HEPiX Storage Working Group
- progress notes 11/2011 -

Andrei Maslennikov

October, 2011 – Vancouver

Summary

- Goals
- Activities July-October 2011
- Current results
- Discussion

Goals

- The group was created in the end of 2006 to make an assessment of the most diffused HEP storage solutions and to compare them.
- In the period of 2007-2010 we ran two major storage questionnaires, performed 8 series of comparative performance measurements with realistic use cases. More than 50 phone conferences were held, some 30 people participated, 9 progress reports were delivered.
- In mid-2011 a meeting was held to discuss the future of the group, and it was unanimously decided that evaluation work should continue. Although we are unable to estimate the direct practical impact of our reports, we continue receiving positive expressions of interest from sites and this is quite stimulating.

Activities July-October 2011

- During the summer meeting it was decided to plan for a new lab session at the test facility at KIT, and to report the results during the Vancouver workshop.
- In September the group ported operating systems and software under test to the new levels. The most recent use cases for ATLAS and CMS experiments were prepared and the tests were run as of the first week of October 2011.
- We started with AFS and this gave us an option to obtain a couple of numbers for most recent GA version (1.6) in time to be able to report them in the European AFS Conference on the 6th of October. We then proceeded with other solutions (NFS, Xrootd, Lustre and GPFS), and have collected new results that will be presented today.

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.

Credits 2011

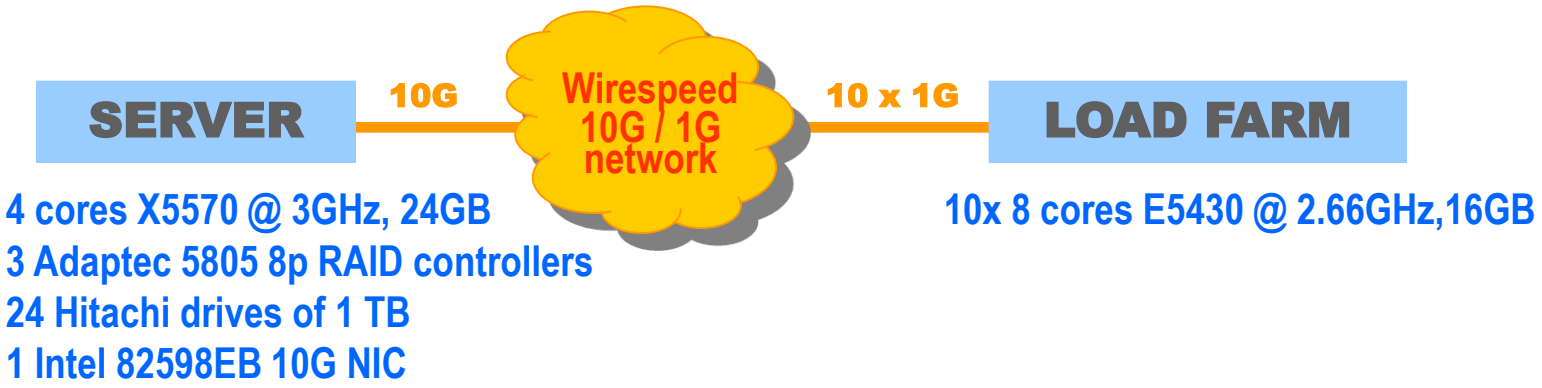
- The test laboratory at KIT was built on the top of hardware kindly provided by Karlsruhe Institute of Technology (rack and network infrastructure, load farm) and E4 Computer Engineering (disk server). CERN contributed with some funds to cover a part of human hours.

- These people participated in provisioning, funding, discussions, laboratory building, preparation of test cases and test framework, tests and elaboration of results (year 2011):

CASPUR	A.Maslennikov (Chair), M.Calori (Web Master)
CERN	B.Panzer-Steindel, D. van der Ster, R.Toebbicke
DESY	M.Gasthuber, P.van der Reest, D.Ozerov
INFN	G.Donvito
KIT	J.van Wezel, Ch-E. Pfeiler, M.Alef, B.Hoeft
LAL	M.Jouvin
LMU	J. Elmsheuser, F.Legger
RZG	H.Reuter

Storage Laboratory

Hardware setup 2011 at KIT



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 5.7+/64bit on all nodes (kernels 2.6.18-274.3.1 on clients and 2.6.18-238.19.1.el5_lustre.g65156ed on server)
- Lustre 2.1
- GPFS 3.4.0-6
- OpenAFS 1.6
- OpenAFS/OSD trunk 1194 from DESY svn server
- Xrootd 3.0.5

Current use cases

- New CMS use case (CMS-2001-1): «Data scan» standalone job fw CMSSW_4_4_0_pre9, root 5.28.00d mostly sequential I/O (Giacinto Donvito /INFN)
- New ATLAS use case (ATLAS-2011-1): ATLAS «Hcloud/athena» standalone job, fw 16.0.3, root 5.26.00e, scans and randomly navigates inside the root data files (Dmitry Ozerov /DESY)
- New ATLAS use case (ATLAS-2011-2): ATLAS/ «Ntuple/root» standalone «athenaless» ntuple analysis job, fw 16.0.3, root 5.26.00e, mostly random I/O (Dmitry Ozerov /DESY)
- Nova use case (NOVA-1): Nova/ANA standalone analysis job with condensed output stream – bidirectional I/O (Andrew Norman/FNAL)

How the tests are performed

In all cases the method was as follows:

- **Configure the server and client parts of a solution under test;**
- **Load the 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;**
- **Calculate the processing speed in terms of events/second (first wait until all the jobs completed the initialization phase and then start counting the events since this moment until the jobs are killed). These speed numbers may then 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 processing speeds.**

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 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: 10MB

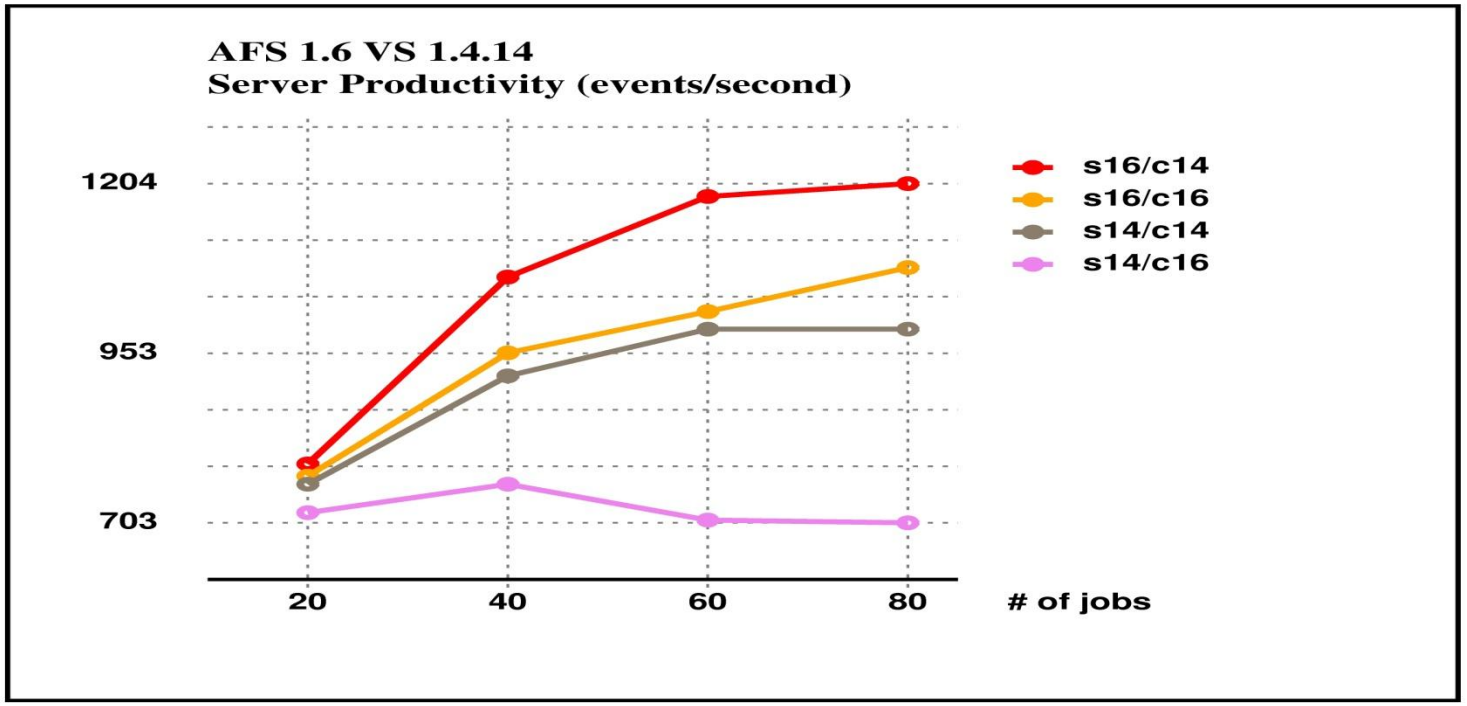
GPFS: 3 NSDs, one per RAID-6 array, 3 file systems (one per NSD)
-B 4M –j cluster - maxMBpS 1250 - maxReceiverThreads 128
nsdMaxWorkerThreads 128 - nsdThreadsPerDisk 8 - pagepool 2G

AFS, 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 16, cache size 500MB"

AFS/VICE Lustre (enable lustre hack, fast read) chunk 16, c.size 65M, Lu readahead 40M
GPFS (fast read) chunk 22 c.size 1GB (ATLAS/CMS), 500MB (Nova)

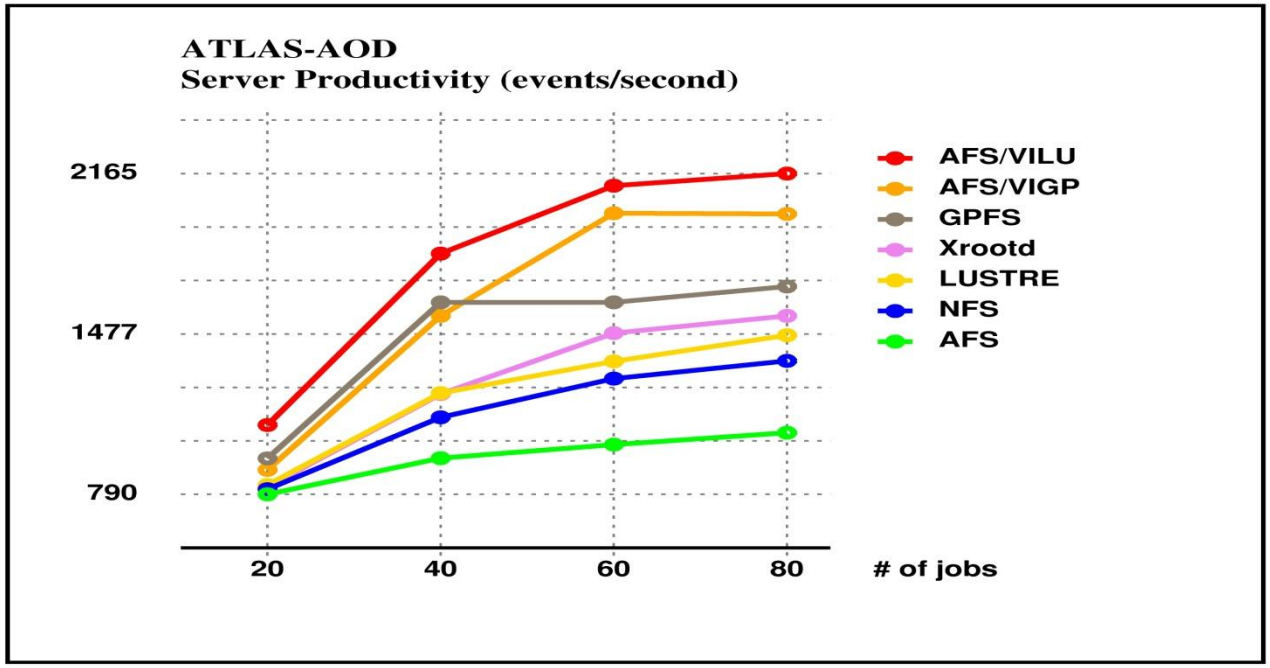
Current results

AFS 1.6 vs 1.4 (Hammercloud 16.0.3 - AOD)



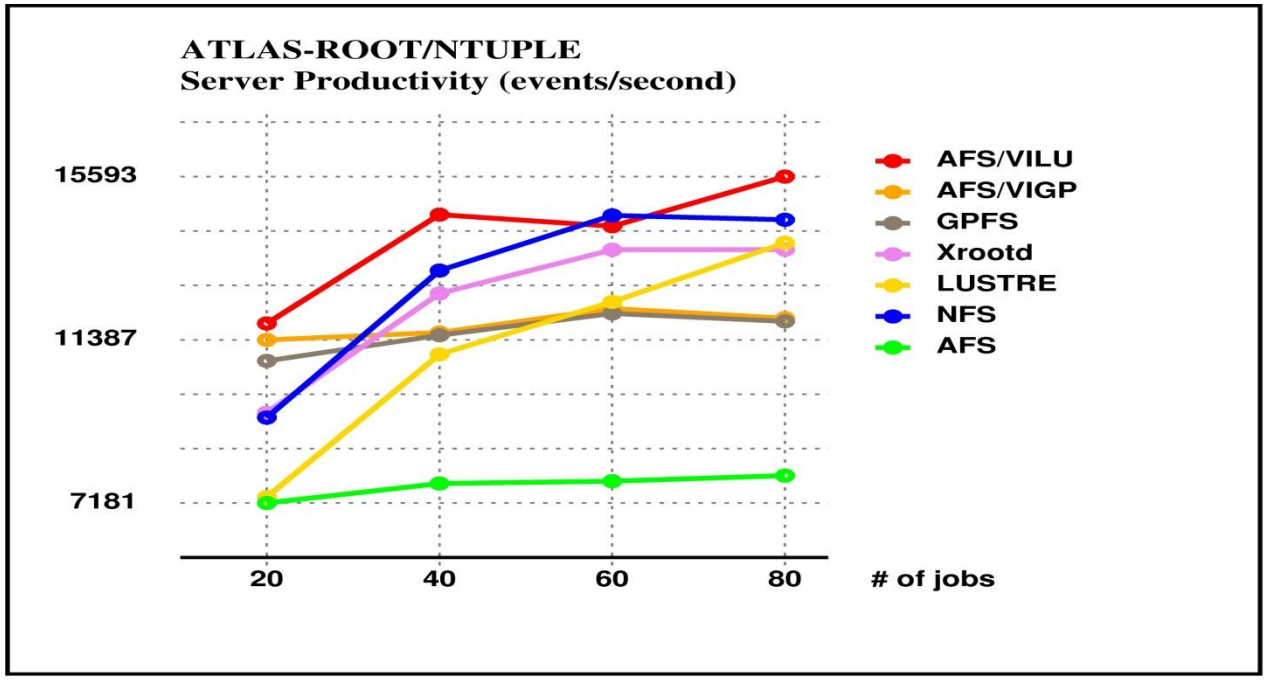
	20 jobs	40 jobs	60 jobs	80 jobs
SRV 1.6	120 MB/sec	166 MB/sec	182 MB/sec	185 MB/sec
CLI 1.4	790 EV/sec	1066 EV/sec	1185 EV/sec	1204 EV/sec
SRV 1.6	150 MB/sec	183 MB/sec	191 MB/sec	194 MB/sec
CLI 1.6	772 EV/sec	954 EV/sec	1015 EV/sec	1080 EV/sec
SRV 1.4	113 MB/sec	143 MB/sec	148 MB/sec	147 MB/sec
CLI 1.4	760 EV/sec	920 EV/sec	989 EV/sec	989 EV/sec
SRV 1.4	134 MB/sec	144 MB/sec	137 MB/sec	130 MB/sec
CLI 1.6	718 EV/sec	760 EV/sec	707 EV/sec	703 EV/sec

ATLAS Hammercloud 16.0.3 - AOD



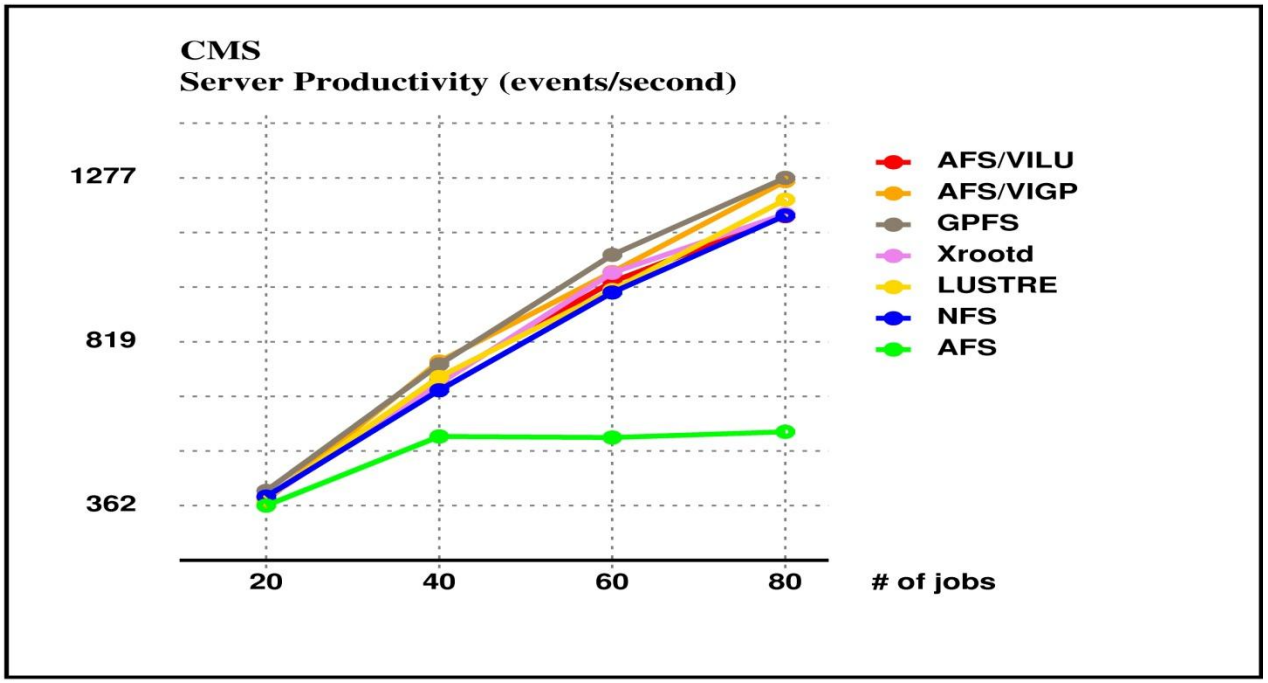
	20 jobs	40 jobs	60 jobs	80 jobs
AFS	152 MB/sec 790 EV/sec	185 MB/sec 945 EV/sec	191 MB/sec 1003 EV/sec	193 MB/sec 1054 EV/sec
NFS p4	118 MB/sec 812 EV/sec	171 MB/sec 1120 EV/sec	186 MB/sec 1286 EV/sec	206 MB/sec 1362 EV/sec
LUSTRE	340 MB/sec 830 EV/sec	510 MB/sec 1224 EV/sec	571 MB/sec 1360 EV/sec	574 MB/sec 1472 EV/sec
Xrootd	67 MB/sec 822 EV/sec	102 MB/sec 1220 EV/sec	115 MB/sec 1481 EV/sec	127 MB/sec 1555 EV/sec
GPFS	420 MB/sec 944 EV/sec	704 MB/sec 1613 EV/sec	730 MB/sec 1613 EV/sec	688 MB/sec 1681 EV/sec
AFS/VIGP	380 MB/sec 895 EV/sec	684 MB/sec 1555 EV/sec	855 MB/sec 1995 EV/sec	842 MB/sec 1992 EV/sec
AFS/VILU	456 MB/sec 1087 EV/sec	759 MB/sec 1822 EV/sec	901 MB/sec 2113 EV/sec	735 MB/sec 2165 EV/sec

ATLAS Hammercloud 16.0.3 – Root/Ntuple



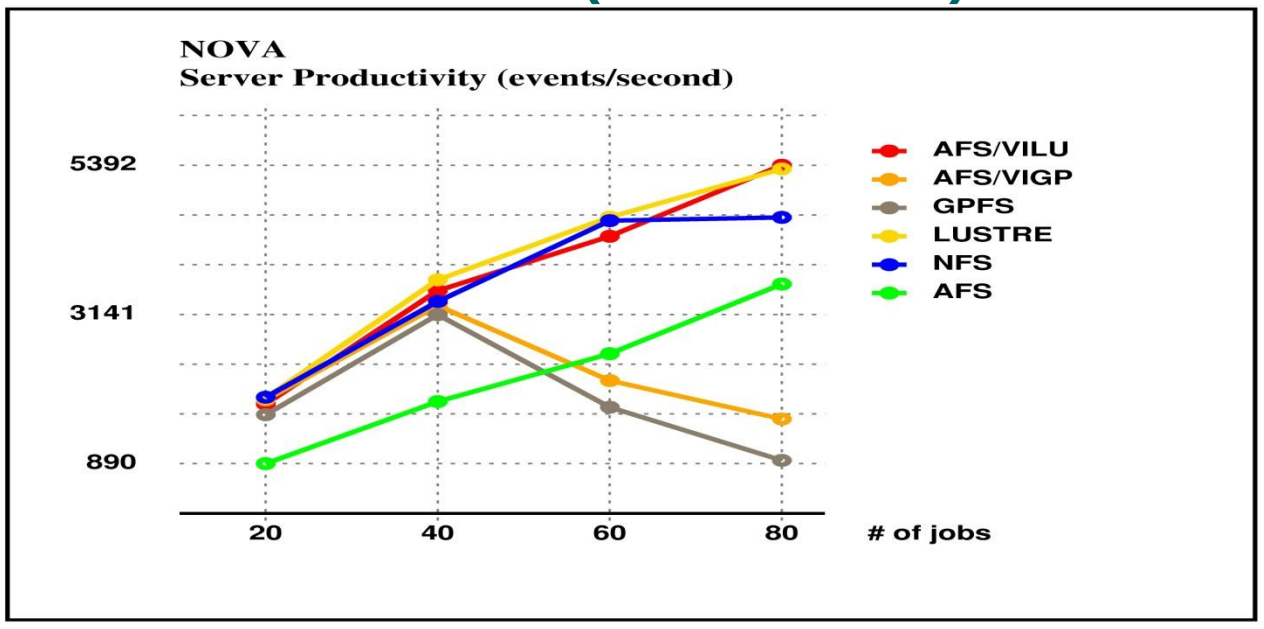
	20 jobs	40 jobs	60 jobs	80 jobs
AFS	200 MB/sec 7181 EV/sec	207 MB/sec 7683 EV/sec	207 MB/sec 7743 EV/sec	208 MB/sec 7887 EV/sec
GPFS	636 MB/sec 10843 EV/sec	679 MB/sec 11504 EV/sec	702 MB/sec 12066 EV/sec	722 MB/sec 11861 EV/sec
AFS/VIGP	633 MB/sec 11384 EV/sec	685 MB/sec 11575 EV/sec	698 MB/sec 12193 EV/sec	708 MB/sec 11944 EV/sec
LU	192 MB/sec 7333 EV/sec	286 MB/sec 11012 EV/sec	331 MB/sec 12364 EV/sec	353 MB/sec 13887 EV/sec
Xrootd	33 MB/sec 9508 EV/sec	44 MB/sec 12582 EV/sec	48 MB/sec 13708 EV/sec	50 MB/sec 13711 EV/sec
NFS	131 MB/sec 9382 EV/sec	194 MB/sec 13171 EV/sec	211 MB/sec 14592 EV/sec	210 MB/sec 14481 EV/sec
AFS/VILU	709 MB/sec 11808 EV/sec	874 MB/sec 14613 EV/sec	881 MB/sec 14313 EV/sec	918 MB/sec 15593 EV/sec

CMS 4.4.0.pre9



	20 jobs	40 jobs	60 jobs	80 jobs
AFS	140 MB/sec 362 EV/sec	224 MB/sec 555 EV/sec	216 MB/sec 552 EV/sec	216 MB/sec 568 EV/sec
NFS	150 MB/sec 387 EV/sec	271 MB/sec 684 EV/sec	370 MB/sec 957 EV/sec	447 MB/sec 1172 EV/sec
Xrootd	148 MB/sec 382 EV/sec	280 MB/sec 703 EV/sec	381 MB/sec 1012 EV/sec	460 MB/sec 1176 EV/sec
AFS/VILU	143 MB/sec 384 EV/sec	268 MB/sec 713 EV/sec	363 MB/sec 987 EV/sec	431 MB/sec 1170 EV/sec
LUSTRE	140 MB/sec 382 EV/sec	260 MB/sec 721 EV/sec	362 MB/sec 964 EV/sec	443 MB/sec 1216 EV/sec
AFS/VIGP	257 MB/sec 376 EV/sec	275 MB/sec 764 EV/sec	374 MB/sec 1014 EV/sec	444 MB/sec 1268 EV/sec
GPFS	151 MB/sec 403 EV/sec	282 MB/sec 756 EV/sec	374 MB/sec 1062 EV/sec	450 MB/sec 1277 EV/sec

NOVA (bidirectional)



		20 jobs	40 jobs	60 jobs	80 jobs
AFS	R	28 MB/sec	58 MB/sec	79 MB/sec	84 MB/sec
	W	28 MB/sec	61 MB/sec	85 MB/sec	92 MB/sec
		752 EV/sec	1256 EV/sec	1464 EV/sec	1081 EV/sec
GPFS	R	121 MB/sec	230 MB/sec	250 MB/sec	158 MB/sec
	W	89 MB/sec	162 MB/sec	220 MB/sec	260 MB/sec
		1628 EV/sec	3137 EV/sec	1741 EV/sec	939 EV/sec
AFS/G	R	130 MB/sec	249 MB/sec	259 MB/sec	207 MB/sec
	W	92 MB/sec	170 MB/sec	197 MB/sec	232 MB/sec
		1851 EV/sec	3274 EV/sec	2143 EV/sec	1563 EV/sec
NFS	R	66 MB/sec	120 MB/sec	161 MB/sec	178 MB/sec
	W	65 MB/sec	120 MB/sec	160 MB/sec	178 MB/sec
		1891 EV/sec	3336 EV/sec	4556 EV/sec	4605 EV/sec
LU	R	110 MB/sec	211 MB/sec	300 MB/sec	360 MB/sec
	W	67 MB/sec	129 MB/sec	170 MB/sec	210 MB/sec
		1880 EV/sec	3662 EV/sec	4609 EV/sec	5336 EV/sec
AFS/L	R	98 MB/sec	200 MB/sec	290 MB/sec	351 MB/sec
	W	60 MB/sec	120 MB/sec	170 MB/sec	210 MB/sec
		1790 EV/sec	3499 EV/sec	4320 EV/sec	5392 EV/sec

Observations

- ATLAS AOD: the spread between Xrootd and best players is visibly improved compared to the previous sessions, but remains pretty large (productivity ratio «best»/Xrootd up to 1.5)
- ATLAS ROOT/NTUPLE: all solutions except AFS are pretty close
- CMS: all solutions except AFS go «nose-to-nose». Each thread consumes close to 100% of a core and is basically busy with data decompression. We cannot saturate the server with this use case and a load farm of 10 nodes.
- NOVA: GPFS results look surprising. But the settings were tuned for ATLAS/CMS read-only use cases. We have to further investigate it.
- Of all solutions, AFS/VILU looks like the only one capable to deliver highest rates for all four use cases..

Immediate plans

- We are planning to continue with the current test session until February 2012. Will be rechecking GPFS/NOVA, will be debugging AFS with the Gatekeepers.
- Might need to potentiate the test setup (more worker nodes, more powerful server) to address the use cases similar to that of CMS.
- Plan to use the KIT setup in 2012 for tests with Openstack/Swift; will relook into Hadoop. May take a peek at other solutions.
- Will repeat the Storage Questionnaire for Prague meeting.

Discussion