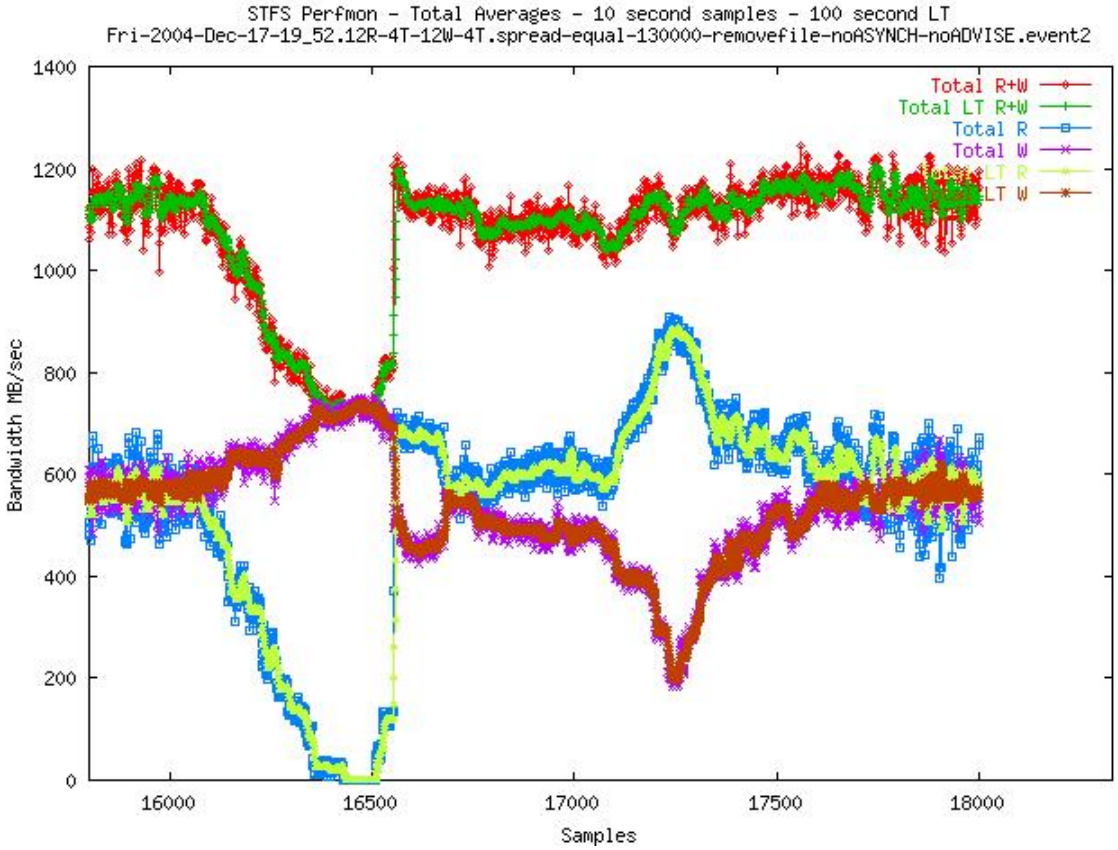


Figure 17: Performance during the first stoppage of readers and writers (samples 16,000 to 18,000). At sample 16,000 (44 h into the run), the reader threads started to complete after processing 999 files each (the default limit in ArloReader). Because different threads complete reading a file at different rates, the reader threads finished at different times. The total read throughput dropped to zero at sample 16,450 (45.6 h into the run). The problem was noticed shortly thereafter at sample16,500. The reader threads were restarted after ascertaining that the read threads had indeed completed. Note that the total throughput never reached zero, but fell below 800 MB/sec around sample 16,500. This is because the writer threads were still active at that time. Further, between samples 16,100 and 16,500, because there were fewer and fewer reader threads to share the bandwidth, writer throughput increased as reader throughput decreased. By the time the readers were restarted, about 1000 files had accumulated in the file system due to continued activity of the writer threads. Normally, reader throughput is limited by the availability of files to be read. Here, for a short period after sample 16,550, the readers did not have to wait for a file to become available for reading. This effect is seen in the above normal total throughput of the readers between samples 16,600 and 16,700; consequently, the writers were slowed down as less bandwidth was available to them. At sample 16,750 the readers and writers achieved balanced throughput. However, at about sample 16,800 the writer threads finished for the same reason as the readers (999 default file limit). Again, the writer threads were restarted. Unlike the reader stoppage event, the system was under observation during the writer stoppage. Because of this the writers were carefully never allowed to finish entirely. At the same time, the total number of writer threads was maintained at or below the total of 48 by the simple procedure of waiting for all four threads in a writer client to complete before substituting four new writer threads in that writer client. This procedure was performed over a period of time, between samples 17,000 and 17,500 (roughly 1.5 h). The reduction in the throughput of the writers and the increase in the throughput of the readers (as explained earlier) can be seen during this time. The reverse is seen as a fresh batch of writer threads (four per client) were introduced, starting at about sample 17,250 and ending at about 17,500. The system achieves balanced throughput again towards the end of the graph.

Neither of the events at 16,000 and 32,000 were intentional, and can be roughly categorized as pilot error: All earlier tries with Arlo at CERN had been for less than 24 h, i.e., not long enough to reach the hard-coded limit of 999 files to process. However, this experience shows that the system can tolerate failures of reader or writer clients in the actual CERN environment, where GDC machines will be the writers and tape servers will be the readers. The events at 16,000 are roughly equivalent to the tape

servers going off-line for a period of more than 1 h. Clearly, the Storage Tank system sustained such interruptions and regained reader/writer balance after a relatively short period of time.

6.4. Seven-day test

In addition to the five-day test, a longer seven-day test was performed from 26th December 2004 to 2nd January, 2005. In this test, the total number of files was 180,000. The number of writer threads per client was increased from 4 to 6. All other parameters were the same as that of the five-day test.

This test was entirely unattended. No components of Storage Tank clients, MDS, iSCSI targets, Arlo, and Perfmon experienced any failures. There were individual disk failures at some iSCSI targets, which affected performance for short durations when RAID rebuild automatically corrected the effects of the failed disks.

6.4.1. Throughput over 174 hours

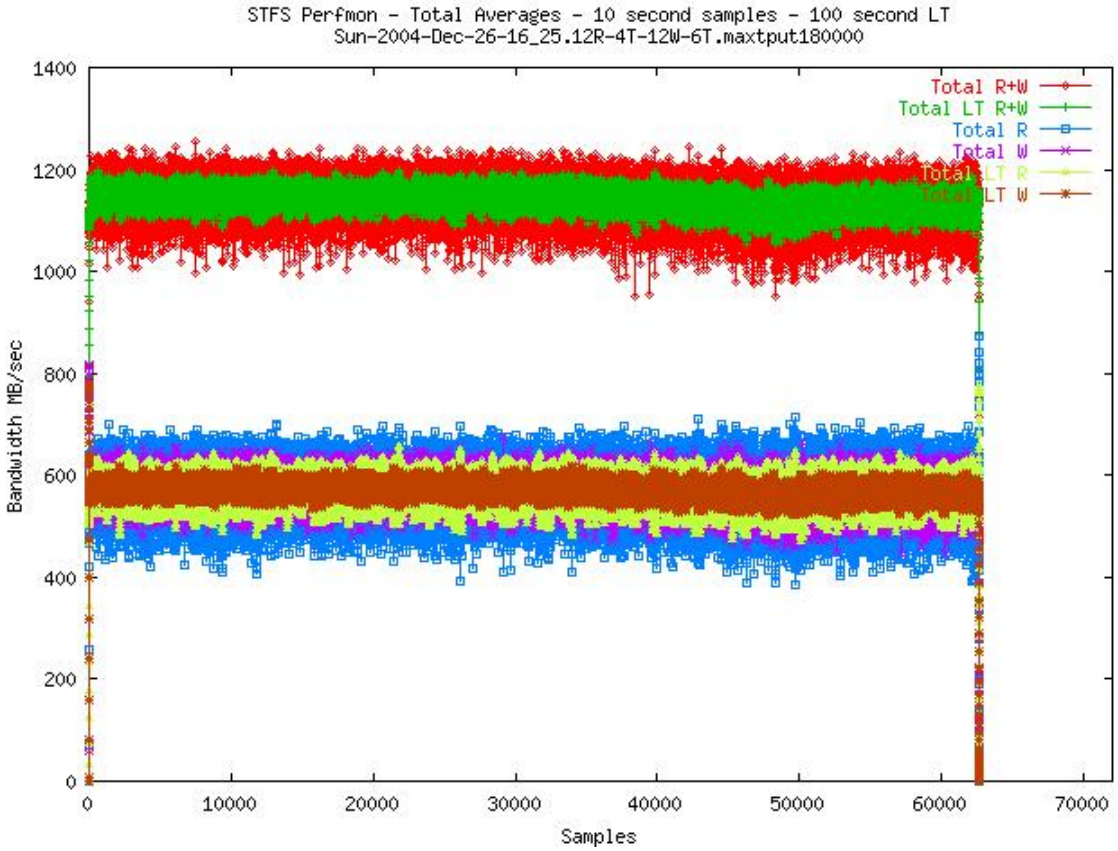


Figure 18: Total throughput across all 15 pools. The graph shows steady performance across 174 h (7 days, 6 h). The slight dip around sample 50,000 (6th day into the run) is due to a disk failure and a subsequent RAID re-build in Pool 15 (more details in Figure 15).

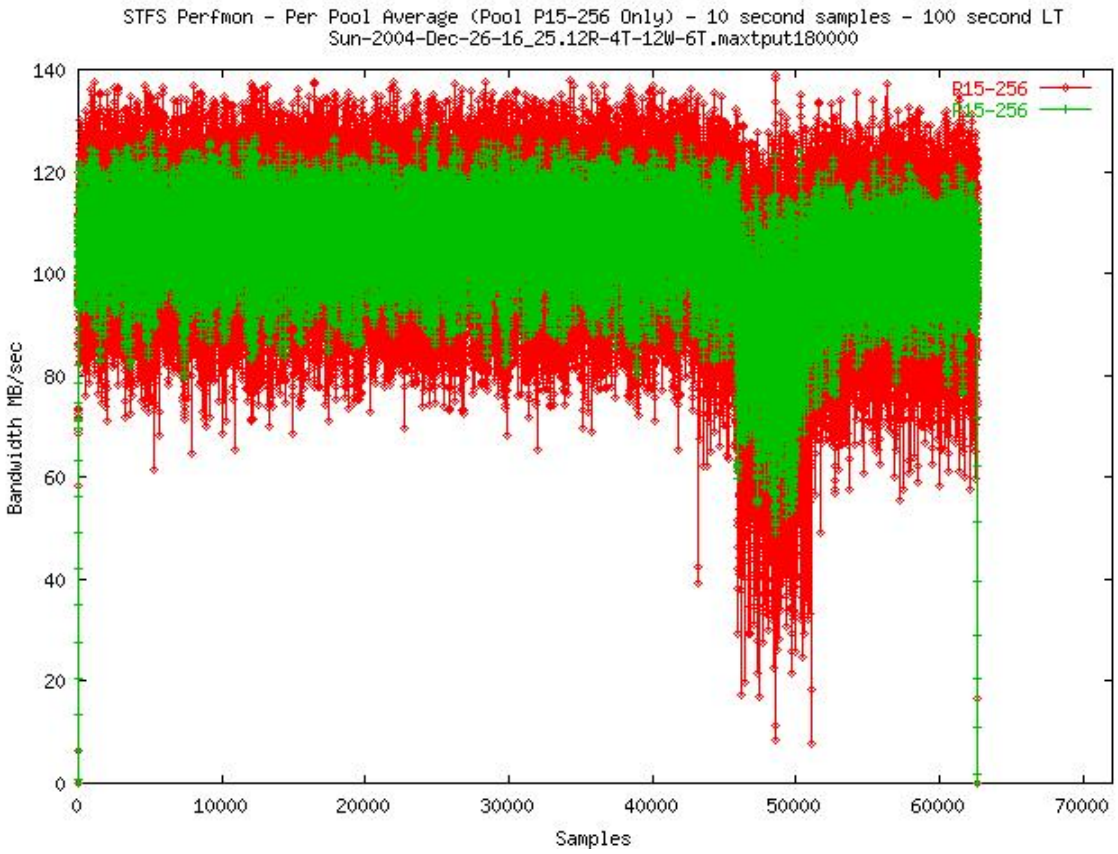


Figure 19: Performance of Pool 15. The effect of a failed disk is seen around sample 50,000 (6th day of the run). The disk was then replaced by the CERN operator.

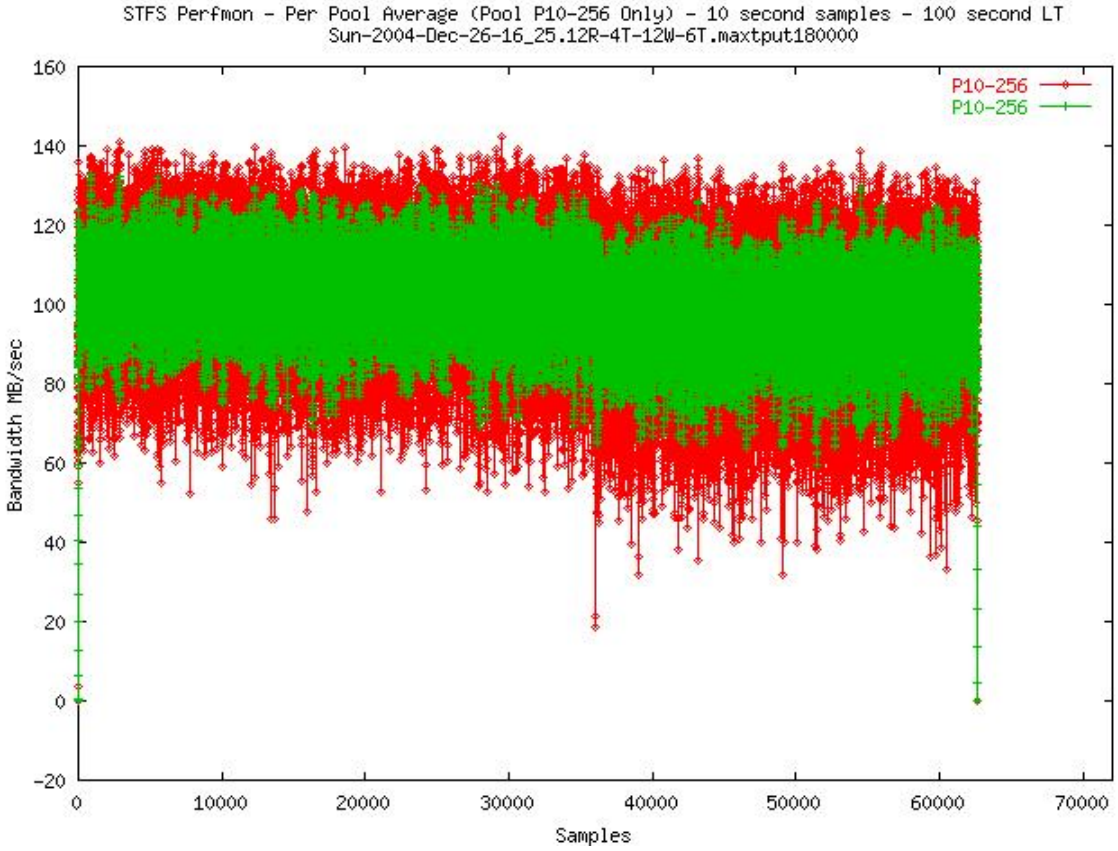


Figure 20: Performance of Pool 10. The corresponding target also had a disk failure. However, the effect of the failure was different from that in Pool 15. Here, the reduction in throughput is much less (starting around sample 35,000 or just after four days of run).

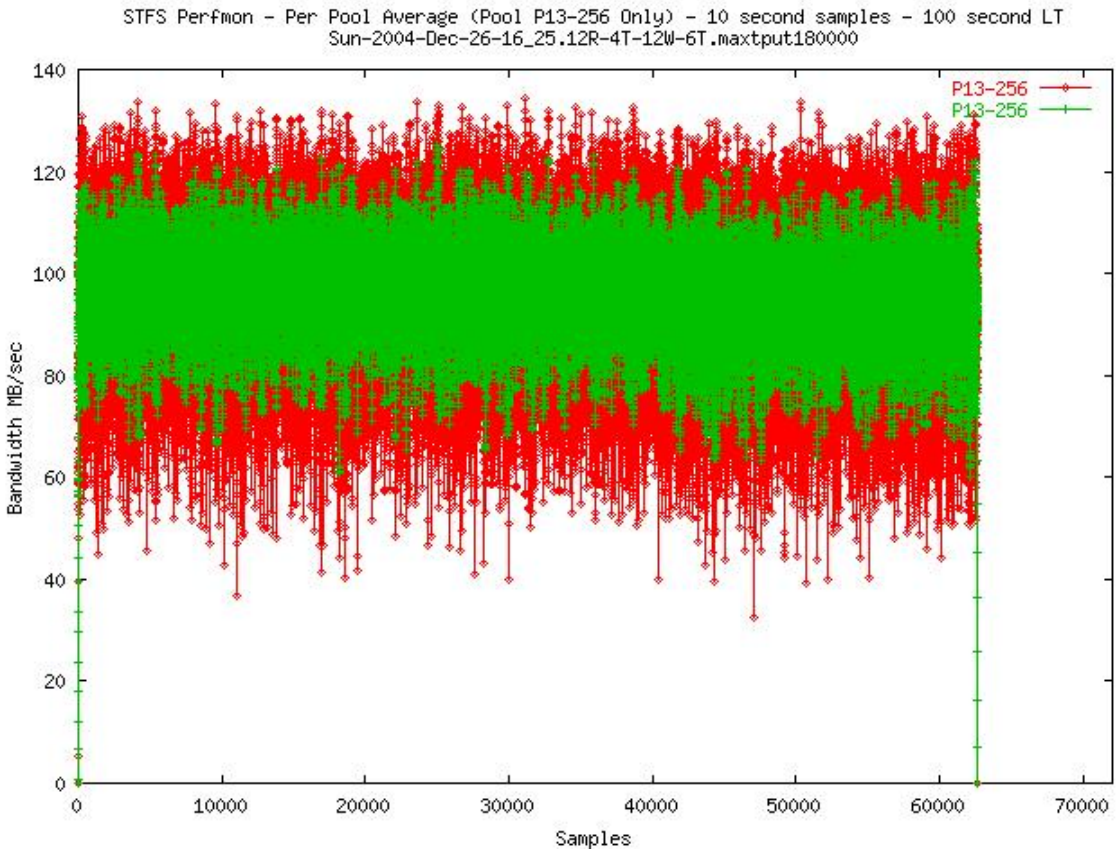


Figure 21: Performance of Pool 13. Another target that had a disk failure. The effect is even more subtle on the total throughput, starting just after sample 40,000. The RAID manager software did not provide a timestamp on the failure of the disk, so it is not clear whether the slight drop at 40,000 is indeed correlated to the disk failure. It is possible that the failure had occurred before the run started, because the average throughput of this particular pool was slightly (5-10%) less than the expected average (see statistics in Table 2).

6.4.2. Statistical analysis for 174 hours

Table 2 shows the statistical analysis of the entire run. Note the following:

1. The end-to-end average total throughput was 1139.421 MB/sec or 1.113 GB/sec with a standard-mean deviation of just 3.77%.
2. The average total throughput from a 200i target was about 58 MB/sec (pools 1 through 7); and that of a 3x5i target was over 100 MB/sec (pools 8 to 15). There were two exceptions:
    a. Pool P9 performed only 63 MB/sec, although it is expected to be above 100 MB/sec. The reason is linked to the earlier failure experiments, in which two LUNS were drained and then added back. After that, the Storage Tank allocation scheme did not use storage space equally on all four volumes.
    b. Pool P13 averaged only 95 MB/sec instead of the expected 100+ MB/sec. The reasons for this are not understood, but this particular target has already shown less smooth performance in earlier trials.

Table 2: Statistical analysis for the seven-day run.

Sun-2004-Dec-26-16_25.12R-4T-12W-6T.maxtput180000

*** Statistics - Sample Begin: 0, Sample End: 62647 (174.02 Hours) ***

NAME	MAX MB/SEC	MIN MB/SEC	AVG (SD, SD/AVG %) MB/SEC (MB/SEC, PERCENT)
Tot-ReadAverage	874.122	0.000	569.710 (+/- 43.38, 7.62 %)
Tot-WriteAverage	816.127	0.000	569.712 (+/- 29.62, 5.20 %)
Tot-ReadWriteAverage	**1254.443**	**0.000**	**1139.421 (+/- 42.99, 3.77 %)**
P1-256-Read	74.192	0.000	28.762 (+/- 9.80, 34.08 %)
P1-256-Write	50.100	0.000	28.762 (+/- 6.39, 22.21 %)
P1-256-ReadWrite	74.192	0.000	57.525 (+/- 4.17, 7.25 %)
P2-256-Read	74.474	0.000	28.834 (+/- 10.30, 35.72 %)
P2-256-Write	51.100	0.000	28.835 (+/- 6.65, 23.07 %)
P2-256-ReadWrite	74.474	0.000	57.669 (+/- 5.37, 9.31 %)
P3-256-Read	74.400	0.000	29.259 (+/- 11.02, 37.67 %)
P3-256-Write	51.105	0.000	29.259 (+/- 7.35, 25.13 %)
P3-256-ReadWrite	74.400	0.000	58.518 (+/- 5.69, 9.72 %)
P4-256-Read	74.353	0.000	28.923 (+/- 10.10, 34.91 %)
P4-256-Write	51.341	0.000	28.923 (+/- 6.91, 23.90 %)
P4-256-ReadWrite	74.353	0.000	57.846 (+/- 6.49, 11.21 %)
P5-256-Read	74.263	0.000	29.122 (+/- 10.15, 34.86 %)
P5-256-Write	50.700	0.000	29.122 (+/- 6.60, 22.65 %)
P5-256-ReadWrite	74.263	0.000	58.244 (+/- 4.78, 8.20 %)
P6-256-Read	73.779	0.000	28.844 (+/- 9.90, 34.32 %)
P6-256-Write	50.721	0.000	28.844 (+/- 6.39, 22.16 %)
P6-256-ReadWrite	73.779	0.000	57.688 (+/- 5.05, 8.75 %)
P7-256-Read	74.000	0.000	28.589 (+/- 10.84, 37.90 %)
P7-256-Write	51.000	0.000	28.589 (+/- 6.76, 23.65 %)
P7-256-ReadWrite	74.000	0.000	57.179 (+/- 5.67, 9.92 %)
P8-256-Read	74.583	0.000	28.426 (+/- 10.18, 35.83 %)
P8-256-Write	51.242	0.000	28.426 (+/- 6.39, 22.49 %)
P8-256-ReadWrite	74.583	0.000	56.852 (+/- 5.91, 10.39 %)
P9-256-Read	97.200	0.000	31.734 (+/- 14.01, 44.15 %)
P9-256-Write	62.200	0.000	31.734 (+/- 10.34, 32.58 %)
P9-256-ReadWrite	**97.800**	**0.000**	**63.468 (+/- 12.09, 19.05 %)**
P10-256-Read	111.600	0.000	50.263 (+/- 22.20, 44.16 %)
P10-256-Write	83.200	0.000	50.264 (+/- 10.65, 21.19 %)
P10-256-ReadWrite	142.400	0.000	100.527 (+/- 15.45, 15.37 %)
P11-256-Read	112.100	0.000	53.165 (+/- 19.63, 36.92 %)
P11-256-Write	85.000	0.000	53.166 (+/- 9.71, 18.26 %)
P11-256-ReadWrite	141.190	0.000	106.331 (+/- 12.71, 11.95 %)
P12-256-Read	110.729	0.000	51.523 (+/- 19.42, 37.70 %)
P12-256-Write	83.258	0.000	51.522 (+/- 9.87, 19.16 %)
P12-256-ReadWrite	138.242	0.000	103.044 (+/- 12.51, 12.14 %)
P13-256-Read	109.384	0.000	47.771 (+/- 19.50, 40.82 %)
P13-256-Write	81.600	0.000	47.770 (+/- 10.24, 21.44 %)
P13-256-ReadWrite	**134.441**	**0.000**	**95.542 (+/- 13.82, 14.47 %)**
P14-256-Read	112.177	0.000	52.365 (+/- 19.32, 36.89 %)
P14-256-Write	83.800	0.000	52.365 (+/- 10.10, 19.29 %)
P14-256-ReadWrite	138.311	0.000	104.730 (+/- 12.26, 11.70 %)
P15-256-Read	112.736	0.000	52.130 (+/- 19.29, 37.01 %)
P15-256-Write	89.326	0.000	52.130 (+/- 11.63, 22.30 %)
P15-256-ReadWrite	138.889	0.000	104.260 (+/- 13.53, 12.98 %)

7. Conclusions

7.1. Storage Tank ready for ALICE 6th data challenge

The two tests have been designed to reproduce as closely as possible the conditions of the 6th ALICE data challenge, i.e., a small number of clients accessing a large amount of files in sequential mode. The results which are about 27% above target (measured average of 1.14 GB/s, target of 900 MB/s) are very encouraging.

In summary, the key enablers for achieving a sustained high throughput are:

- Maximum block size (file system data block size) of 256 kB used for all pools to minimize load on the MDS, and to limit the size of the write-ahead log.
- Maximum pool partition size of 256 MB.
- Vertical pools for each target to ensure homogeneous nominal performance within a pool.
- Enhanced iSCSI target code to increase fairness and performance of flows.
- Spreading the lxs clients, storage servers, and MDSes, over the Gigabit-Ethernet switch to avoid that more than four machines are connected to any group of 10 GE ports on the Entrasys N7 switch.

Some issues remain that could influence the result in the ALICE test scenario:

- Sharing network bandwidth with other traffic. The tests were done in isolation, i.e., with Storage Tank traffic only.
- Sharing CPU on the test clients. We did not measure CPU load in the test clients due to storage.

7.2. Outlook

In addition to high sustained throughput performance, robustness and performance under heterogeneous and/or failure conditions are critical aspects of a successful deployment of Storage Tank in the CERN environment. From a virtualized storage viewpoint, the flexible policy-based allocation of files to storage pools implemented in Storage Tank could clearly benefit from such real-time measurements.

Even though no storage server failed during the five-day and seven-day test (only individual disk failures, none leading to data loss), the configuration with vertical pools was designed to limit damages in the case of target failures (only threads operating on files in these targets/pools would have been affected). Horizontal pools, i.e., pools that span multiple targets, would require at least a similar robustness, for instance by ensuring that entire files end up in a single volume according to policy rules.

8. Appendix

8.1. Storage server detailed RAID configuration

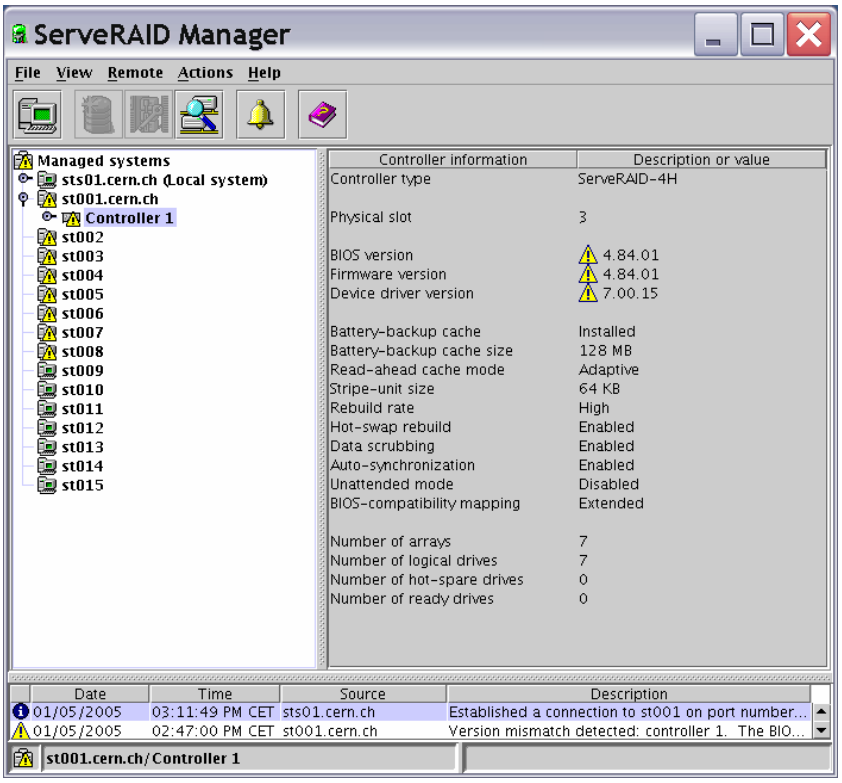


Figure 22: Target st001 RAID configuration (launch root@sts01:/usr/RaidMan/RaidMan.sh)

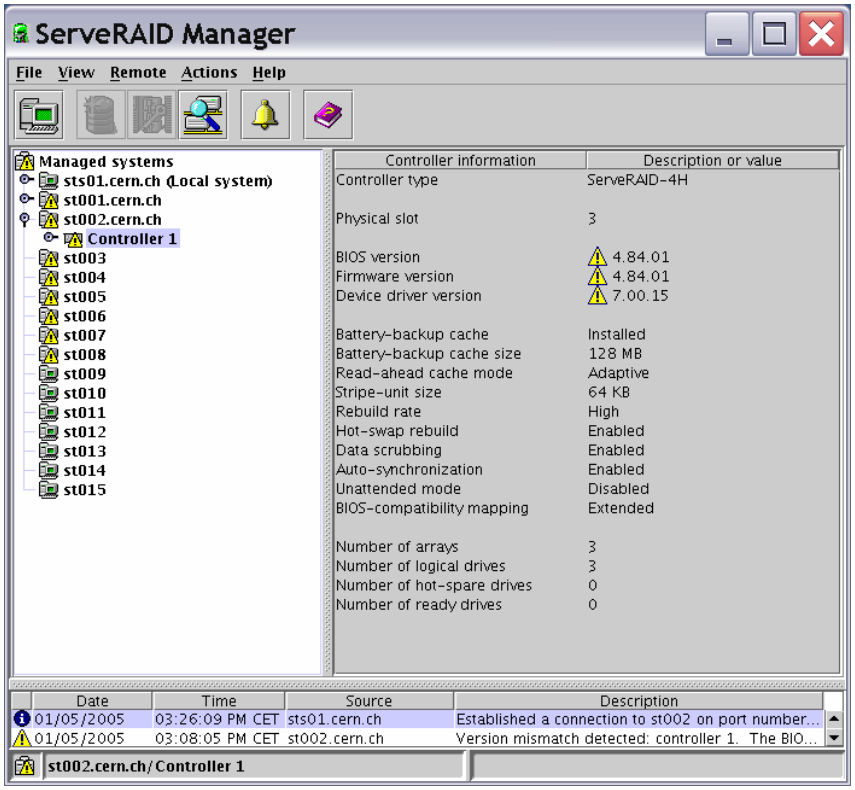


Figure 23: Targets st002-st008 RAID configuration (identical to st001 except for the number of arrays/logical drives)

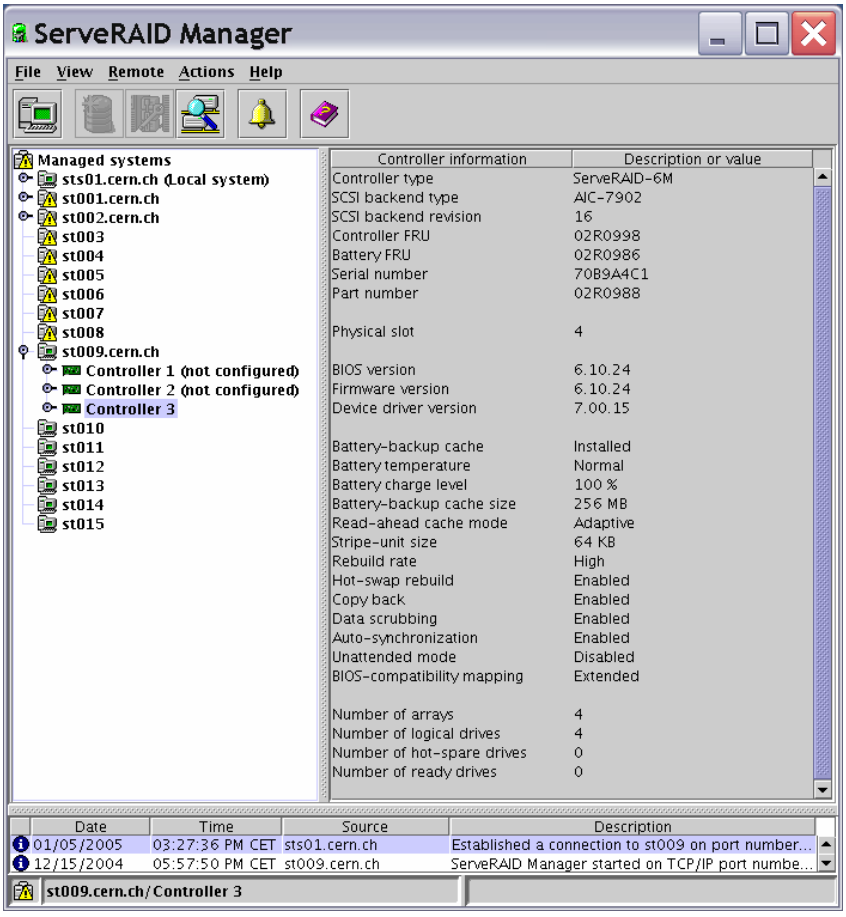


Figure 24: Targets st009-st015 RAID configuration.

8.2. Storage Tank detailed configuration

Below is the sequence of actions executed on the MDS to configure the pools, filesets, and policy rules. Note that the LUN id is always VPD83Type2 followed by the ASCII code of the VLUN name (with a padding to 96 characters), which is used as identifier.

```
for i in `seq 1 15`; do echo mkpool -blocksize 256k -partsize 256m P$i-256 | /root/admsim localhost 10191;
done;
```

```
sfscli mkvol -f -pool P1-256 -lun VPD83Type2=73743030312E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000 st001.cern.ch-
                2
sfscli mkvol -f -pool P1-256 -lun VPD83Type2=73743030312E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000 st001.cern.ch-
                3
sfscli mkvol -f -pool P1-256 -lun VPD83Type2=73743030312E6365726E2E63682D340000000000000000000000000000000000000000000000000000000000000000000000000000 st001.cern.ch-
                4
sfscli mkvol -f -pool P2-256 -lun VPD83Type2=73743030322E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000 st002.cern.ch-
                0
sfscli mkvol -f -pool P2-256 -lun VPD83Type2=73743030322E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000 st002.cern.ch-
                1
sfscli mkvol -f -pool P2-256 -lun VPD83Type2=73743030322E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000 st002.cern.ch-
                2
sfscli mkvol -f -pool P3-256 -lun VPD83Type2=73743030332E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000 st003.cern.ch-
                0
sfscli mkvol -f -pool P3-256 -lun VPD83Type2=73743030332E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000 st003.cern.ch-
                1
sfscli mkvol -f -pool P3-256 -lun VPD83Type2=73743030332E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000 st003.cern.ch-
                2
sfscli mkvol -f -pool P4-256 -lun VPD83Type2=73743030342E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000 st004.cern.ch-
                0
sfscli mkvol -f -pool P4-256 -lun VPD83Type2=73743030342E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000 st004.cern.ch-
                1
sfscli mkvol -f -pool P4-256 -lun VPD83Type2=73743030342E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000 st004.cern.ch-
                2
sfscli mkvol -f -pool P5-256 -lun VPD83Type2=73743030352E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000 st005.cern.ch-
                0
sfscli mkvol -f -pool P5-256 -lun VPD83Type2=73743030352E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000 st005.cern.ch-
                1
sfscli mkvol -f -pool P5-256 -lun VPD83Type2=73743030352E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000 st005.cern.ch-
                2
sfscli mkvol -f -pool P6-256 -lun VPD83Type2=73743030362E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000 st006.cern.ch-
                0
sfscli mkvol -f -pool P6-256 -lun VPD83Type2=73743030362E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000 st006.cern.ch-
                1
sfscli mkvol -f -pool P6-256 -lun VPD83Type2=73743030362E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000 st006.cern.ch-
                2
```

```
sfscli mkvol -f -pool P7-256 -lun VPD83Type2=73743030372E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000 st007.cern.ch-
                0
sfscli mkvol -f -pool P7-256 -lun VPD83Type2=73743030372E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000 st007.cern.ch-
                1
sfscli mkvol -f -pool P7-256 -lun VPD83Type2=73743030372E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000 st007.cern.ch-
                2
sfscli mkvol -f -pool P8-256 -lun VPD83Type2=73743030382E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000 st008.cern.ch-
                0
sfscli mkvol -f -pool P8-256 -lun VPD83Type2=73743030382E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000 st008.cern.ch-
                1
sfscli mkvol -f -pool P8-256 -lun VPD83Type2=73743030382E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000 st008.cern.ch-
                2
sfscli mkvol -f -pool P9-256 -lun VPD83Type2=73743030392E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000 st009.cern.ch-
                0
sfscli mkvol -f -pool P9-256 -lun VPD83Type2=73743030392E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000 st009.cern.ch-
                1
sfscli mkvol -f -pool P9-256 -lun VPD83Type2=73743030392E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000 st009.cern.ch-
                2
sfscli mkvol -f -pool P9-256 -lun VPD83Type2=73743030392E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000 st009.cern.ch-
                3
sfscli mkvol -f -pool P10-256 -lun VPD83Type2=73743031302E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st010.cern.ch-0
sfscli mkvol -f -pool P10-256 -lun VPD83Type2=73743031302E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st010.cern.ch-1
sfscli mkvol -f -pool P10-256 -lun VPD83Type2=73743031302E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st010.cern.ch-2
sfscli mkvol -f -pool P10-256 -lun VPD83Type2=73743031302E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st010.cern.ch-3
sfscli mkvol -f -pool P11-256 -lun VPD83Type2=73743031312E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st011.cern.ch-0
sfscli mkvol -f -pool P11-256 -lun VPD83Type2=73743031312E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st011.cern.ch-1
sfscli mkvol -f -pool P11-256 -lun VPD83Type2=73743031312E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st011.cern.ch-2
sfscli mkvol -f -pool P11-256 -lun VPD83Type2=73743031312E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st011.cern.ch-3
sfscli mkvol -f -pool P12-256 -lun VPD83Type2=73743031322E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st012.cern.ch-0
sfscli mkvol -f -pool P12-256 -lun VPD83Type2=73743031322E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st012.cern.ch-1
sfscli mkvol -f -pool P12-256 -lun VPD83Type2=73743031322E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st012.cern.ch-2
sfscli mkvol -f -pool P12-256 -lun VPD83Type2=73743031322E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st012.cern.ch-3
sfscli mkvol -f -pool P13-256 -lun VPD83Type2=73743031332E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st013.cern.ch-0
sfscli mkvol -f -pool P13-256 -lun VPD83Type2=73743031332E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st013.cern.ch-1
sfscli mkvol -f -pool P13-256 -lun VPD83Type2=73743031332E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st013.cern.ch-2
sfscli mkvol -f -pool P13-256 -lun VPD83Type2=73743031332E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st013.cern.ch-3
sfscli mkvol -f -pool P14-256 -lun VPD83Type2=73743031342E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st014.cern.ch-0
sfscli mkvol -f -pool P14-256 -lun VPD83Type2=73743031342E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st014.cern.ch-1
sfscli mkvol -f -pool P14-256 -lun VPD83Type2=73743031342E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st014.cern.ch-2
sfscli mkvol -f -pool P14-256 -lun VPD83Type2=73743031342E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st014.cern.ch-3
sfscli mkvol -f -pool P15-256 -lun VPD83Type2=73743031352E6365726E2E63682D300000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st015.cern.ch-0
sfscli mkvol -f -pool P15-256 -lun VPD83Type2=73743031352E6365726E2E63682D310000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st015.cern.ch-1
sfscli mkvol -f -pool P15-256 -lun VPD83Type2=73743031352E6365726E2E63682D320000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st015.cern.ch-2
sfscli mkvol -f -pool P15-256 -lun VPD83Type2=73743031352E6365726E2E63682D330000000000000000000000000000000000000000000000000000000000000000000000000000000000
                  st015.cern.ch-3
```

```
for i in `seq 1 15`; do sfscli mkfileset -attach cern.ch/c-filesets -dir c$i-1 c$i-1; done

cat complete.rule

   VERSION 1

   rule 'stgRule3' set stgpool 'P1-256' for fileset ('c1-1')
   rule 'stgRule5' set stgpool 'P2-256' for fileset ('c2-1')
   rule 'stgRule7' set stgpool 'P3-256' for fileset ('c3-1')
   rule 'stgRule9' set stgpool 'P4-256' for fileset ('c4-1')
   rule 'stgRule11' set stgpool 'P5-256' for fileset ('c5-1')
   rule 'stgRule13' set stgpool 'P6-256' for fileset ('c6-1')
   rule 'stgRule15' set stgpool 'P7-256' for fileset ('c7-1')
   rule 'stgRule17' set stgpool 'P8-256' for fileset ('c8-1')
   rule 'stgRule19' set stgpool 'P9-256' for fileset ('c9-1')
   rule 'stgRule21' set stgpool 'P10-256' for fileset ('c10-1')
   rule 'stgRule23' set stgpool 'P11-256' for fileset ('c11-1')
   rule 'stgRule25' set stgpool 'P12-256' for fileset ('c12-1')
   rule 'stgRule27' set stgpool 'P13-256' for fileset ('c13-1')
   rule 'stgRule29' set stgpool 'P14-256' for fileset ('c14-1')
   rule 'stgRule31' set stgpool 'P15-256' for fileset ('c15-1')

sfscli mkpolicy -file complete.rule complete
sfscli usepolicy complete
```

8.3. Test procedure details

The initial state is:

- All clients stopped,

- MDS cluster stopped, and
- all targets stopped.

The steps to start a test are:

- Start the targets
  ```
  rha@lxplus# wassh root@st0[01-15] ‘service iscsi_tgt start’
  ```
- Start the cluster
  ```
  root@sts01# sfscli startcluster
  ```
- Start the clients
  ```
  rha@lxplus# wassh root@lxs50[13-37]
  ‘/usr/tank/client/bin/client_control.sh start’
  ```
- Start the perfmon daemons on all clients
  ```
  rha@lxplus# wassh root@lxs50[13-37] ‘service stfsperfmon start’
  rha@lxplus# wassh root@sts06 ‘service stfsperfmon start’
  ```
- Start the test procedure with desired readers/writers and threads
  ```
  rha@lxplus# go.wassh lxs50[13-25] 4 lxs50[26-37] 6
  ```
- Start the perfmon collector on sts06
  ```
  root@sts06# nohup stfsperfmonc 1000000 10 10 localhost 666 >
  stfsperfmonc.Sun-2004-Dec-26-16_25.12R-4T-12W-6T.maxtput180000.out &
  ```
- Start ArloCop on sts06 with the desired number of files to be processed, and as stdin the mapping between pools and full fileset path (CERN.cfg).
  ```
  root@sts06# nohup ./ArloCop -N 180000 < CERN.cfg > arlo.Sun-2004-Dec-
  26-16_25.12R-4T-12W-6T.maxtput180000.out &
  ```
- In the newly created directory, give the signal for the test to start
  ```
  root@sts06 /tank/cern.ch/arlo.Sun-2004-Dec-26-16_25.12R-4T-12W-
  6T.maxtput180000#  date > go.reader; date > go.writer
  ```