

CASTOR 2.1.15 Development Plans

Giuseppe Lo Presti

on behalf of the CASTOR dev team

CERN/RAL Video Conf, 12/06/2014

- Stager refactoring
- Ceph and its integration in CASTOR
- Xroot as the main internal protocol
- Movers (protocols) handling
- Tape servers
- SL(C) 5/6 support



- Up until 2.1.14, the stager daemon goes to the DB multiple times per file request
 - Using C++ legacy framework
 - Lots of code that can be simplified
- In 2.1.15
 - *stagerd* goes to DB the minimum number of times
 - All user requests handling logic is PL/SQL
 - *stagerjob* is entirely dropped, along with a number of internal requests
 - File close happens in the Stager DB,
using the DB Link to the Nameserver DB

- Advantage: (much) less code, stepping stone to transparently introduce **DataPools**
- **In the plans:** support different nbReplicas on the same shared DiskPool according to the different service classes
 - Again the simplified code allows a more straightforward implementation
 - Not so straightforward in current 2.1.14 branch, therefore the proposal is to NOT backport this

- *See Sebastien's slides*



- New interface between movers and the scheduling system (*diskmanagerd*):
 - At transfer starting, *diskmanagerd* spawns the appropriate mover with its environment
 - No *stagerjob* needed any longer
 - At transfer closing, all movers synchronously connect to *diskmanagerd* to pass the metadata to close the file
 - Uniformly across all {*rfio*, *xroot*, *gsiftp*} movers
 - *Xroot* is not special (any longer)
- On the way, dropped support for updates

- CASTOR plugin ported to Xroot 4
 - Lots of code refactoring on the way
 - 3rd party copy custom implementation dropped
 - Use of standard **xrdcp --tpc** required
 - Ability to catch client disconnections and notify diskmanagerd about those failed transfers
 - Removes the need for configuring a long (**hours**) timeout for xroot transfers – reducing the chance of DoS...
 - **A timeout will still exist for the start of the transfer**, because of the intrinsic architecture of xroot, i.e. the inability to call back the clients when a slot is ready

- A new daemon called *tapeserverd* will eventually replace the legacy tape-server daemons *tapebridged*, *rtcpd* and *taped*
- The new *tapeserverd* daemon will be compatible with the *cupvd*, *tapegatewayd*, *vdqmd* and *vmgrd* protocols
- Early releases of 2.1.15 will provide both the legacy tape-server daemons and the new *tapeserverd* daemon
 - The *tapeserverd* daemon of early 2.1.15 releases will NOT be production ready
- The log messages of *tapeserverd* will be different from those of *rtcpd* and *taped*
- Both the legacy *taped* and new *tapeserverd* daemons will have a hardcoded initial state of DOWN

- Both the legacy rtcpd and new tapeserverd daemons will ONLY talk xroot (cf. Sebastien)
 - An instance of the xrootd daemon MUST be run on each and every tape server
- An instance of the rmcd daemon MUST be run on each and every tape server
 - The rmcd daemon of early releases of 2.1.15 will support SCSI-based tape-libraries such as IBM
 - The rmcd daemon of later releases of 2.1.15 will also support Oracle tape-libraries

- The bold statement is:
Support for SL5 is deprecated in v2.1.15
- The reality:
 - Head nodes software will only support SL6
 - Tape servers software will only support SL6
 - Disk servers will be partially supported in SL5 and fully in SL6, provided python 2.6 is installed
 - A python26 RPM has been ported to SL5
 - Ceph is not supported in SL5





DSS

Wishlist (a.k.a. beyond 2.1.15)

- VDQM/VMGR merge and rewrite
- Namespace split, CUPV refactoring
- SRM
- Secure Request Handler
- . . .

(More) Questions?