AFS-OSD Status Update

Hartmut Reuter
reuter@rzg.mpg.de

What is new?

- Since the meeting in Rome only few things have been changed in AFS-OSD
  - New algorithm for choosing OSDs to get a more uniform life time of files in the on-line storage (OSDs)
  - Introduction of shared locks in the fileserver to allow read access to files during the creation of archival copies.
  - many bug-fixes, of course
  - Integration of the OSD code into OpenAFS 1.5
  - Interface to HPSS as underlying HSM-system (separate talk tomorrow)

Algorithm for Choosing OSD

- During the last year our cell experienced a substancial data growth
  - from 340 TB as reported at Rome to ~ 600 TB now.
  - Most of these data are large files of long time archives which get archival copies in the underlying HSM system (TSM-HSM), but don't need to stay on-line
- It turned out that the old allocation algorithm lead to very different life-times of files on the different OSDs
- The new algorithm uses the **atime** of the newest wiped file on each OSD which the wiper daemon stores in the OSDDB database.
  - Presently the on-line life-time of files in wipeable OSDs varies only between 9 and 11 weeks which is much better than before

Shared Lock in the fileserver

- The biggest part of the data growth seen in the last slide was produced by experiment files transferred from SLAC into our cell.
  - They were copied from SLAC to Garching using Barbar-Copy which is much faster than AFS.
- The data arrived on the OSDs with a average rate of about 50 MB/s which is faster than our archival server.
  - Therefore we used two different archival servers to keep pace with the incoming data. Nearly permanently one file in the directory was being archived blocking „ls -l“-commands.
- Now the archiving is done under a shared lock which allows read access, but no write or other access which modifies the metadata (another archive e.g.)
  - „ls -l“ and also read operations are now not blocked any more.
- While in the cache manager shared locks were used since long they are new in the fileserver.

OpenAFS-1.5-osd

- As Jeffrey Altman stated last year in his YFS talk:
  - Integration of OSD-Support is scheduled for the stable release 1.8
  - After branching off the 1.6 release OSD patches may start to drop into git.
- The current 1.5 release of OpenAFS is based on the git master which differs substancially from the 1.4 release
  - Therefore it is a good preliminary practice to try the integration into 1.5
- I am using Subversion to keep track of the integration steps.
  - It might seem more appropriate to use git, but I am more experienced with subversion.
- I started with OpenAFS-1.5.74 and have now upgraded to OpenAFS-1.5.76

OpenAFS-1.5-osd

- Unlike in my 1.4 version in 1.5 the client should unconditionally understand OSD and also embedded shared filesystems for OSD or fileserver partitions (vicep-access).
- The configure options –enable-object-storage and –enable-vicep-access control only the server side, not the client code.
  - So any client in the world should be able to access efficiently data in OSDs, even if its home cell doesn't make use of OSDs.
- Porting code from 1.4 to 1.5 is quite some work because of changes in some structures in the cache manager.
- On the server side, of course, the Demand Attach server is supported in 1.5-osd.
- We are not running the 1.5 servers and clients in production yet, only in our test cell and on my laptop.

OpenAFS-1.5 Experience

- In the beginning the cache manager was extremly slow for read.
  - Jeffrey Altman helped me with a number of patches in the rx-layer which finally solved the problem and made it into 1.5.76
  - With the bigger rx-window-size you need to make sure the system wide UDP buffer-sizes are big enough. Use “netstat -s” to check no packets were thrown away.
- The Demand Attach Fileserver works fine at least in the limited test environment.
  - I am happy that the „–unsafe-nosalvage“ command line option has been added in 1.5.76 because at our site salvaging each volume during attach after a fileserver crash would be far too slow.
  - The present 1.4.12 fileserver requires the –enable-fast-restart because with the huge partitions the salvager gets out of memory without having salvaged the volumes which then have to be salvaged „by hand“.

Building 1.5.76-osd

After checking out the source code from our subversion server by

svn checkout http://pfanne.rzg.mpg.de/svn/RZG-AFS/trunk/afs_kerberos/openafs-1.5-osd/

configure has the additional options:

--enable-object-storage	enable use of objects storage for AFS files
--enable-vicep-access	enable direct client access to visible fileserver and OSD partitions
--enable-hpss-hsm	enable use of HPSS as HSM system for object storage
--with-hpss-path=path	where include and lib for HPSS can be found, typically /opt/hpss
--enable-dcache-hsm	enable use of DCACHE as HSM system for object storage
--with-dcache-path=path	where include and lib for DCACHE can be found

Questions or comments?

Thank you