HEPiX Storage Working Group
- progress report 4.2010 -

Andrei Maslennikov

November, 2010 – Ithaca

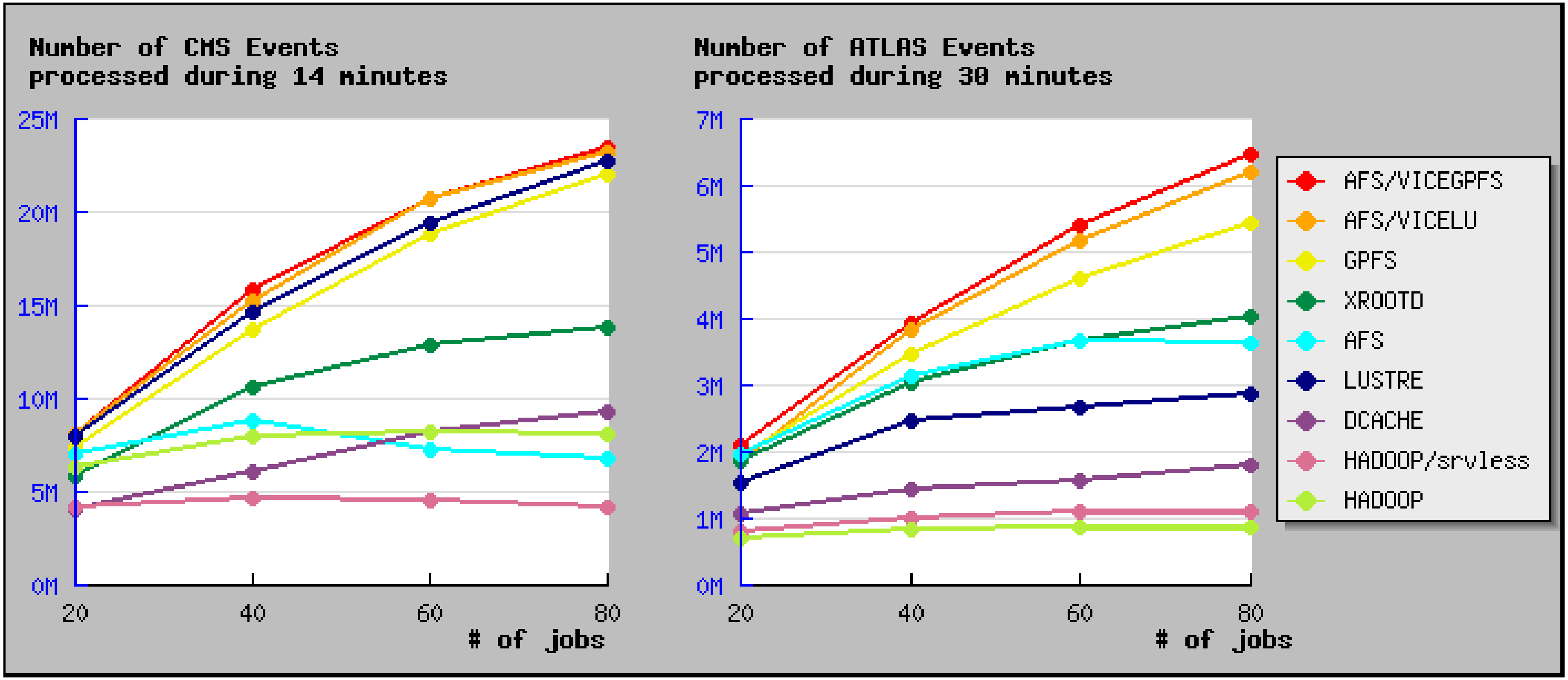
Summary

- Activities May-October 2010
- Current results obtained at KIT
- Plans for the next months
- Discussion

Activities May-August 2010

- In May-June 2010 the group concluded the full round of performance measurements and presented its second progress report at the WLCG Storage Jamboree meeting in Amsterdam in mid-June.
- In July 2010 FNAL offered us to investigate two new use cases (Minos and Nova experiments), and initial setup and first measurements were performed. It came out that both cases were mostly CPU-bound and thus the influence of underlying storage proved to be marginal. Nova people then decided to prepare a new variant of their job which proved to be successful; the updated use case was then included into our list as of October.
- In August 2010 further dCache investigation and tuning was performed.

Results 2.2010



These summary results were obtained with the CMS-1 and ATLAS-1 use cases and reported in Amsterdam. During that meeting it was decided to switch to the newest ATLAS and CMS frameworks starting as of October 2010. See the Amsterdam report for details.

Activities September-October 2010

- In September 2010 the group was evaluating a new development version of AFS (1.5.77) only to discover that it performed visibly worse compared with the current one (1.4.12.1). These results were then presented at the European AFS Conference in Plzen in September as progress report 3.2010 and triggered an intensive debug session with AFS Gatekeepers that helped them to locate and fix this problem as of the 1.5.78 release.

- In the beginning of October 2010 we updated the operating system and migrated the file system software to most recent versions. In parallel, new use cases were being prepared;

- As of the 10th of October started a new round of measurements; some first results were already obtained, but a lot more time is needed as the program of tests is quite large, numbers have to be verified, further tuning has to performed etc.

Credits 2010

- The new test laboratory at KIT was built on the top of hardware kindly provided by Karlsruhe Institute of Technology (rack and network infrastructure, load farm) and E4 Computer Engineering (new disk server). CERN had contrubuted with some funds to cover a part of human hours.
- These people participated in provisioning, funding, discussions, laboratory building, preparation of test cases and test framework, tests and elaboration of the results:

CASPUR	A.Maslennikov (Chair), M.Calori (Web Master)
CEA	J-C.Lafoucriere
CERN	B.Panzer-Steindel, D. van der Ster, R.Toebbicke
DESY	M.Gasthuber, P.van der Reest, D.Ozerov
E4	C.Gianfreda
FNAL	G.Garzoglio, A.Norman, R.Hatcher
INFN	G.Donvito, V.Sapunenko
KIT	J.van Wezel, A.Trunov, M.Alef, B.Hoeft
LAL	M.Jouvin
RZG	H.Reuter
U of Edinburgh	W.Bhimji

Storage Laboratory (Oct.2010->)

Goals

- **As in the previous years, we aim at the performance comparison of most diffused storage solutions (AFS, GPFS, Lustre, dCache, Xrootd , Hadoop etc)**

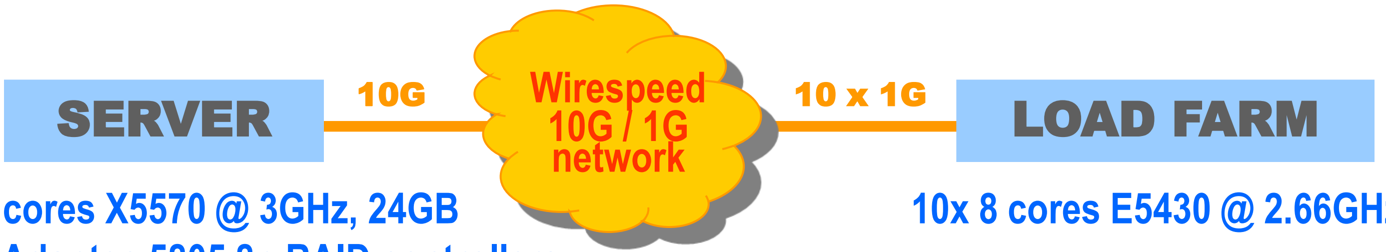
- **Comparison is being done on the common hardware base, employing a set of realistic use cases relevant for the HEP community; one of our ancillary goals is thus to enlarge and keep up-to-date the use case library.**

Disclaimer

- **We are constantly dealing with the “moving target”: data formats and use cases are evolving, hardware base is changing, new versions of storage access and archival software replace the old ones. This implies that results obtained in the storage laboratory are and will always remain a subject to change.**

- **Whatever we report should hence aways be seen as “work in progress”. We are not trying to provide any final recommendations but are rather sharing with you our findings and are ready to accept any advice and feedback.**

Hardware setup 2010 at KIT



8 cores X5570 @ 3GHz, 24GB
3 Adaptec 5805 8p RAID controllers
24 Hitachi drives of 1 TB
1 Intel 82598EB 10G NIC

10x 8 cores E5430 @ 2.66GHz,16GB

This setup reperesents well an elementary fraction of a typical large hardware installation and has basically no bottlenecks:

- **Each of the three Adaptec controllers may deliver 600+ MB/sec (R6)**
- **Ttcp memory-memory network test (1 server – 10 clients) shows full 10G speed**

Details of the current test environment

- RHEL 5.5+/64bit on all nodes (kernels 2.6.18-164.11.1.lustre and 2.6.18-194.17.1)
- Lustre 2.0.0.1
- GPFS 3.2.1-23
- OpenAFS/OSD 1.4.12 (trunk 984)
- dCache 1.9.7 (to be updated)
- Xrootd 20100617-1658 with default settings
- Hadoop 0.20-1+169.89 from Cloudera (to be updated)

Use cases October 2010 - April 2011

- New CMS use case (CMS-2): CMS/ “MTR3” standalone job
  fw v.3.9.0pre5, read-in + basic computations (inv.masses, track isolation)
  (Giacinto Donvito)

- New ATLAS use case (ATLAS-2): ATLAS/“Hammercloud” standalone job
  fw v.15.9.0, root 5.26.00c, TTcache support, scans and randomly navigates inside the root data files (Daniel van der Ster)

- New ATLAS use case (ATLAS-3): ATLAS/ “D3PDMaker” standalone job
  fw v.15.9.0 (Wahid Bhimji)

- New Nova use case (NOVA-1): Nova/ANA standalone analysis job with condensed output stream (Andrew Norman)

How the tests are performed

In all cases with the only exception of Hadoop/serverless, the method was as follows:

- Configure the server and client parts of a solution under test;
- Load the ATLAS and CMS data files into the data area under test;
- Run 20,40,60,80 jobs per 10-node cluster (2,4,6,8 jobs per node); each of the jobs is processing a dedicated non-shared set of event files;
- In each of the measurements start all the jobs simultaneously and then kill them simultaneously, after some predefined period of smooth running;
- Count the total numbers of events processed in each of the runs; These numbers may be compared directly for all solutions under test.
- While the jobs are running, measure also the average incoming MB/sec on each of the 10 Ethernet interfaces of the worker nodes;
- Try to tune each of the solutions under test to get the largest possible numbers of events processed per predefined period;

Hadoop/serverless configuration:

All 10 worker nodes all acted as data providers and data clients. Each of the nodes had 2 disk drives, so in the end we had 20 data drives. As in the case of server we had 18 data drives after R6 formatting, it made sense to compare the Hadoop/serverless test results with those of the server-based configurations.

Tunables

We report here, for reference, some of the relevant settings that were used so far.

Diskware: three stanadlone RAID-6 arrays of 8 spindles, stripe size=1M;
played a lot with disk readaheads, negligible influence on final results

Lustre:
No checksumming, No caching on server
Formatted with: "-E stride=256 -E stripe-width=1536"
Data were spread over 3 file systems (1 MGS +3 MDT)
OST threads: "options ost oss_num_threads=512"
Read-aheads on clients: 4MB (CMS), 10MB (ATLAS) later converged on 4MB

GPFS:
3 NSDs, one per RAID-6 array, 3 file systems (one per NSD)
-B 4M –j cluster - maxMBpS 1250 - maxReceiverThreads 128
nsdMaxWorkerThreads 128 - nsdThreadsPerDisk 8 - pagepool 2G

AFS, dCache, Xrootd
3 XFS partitions (one per RAID array)
Formatted with: "-i size=1024 -n size=16384 -l version=2 -d sw=6,su=1024k"
Mounted with: "logbsize=256k,logbufs=8,swalloc,inode64,noatime"
Afsd options: "memcache, chunksize varied, cache size 500MB" (Vice/Lu, Vice/GPFS)
"memcache, chunksize varied, cache size 4GB" (Native)
Xrootd with TTreeCache on (ATLAS-2)
dCache library: libdcap++ from Ganga

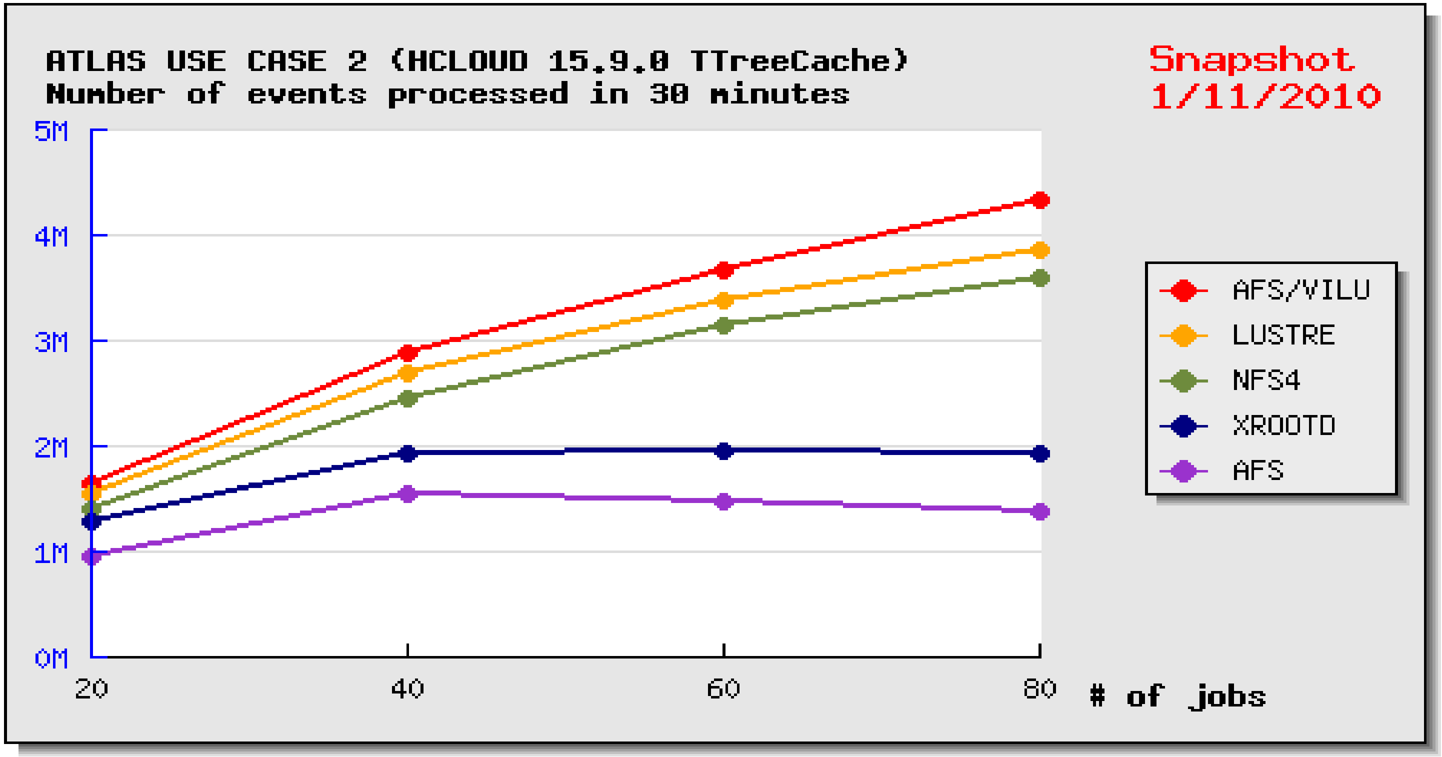
Hadoop
fuse 2.7.4-8, rdbuffer=131072, /dev/sdX readaheads of 16M
3 XFS partitions (with server) like in dCache test, or 20 ext4 partitions (serverless)
(*) Unstable under heavy load (write aborts on massive writes, few crashes on reads)

Where we are at the moment

- We tried all 4 use cases; initially each of them had running and tuning issues. After a first series of runs we've discovered that the totally new CMS use case might require further tuning on the server side, so we decided put it into bottom of our list. As well, the ATLAS-3 D3PDMaker use case proved to be mostly CPU-bound and hence was excluded.

- Thus we started with ATLAS-2 and Nova use cases and were already able to obtain some first results with AFS, Lustre, NFS4, AFS/Lu and Xrootd.

Current ATLAS-2 results (HCLOUD 15.9.0/TTcache)



	20 threads	40 threads	60 threads	80 threads
AFS	140 MB/sec 944961 evs	231 MB/sec 1544591 evs	220 MB/sec 1477365 evs	210 MB/sec 1390900 evs
Xrootd TT	81 MB/sec 1281975 evs	120 MB/sec 1921599 evs	126 MB/sec 1945455 evs	122 MB/sec 1930212 evs
NFS4 TT	193 MB/sec 1407548 evs	337 MB/sec 2447510 evs	439 MB/sec 3140749 evs	501 MB/sec 3593481 evs
LU TT	274 MB/sec 1544840 evs	488 MB/sec 2688586 evs	665 MB/sec 3382907 evs	807 MB/sec 3847666 evs
AFS/LU TT	287 MB/sec 1640129 evs	532 MB/sec 2878453 evs	712 MB/sec 3657741 evs	842 MB/sec 4322025 evs

TTreeCache effects

- The previous ATLAS framework under test was assembled using the production version of Root of 2009 (5.22.00d). It was sensitive to the Root caching parameters passed via the file name suffix. In particular, we were able to increase 4+ fold the efficiency for ATLAS/Xrootd using these parameters as was suggested by F. Furano.
- For instance, this is an example of how ATLAS/Xrootd framework behaved in vanilla variant, and after feeding in the client caching instructions:

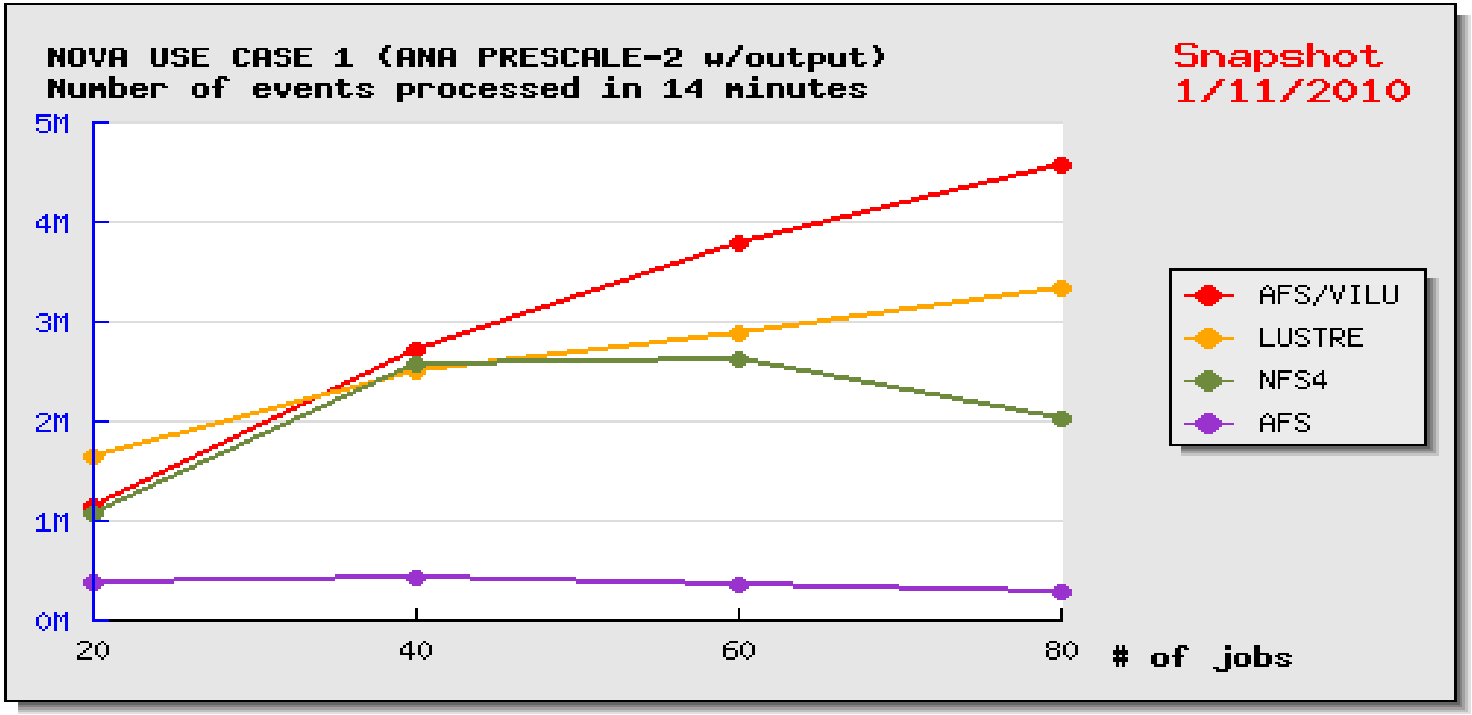
```
+---------+---------------------------------------------------------+
|Xrootd   |     985 MB/sec    1132 MB/sec    1153 MB/sec   1156 MB/sec   |
|Vanilla  |     808374 evs     913080 evs     910937 evs    895540 evs   |
+---------+---------------------------------------------------------+
|Xrootd   |     445 MB/sec     745 MB/sec     913 MB/sec   1035 MB/sec   |
|Cache.suf|    1855726 evs    3034830 evs    3659365 evs   4024395 evs   |
+---------+---------------------------------------------------------+
```

- With the new 15.0.9 framework based on root 5.26.00c a similar behaviour may be obtained activating the 10 MB TTreeCache in the Athena input file:

```
+---------+---------------------------------------------------------+
|Xrootd   |      32 MB/sec      31 MB/sec      31 MB/sec     31 MB/sec   |
|Vanilla  |     486496 evs     479467 evs     468579 evs    464436 evs   |
+---------+---------------------------------------------------------+
|Xrootd   |      81 MB/sec     120 MB/sec     126 MB/sec    122 MB/sec   |
|TTreeCach|    1281975 evs    1921599 evs    1945455 evs   1930212 evs   |
+---------+---------------------------------------------------------+
```

NB Switching TTC on improves file systems results, as well!

Current Nova results (ANA P-2 R/W)



		20 threads	40 threads	60 threads	80 threads
AFS	R	4 MB/sec	9 MB/sec	21 MB/sec	33 MB/sec
	W	11 MB/sec	20 MB/sec	28 MB/sec	34 MB/sec
		379552 evs	431680 evs	356838 evs	286523 evs
NFS4	R	62 MB/sec	117 MB/sec	152 MB/sec	312 MB/sec
	W	62 MB/sec	116 MB/sec	152 MB/sec	168 MB/sec
		1069374 evs	2568922 evs	2622714 evs	2024504 evs
LUSTRE	R	89 MB/sec	165 MB/sec	226 MB/sec	247 MB/sec
	W	66 MB/sec	120 MB/sec	168 MB/sec	187 MB/sec
		1646140 evs	2501257 evs	2869667 evs	3324682 evs
AFS/LU	R	44 MB/sec	159 MB/sec	221 MB/sec	270 MB/sec
	W	57 MB/sec	120 MB/sec	170 MB/sec	203 MB/sec
		1152652 evs	2717537 evs	3785615 evs	4582494 evs

Immediate plans

- **The group is planning to run and round up several series of lab tests at KIT by the Spring 2011 meeting at GSI. Starting March 2011 we shall be publishing the detailed results' summaries in the hope to get some preliminary feedback.**
- **The minimal program includes the new ATLAS, CMS and Nova probes against dCache, NFS4.1, AFS 1.4.xx, AFS 1.5.xx, Lustre, AFS/VILU, GPFS, AFS/VIGPFS, Xrootd and Hadoop.**
- **This time special efforts will be made to tune the hardware RAID setup individually for each of the solutions under test, also with the help of I/O pattern profiling.**

Discussion