


New stager architecture and operation

CASTOR external operation meeting
PIC Barcelona
19/11/2004
Olof Bärring, CERN-IT


Outline

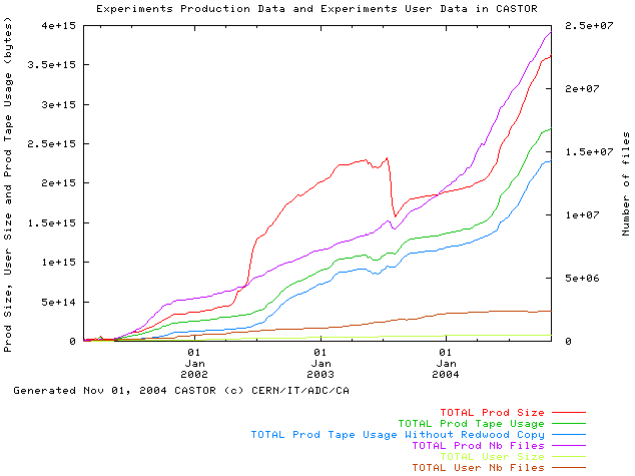
- CASTOR @ CERN
- Features in 2004
- New stager
  - Planning and progress
  - Delays
  - Architecture
  - Operational aspect
- Conclusions


Usage at CERN
New features in 2004

CASTOR @ CERN

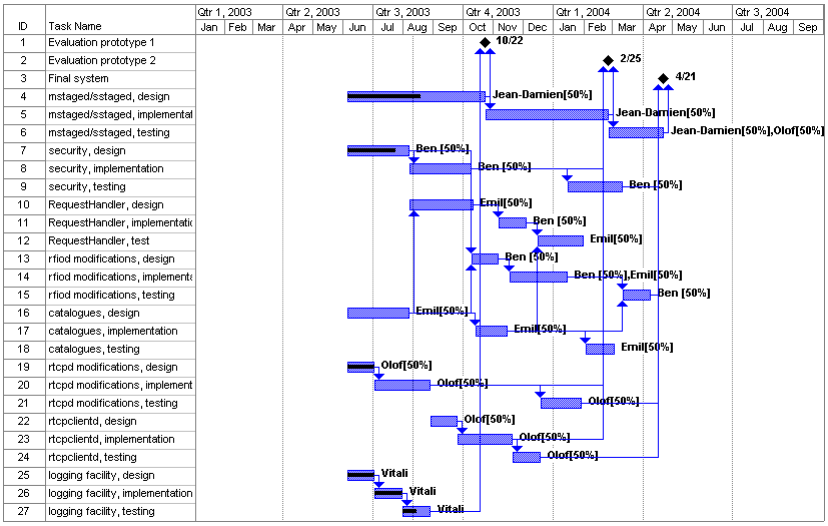
- Usage at CERN
  - ~3.4 PB data
  - ~26 million files
- New features deployed in 2004
  - Repack in production (since 2003): >1PB of data repacked from STK 9940A → 9940B media
  - Tape segments checksum calculation and verification is in production since March 2004
    - All new file segments are check-summed when written to tape
    - Checksum is stored in the name server database
    - Checksum is calculated and verified each time the segment is staged back to disk
    - For CASTOR files written before March 2004, the checksum row is populated in the name server as the files are staged to disk for the first time
  - Sysreq/TMS definitely gone in July
  - VDQM prioritize tape write over read deployed in September
    - Migration requests are usually long and well-behaved
    - Recalls usually chaotic and inefficient
    - By prioritizing migration requests, there is no need anymore to dedicate drives to CDR (Central Data Recording)
    - VDQM prioritization can be switched off through simple configuration (/etc/shift.conf)
  - To come soon: vmgr_gettape() with weighted device group selection developed by Paco
- Problems seen in 2004
  - During 2004 some experiments hit stager catalogue limitation (~200k files) beyond which the stager response can be very slow

CASTOR@CERN evolution

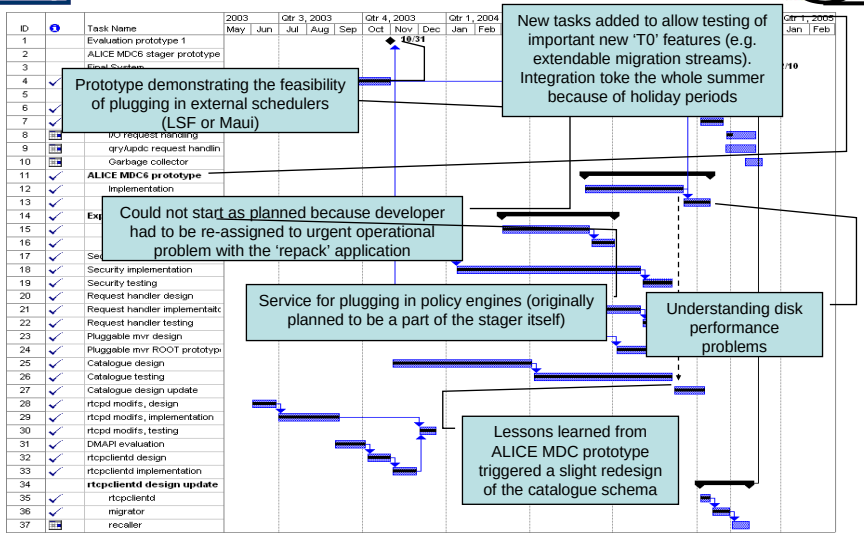


New stager development Planning & progress

New stager developments
Original plan



New stager developments
actual task workflows




New stager developments
Principal reasons for delays

New stager developments delay
Main reason: The "repack problem"

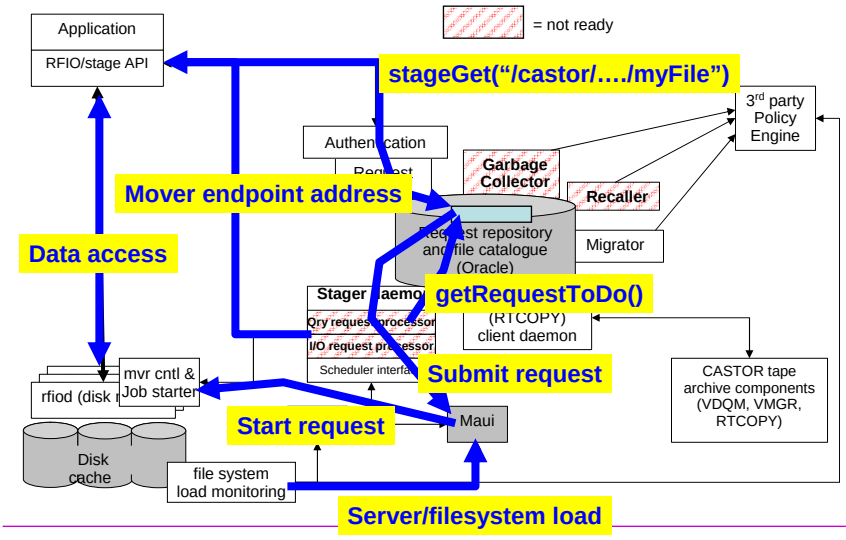
- Repack: standard HSM utility to recover tape media:
  - 'Holes' created because of deleted files
  - Migration to higher capacity media
- A test version of the CASTOR repack utility was released in April 2003
  - Tested during summer for repacking CASTOR log files and other CASTOR operation files
  - Tests OK, started with some (mostly inactive) user files in September
- End November 2003: bug detected
  - Bug found in stager API during the certification of first production release of repack
  - The effect was that a fraction (~5%) of the repacked files got wrongly mapped in the CASTOR name server
- December 2003 – May 2004
  - One CASTOR developer working full time on finding and repairing incorrectly mapped CASTOR files
  - A bit less than 50,000 files wrongly mapped out of >1 million
  - Repair applied to the CASTOR name server the 26th of April 2004
  - Affected users (L3C) were informed about the problem

New stager developments delays Unplanned grid activities

- SRM interoperability
  - Drilling down the GSI (non-)interoperability details
  - Holes in the SRM specs
  - Time-zone difference (FNAL-CERN) does not favor efficient debugging of interoperability problems
- Other grid activities: CASTOR as a disk pool manager without tape archive
  - We provided a packaged solution for LCG
  - But… support expectations pointed towards a development sidetrack
    - Castor is not well suited for such configurations
  - Decided to drop all support for CASTOR disk-only configurations and focus on the CERN T0/T1 requirements

New stager Architecture

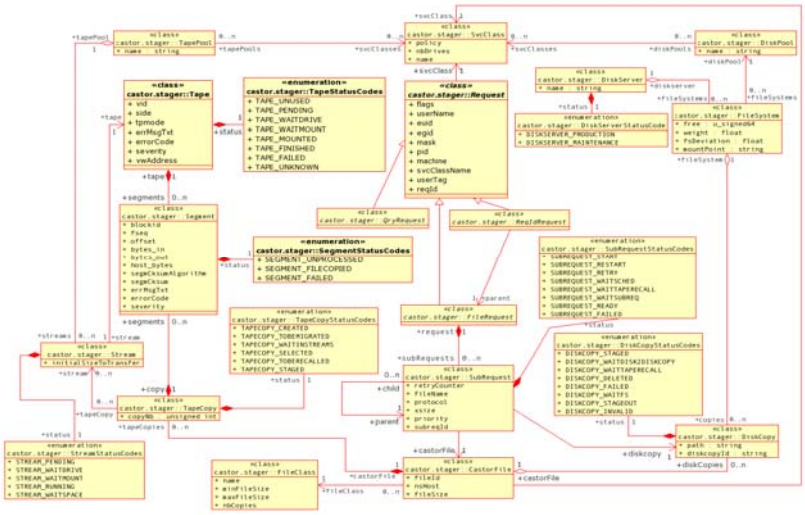
New stager architecture



New stager design

- Database centric architecture
  - "Surrounding" daemons are stateless
  - Important operational decisions can be translated into simple SQL statements
    - Preparation of migration or recall streams
    - Weighting of file systems used for migration/recall
    - Draining of disk servers or file systems
- Disk access is scheduled
  - All rfio_open()/stagein/stageout requests are scheduled (Maui or LSF)
  - Advanced scheduling features for 'free' (e.g. fair-share)
- Minimal footprint of inactive requests
  - Requests are not instantiated in terms of processes until they run
    - Stored in DB and/or scheduler while waiting for resources
    - Number of migrator/recaller instances ≤#drives (no process instances while waiting for drive)
- Dynamic migration/recall streams
  - Multiple concurrent requests for same volume will be processed together
  - New requests arriving after the stream has started are automatically added to the stream

Catalogue DB schema



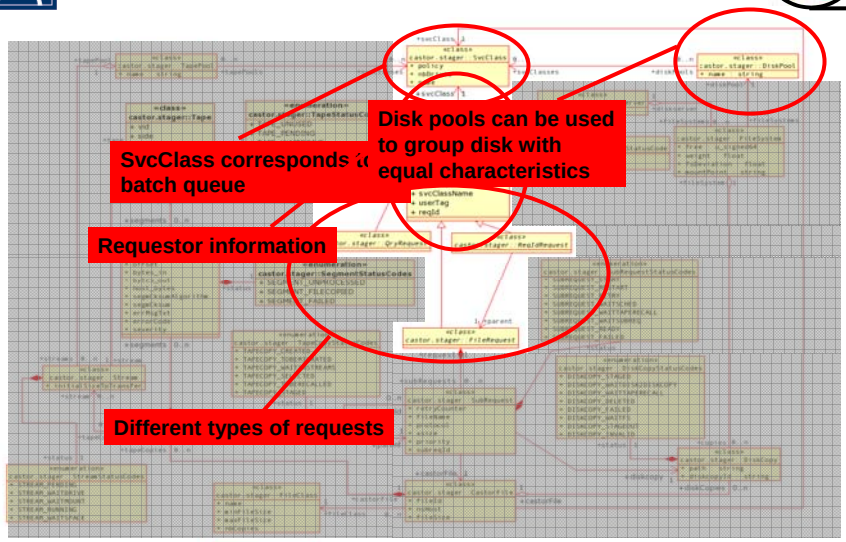
Operational aspects

- Externalized decisions
  - Request scheduling
  - Migration/recall/GC
  - File system selection
  - Tape status
- Problem tracing
  - Distributed Logging Facility (DLF)
  - DLF GUI
- Castor Monitoring
  - Cmonitd
  - GUI

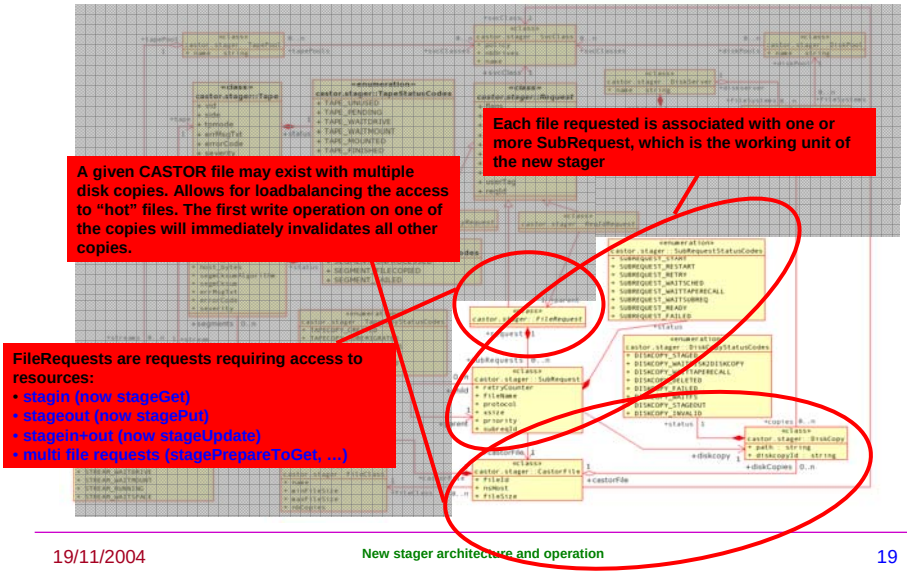
Request scheduling

- Request scheduling
  - All rfio_open() (stagein/out) requests are scheduled using an external scheduler
    - LSF or Maui supported
    - Maui is the best tested for the moment
  - Scheduler configuration determines how user requests should be dispatch to the disk server
    - In the simplest case a scheduler queue ≈ disk pool
    - However more complex configurations with disk pools shared among queues are possible
    - Scheduler configuration can be used for setting the resource shares between different users and groups

Request information



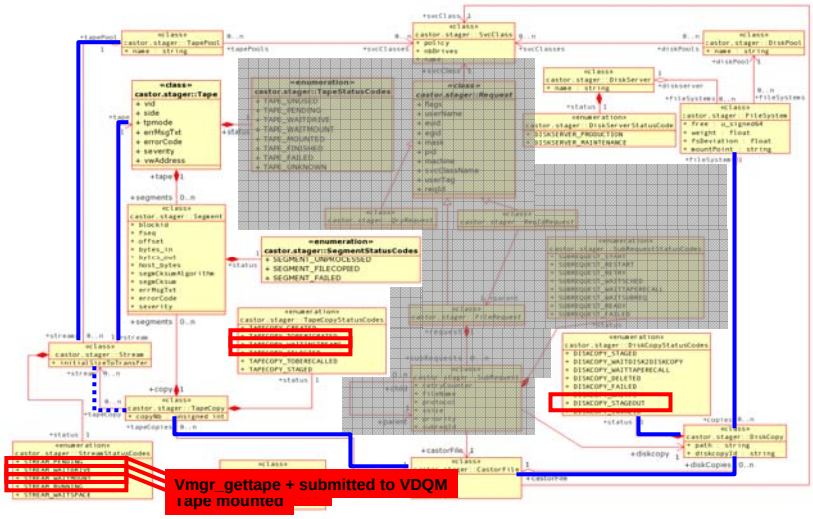
SubRequest information



Migration

- The decision which file to migrate and when is entirely externalized
  - Files to be migrated are linked to one or more "Streams"
    - Stream is a container of files to be migrated
    - By default a given migration candidate is associated to as many streams as it is defined by the maxdrives attribute of the fileclass
    - A running Stream is associated with a Tape. However, the same Stream may survive several Tapes.
    - The Stream is destroyed when there are no more migration candidates associated with it
    - If several Streams are associated with a given migration candidate, it will be picked up by the fastest
  - Stream creation and linking of migration candidates to streams are pure DB operations
    - Could be performed directly with a SQL script
    - Several scripts for different policies can work concurrently
- New stager will *NOT* segment files

Migration candidate



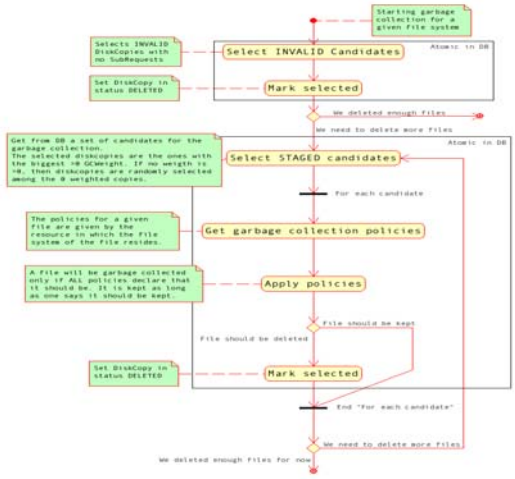
Recall

- Recall differs from Migration in that it is usually executed on demand
  - An active request is waiting for the file to be recalled
- However, with the new architecture the decision what to recall and when is externalized
  - A recall candidate is always associated with tape Segments (usually only one)
  - Like for the Migration the association with tape Segment can be performed from an SQL script but the information must be taken from the name server
  - In particular you may choose to wait with creating the tape Segment association in order to see if there are other recalls for the same tape

Garbage collection

- Like for migration/recall a disk file garbage collection is triggered via a DB update
  - A GCWeight attribute of the disk copy is provided for externally setting its weight to be used when compared with other candidates
    - This could be based on experiment policies: e.g. all files beginning with "ABC" should be given low weight for removal
    - By default all weights are zero

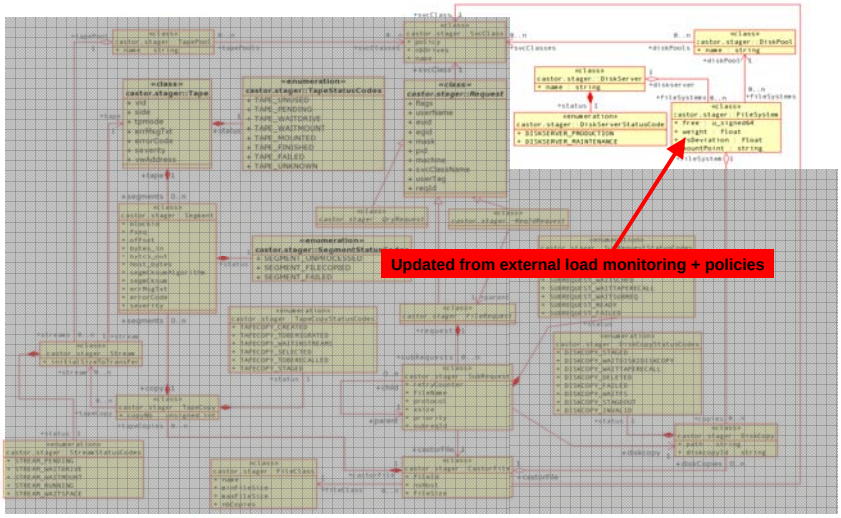
Garbage collection



File system selection

- The file system selection is called from several places
  - When scheduling access for a given client request
  - When selecting the best migration candidate
  - When selecting a file system for recalling a tape file
- The FileSystem table has several attributes updated by external policies based on load and status monitoring
  - "free" is the free space on the file system
  - "weight" reflects the current load calculated using an associated policy
  - "fsDeviation" is the deviation to be deduced from the weight every time a new stream is added to the file system. This assures that the same file system is not selected twice

Request information



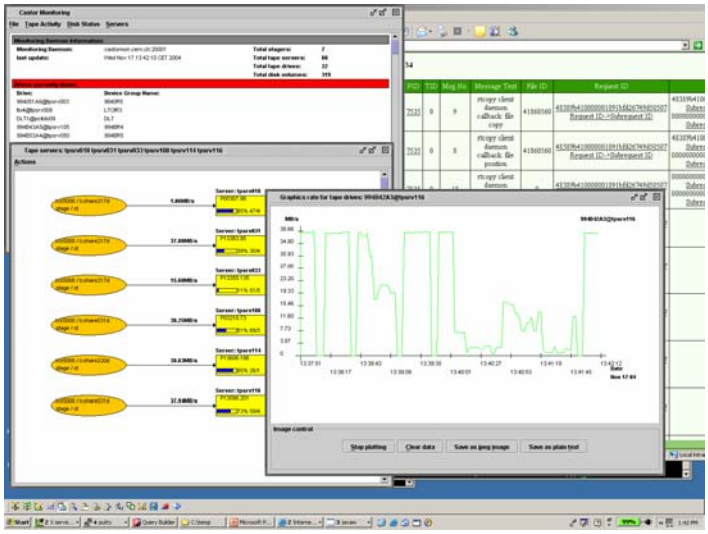
Logging facility

- All new castor components use the Distributed Logging Facility (DLF)
  - Log to files and/or database (Oracle or MySQL)
  - Web based GUI for problem tracing using the DLF database

Instant performance views

- Cmonitd has been part of CASTOR since 2002
  - Central daemon collecting UDP messages from
    - Stagers
    - Tape movers (rtcpd)
    - Tape daemon (mount/unmount)
  - Original GUI written in Python
  - GUI rewritten in Java (swing), September '04
    - Web start
    - Drive performance time-series plots

Monitoring GUI



New stager developments Deployment issues

- Security issues
  - All CASTOR services are technically prepared for strong authentication
    - http://cern.ch/castor/DOCUMENTATION/CODE/SECURITY/CASTOR_Security_Implementation.pdf
    - Kerberos-4, 5 and GSI supported
  - CASTOR security plug-ins used by other projects (LCG, EGEE)
  - A number of deployment issues remain:
    - Kerberos-5 infrastructure not yet in place
    - Batch job clients must have appropriate credentials
    - No solution yet for windows clients
    - Management of CASTOR service keys
  - Propose to do first deployment without strong authentication and upgrade when all infrastructure issues are solved
- Packaging
  - New packaging model envisaged:
    - One RPM for each CASTOR client and server
      - rfio
      - Stage
      - Nameserver
      - VMGR
      - …
    - One RPM for libraries
    - One 'devel' RPM (include files, man-pages)
- It will be possible to import disk servers from current to the new stager without having to re-stage the files

Conclusions

- CASTOR production status is OK
  - Important new features in 2004:
    - Checksum calculation/verification in production
    - Tape mover with all necessary features needed by new stager is running in production since March
    - VDQM prioritization of tape write since September
    - Vmgr weighted tape selection according to device group "load"
  - But, for the first time some experiments have hit the limitations of the current stager
- New stager developments
  - Important delays mainly due to high priority investigation and cleanup of repack problem
  - Database centric and stateless daemons
  - Externalized decisions for scheduling, migration, recall, garbage collection, file system selection
  - Prototype for ALICE being stress tested