



CASTOR and SRM Databases

Giuseppe Lo Presti (IT/DM)





DM

CASTOR and SRM2 DBs

- Overall smooth operations...
- ...But some cases to highlight, with some lessons learnt:
 - Castor nameserver
 - Castor stager, May 9
 - SRM2 “extreme deadlocking”, May 13
 - SRM2 blockage, May 24
- Gained experience





Castor nameserver: backup

- Castor has introduced a bulk query in the nameserver
 - Implements “loose” synchronization between stagers/diskservers and namespace
 - Needs to scan a large number of entries in one go, so read intensive
- Nameserver slowed down during database backup times
 - Fixed by detecting long standing activity in the db from internal Oracle tables and suspending the synchronization
- Still performance problems...
 - Finally fixed by defragmenting db filesystems
 - ...At least, we didn't see slowdowns anymore...
 - But we still suspend the synchronization during backups

DM Castor: exec plans (un)stability

- stager_rm was very slow under a particular case
 - File already dropped in the namespace, only disk cache to be eventually cleaned up
 - Triggered by srmRm
- Tracked down to a (very simple) query doing **Full Table Scans of the CastorFile table, O(1M) rows**
 - Fixed by forcing index usage via hint
- Rest of the service almost unaffected, only some putDone requests were slowed down
 - Handled by the same thread pool which handles stager_rm requests



DM

SRM2 “extreme deadlocking”

- Triggered by srmAbortRequest requests interfering with normal request processing
 - Indeed, it may happen...
 - Root cause of the deadlock understood and being fixed for next version of Castor SRM
- The episode was worsened by a general context of high-load, in particular for the srm-atlas endpoint
 - Requests were timing out, more abort requests were coming
 - Database suffered from too many open connections and started swapping
- ...Giving first symptoms of what would have happened again soon without reducing #connections



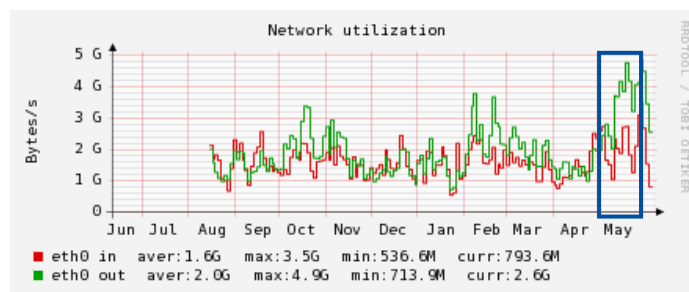
DM

SRM2 blockage on May 24

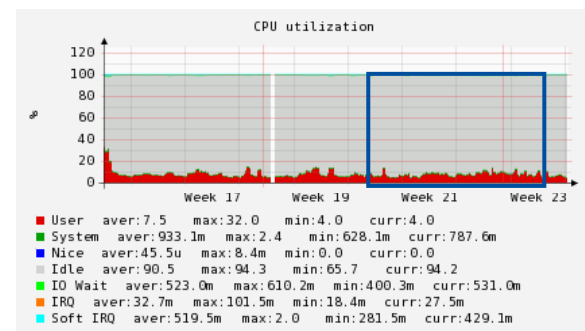
- Triggered by Castor public instance slowness
 - Not yet fully understood, see:
<https://twiki.cern.ch/twiki/bin/view/FIOgroup/PostMortemMay24>
- DTEAM/OPS requests started to pileup in all endpoints as they are served by castorpublic
- Too many db connections opened by the SRM frontends made the db server respond very badly
 - Each thread in the SRM frontends holds a db connection **and** is a Castor client
- This degraded even more the whole SRM2 service
 - All endpoints share(d) same database backend
- Actions taken from the db perspective:
 - Lowered maximum #connections accepted by Oracle
 - Lowered number of SRM threads (== client connections)
 - In the process of splitting db into separate RAC clusters



- No extra/special tuning has been needed at database level
 - At least, no more than during “normal” operation
 - Castor service @ CERN runs with a significant and constant load since several months
 - Continuous collaboration with IT/DES to keep improving database performances



Castor instances overall I/O throughput



ATLAS Stager DB during Apr and May

Gained experience on SRM2

- The CCRCs were the first production-scale experiences
- Opportunity for monitoring/tuning up db performances in collaboration with IT/DES
 - New deadlocks understood and being fixed
 - New indexes added
 - Better internal garbage collection procedure under development
 - ...not mentioning basic parameters like number of allowed connections

