Introduction and Overview

Olof Bärring
CERN / IT

Castor Readiness Review – June 2006

Outline

- Scope and constraints
- History
  - SHIFT: users managing tapes
  - CASTOR1: managed storage
  - CASTOR2: scalable disk cache
  - CASTOR outside CERN
- Main requirements driving CASTOR1 → CASTOR2
  - Data recording and reconstruction
  - Data analysis: what was known at the time
- New requirements
  - Grid
  - Data analysis: what changed with time
- Conclusions

Outline

- Scope and constraints
- History
  - SHIFT: users managing tapes
  - CASTOR1: managed storage
  - CASTOR2: scalable disk cache
  - CASTOR outside CERN
- Main requirements driving CASTOR1 → CASTOR2
  - Data recording and reconstruction
  - Data analysis: what was known at the time
- New requirements
  - Grid
  - Data analysis: what changed with time
- Conclusions

Scope and constraints

- Provide a managed storage service for all physics data at CERN
  - Transparent tape media management
  - Automated disk cache management
  - Unique global name space
- Assure that CERN can fulfill the Tier-0 and Tier-1storage requirements for the LHC experiments
  - Central Data Recording (CDR)
  - Data reconstruction
  - Data export to Tier-1 centers
- Strive to meet the CERN Tier-2 requirements
  - The exact requirements and scale are unknown but CASTOR should prove to integrate well with other services that may be part of an analysis facility
    - E.g. xrootd interface now requested
- CASTOR2 assumes backend mass-storage
  - Not designed to work as a stand-alone disk cache solution for Tier-2 institutes

Outline

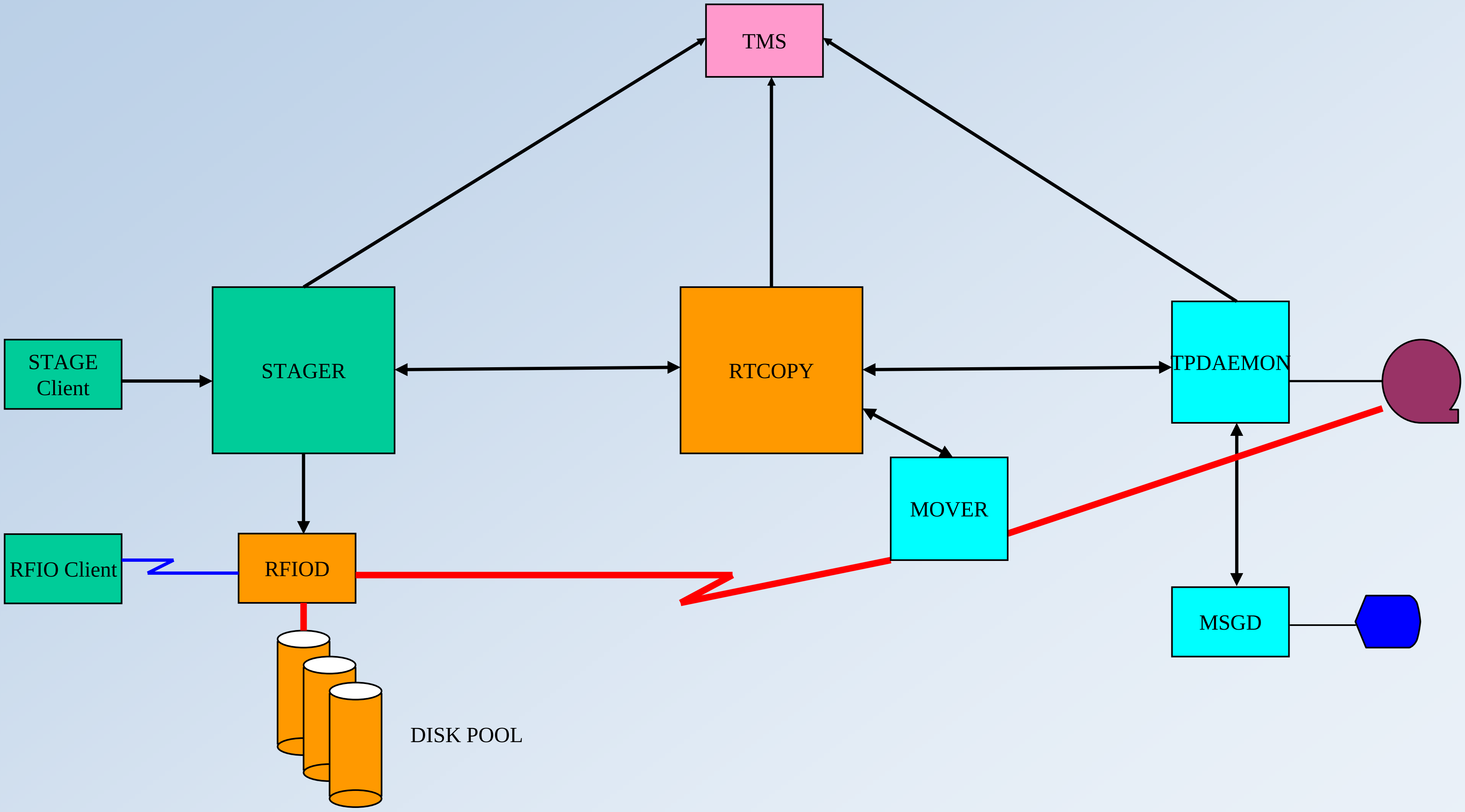
- Scope and constraints
- History
  - SHIFT: users managing tapes
  - CASTOR1: managed storage
  - CASTOR2: scalable disk cache
  - CASTOR outside CERN
- Main requirements driving CASTOR1 → CASTOR2
  - Data recording and reconstruction
  - Data analysis: what was known at the time
- New requirements
  - Grid
  - Data analysis: what changed with time
- Conclusions

SHIFT

- **S**calable **H**eterogeneous **I**ntegrated **F**acili**T**y
  - Started as a project (HOPE) between the OPAL experiment and the CN division in early 90's
  - Main authors: Jean-Philippe Baud, Fabrizio Cane, Frederic Hemmer, Eric Jagel, Ashok Kumar, Gordon Lee, Les Robertson, Ben Segal, Antoine Trannoy
- All user file access on disk. No direct tape access
  - The idea of tape staging at CERN dates back to the early 70's (CDC7600)
- Components
  - Stager daemon (stgdaemon) managing the disk pool
  - Remote Tape Copy (RTCOPY)
  - Remote File IO (RFIO)
  - Tape allocation and control (tpdaemon)
    - Label processing, load, unload, positioning
    - Operator interface for manual mounting
    - Interface to robotic mounting
  - Tape Management System (TMS)
    - Logical volume repository
- Users access files by Tape volume (VID) + tape file sequence number (FSEQ) → flat namespace: stagein –V EK1234 –q 35 …
  - The experiments normally had their own catalogue on top (e.g. FATMEN)

- CERN was awarded the 21st Century Achievement Award by Computerworld in 2001

SHIFT architecture



SHIFT limitations

- Data rate: more than 10MB/s per stream is difficult to achieve
- Stager catalog does not really scale over 10,000 files
- SHIFT does not support many concurrent accesses
- No automatic allocation of tapes
- No easy access to files by name without an external package like fatmen
- HSM (automatic migration/recall of files) is not available

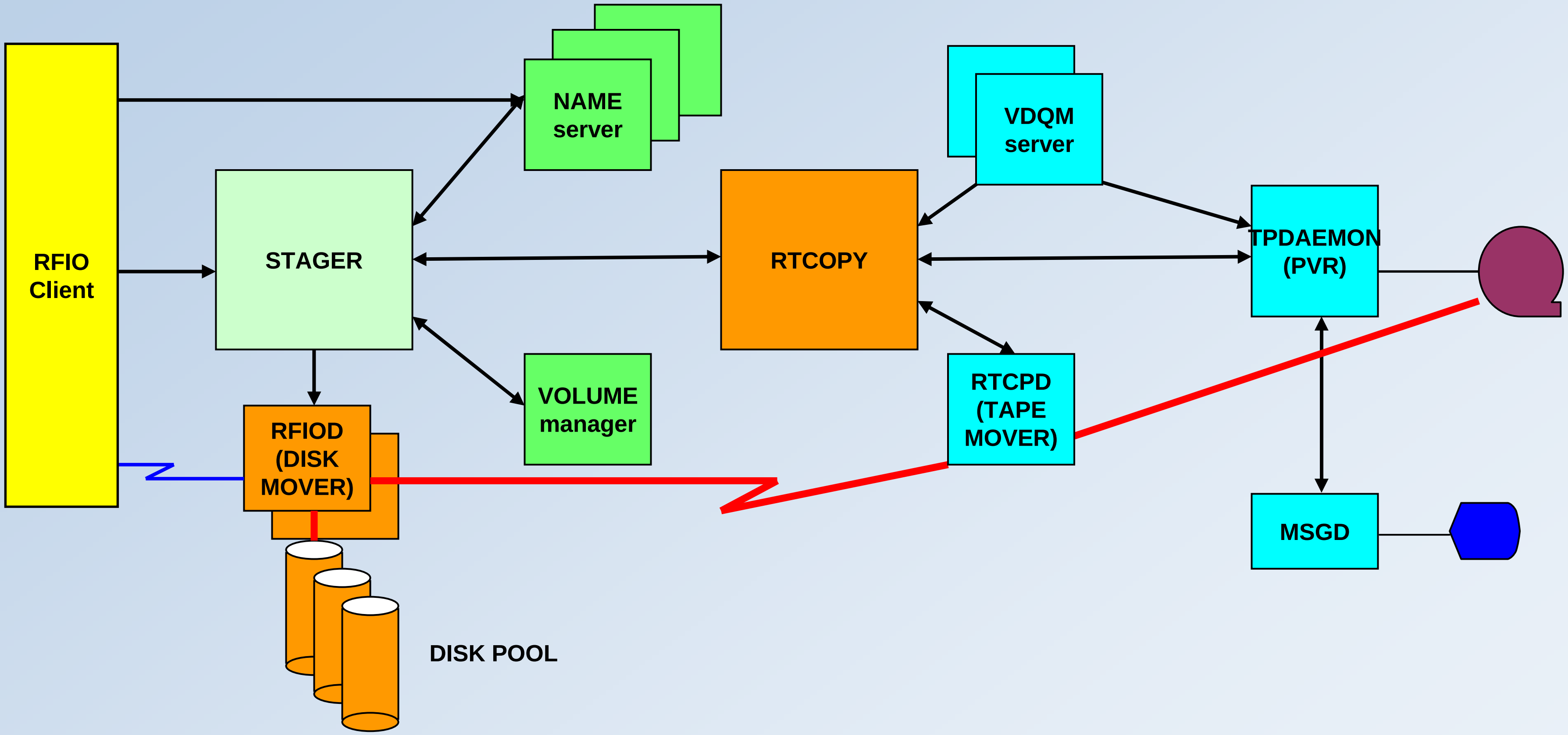
Alternative solutions

- Since the mid-90's CERN had been looking and testing alternative solutions to SHIFT
  - OSM
  - ADSM (now TSM):
  - HPSS
  - Eurostore
- Only HPSS was run in production for 3 years (1998 – 2001)

CASTOR1

- Cern Advanced STORage manager
  - Project started in 1999 to address the immediate needs of NA48 and COMPASS but also provide a base that would scale to meet the LHC requirements
    - SHIFT had proven to not scale in terms of data rates and request handling (see SHIFT limitations listed earlier)
  - Managed storage: tapes hidden from the users
  - Main authors: Jean-Philippe Baud, Fabien Collin, Jean-Damien Durand, Olof Bärring
- Components
  - Stager daemon: enhancements to the SHIFT stgdaemon
    - home-written database backend (cdb)
    - Automatic tape migration added
    - Support for castor name server
  - Multithreaded Remote Tape Copy (RTCOPY): completely rewritten but backward compatible with SHIFT
  - Remote File IO (RFIO): same as SHIFT but with support for data streaming
  - Tape allocation and control (tpdaemon): same as SHIFT
  - Volume Manager (VMGR): replaces TMS in SHIFT
  - Name server (Cns): provides the CASTOR name space
  - Tape Volume and Drive Queue service (VDQM)
  - Tape repack: automated media migration
- Users access file by their CASTOR file names → hierarchical UNIX directory namespace, e.g. stagein –M /castor/cern.ch/user/…

CASTOR1 architecture



CASTOR1 limitations

- Stager catalogue limitations:
  - Stager unresponsive beyond 200k disk resident files
- Tape migration/recall not optimal
  - Migration streams are statically formed and ordering among files cannot be changed depending on the load picture
  - Tape recalls request for same tape are not automatically bundled together
  - Large footprint from waiting requests
- No true request scheduling
  - Throttling, load-balancing
  - Fair-share
- Operational issues
  - No unique request identifiers
  - Problem tracing difficult
- Stager code had reached a state where it had become practically un-maintainable
  - Based on the >10 years old SHIFT stager with a long history of patches, hacks and add-ons for CASTOR1

CASTOR2

- The original CASTOR plans from 1999 contained a development of a new stager
  - The re-use of the SHIFT stager was a temporary solution
- Project proposed at CASTOR users' meeting, June 2003: develop a replacement for the CASTOR1 stager
  - The CASTOR1 stager had already proved to not scale to meet LHC requirements (see previous slide on CASTOR1 limitations)
  - Original authors: Ben Couturier, Jean-Damien Durand, Sebastien Ponce, Olof Bärring
- Components
  - Same as CASTOR1 except for the stager, which is now split up into several services
- Users access file by their CASTOR file names → hierarchical UNIX directory namespace, e.g. stager_get –M /castor/cern.ch/user/…

CASTOR2 vision...
(from users meeting 2003)

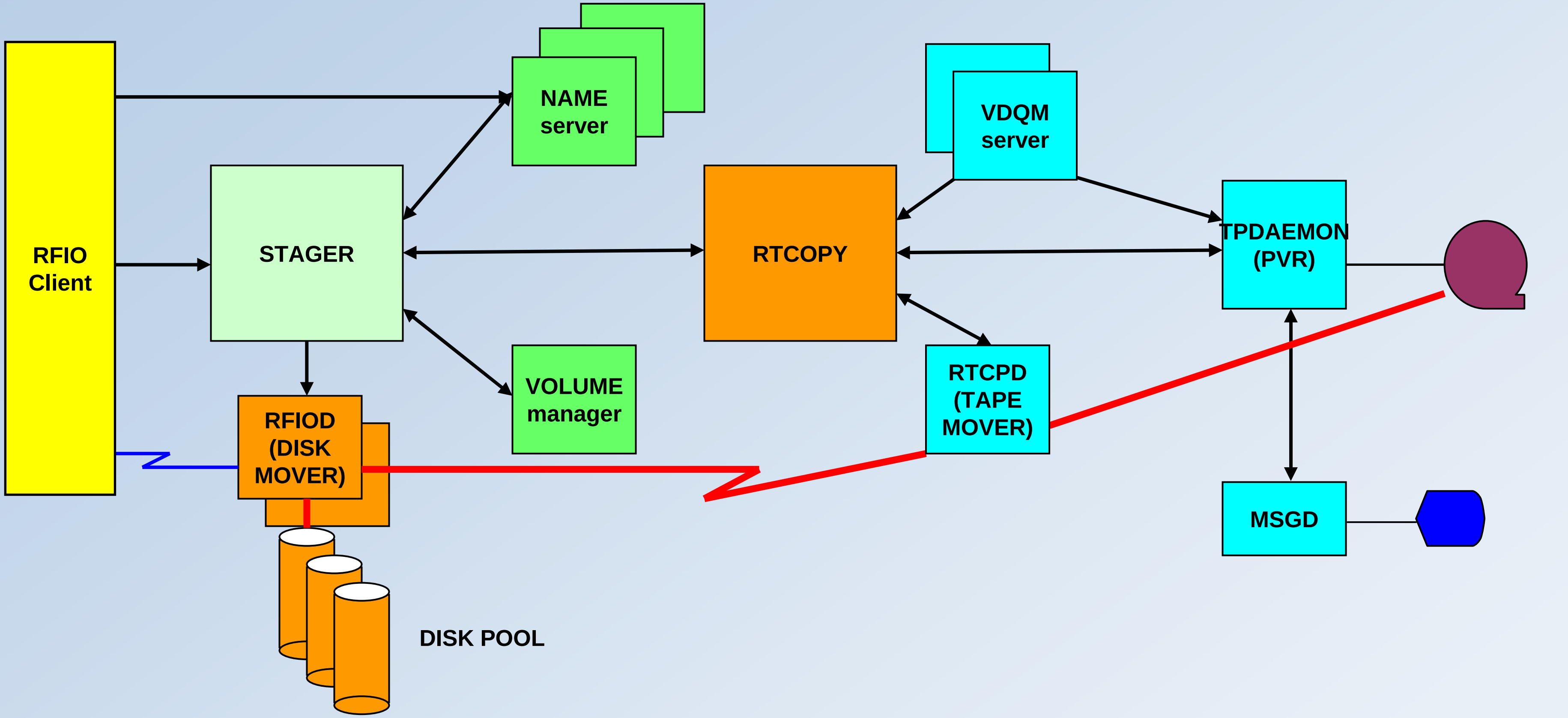
- With clusters of 100s of disk and tape servers, the automated storage management faces more and more the same problems as CPU clusters management
  - (Storage) Resource management
  - (Storage) Resource sharing
  - (Storage) Request scheduling
  - Configuration
  - Monitoring
- The stager is the main gateway to all resources managed by CASTOR

Vision: **Storage Resource Sharing Facility**

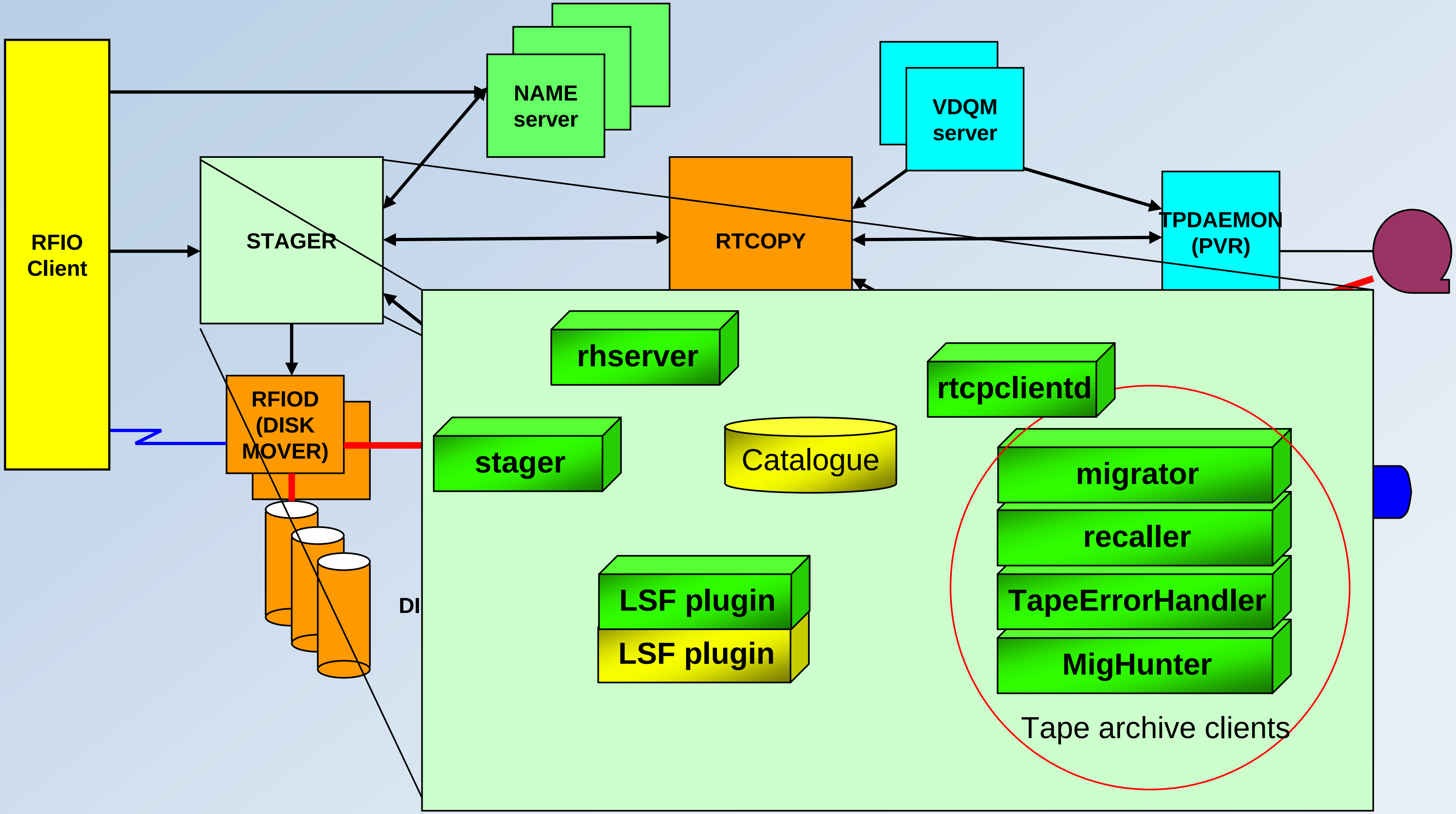
CASTOR2 Proposal
(from users meeting 2003)

- Pluggable framework rather than total solution
  - True request scheduling: delegate the scheduling to a pluggable black-box scheduler. Possibly using third party schedulers, e.g. Maui or LSF
  - Policy attributes: externalize policy engines governing the resource matchmaking. Start with today's policies for *file system selection, GC, migration, ...* Could move toward full-fledged policy languages, e.g. implemented using "GUILE"
- Restricted access to storage resources to achieve predictable load
  - No random rfiod eating up the resources behind the back of the scheduling system
- Disk server autonomy as far as possible
  - In charge of local resources: file system selection and execution of garbage collection
  - Loosing a server should not affect the rest of the system

CASTOR2 architecture



CASTOR2 architecture



CASTOR outside CERN

- SHIFT was used outside CERN at several HEP and non-HEP sites (e.g. Canadian fishing company)
  - Truly lightweight with no external dependency
- CASTOR1 was sufficiently lightweight in terms of deployment to be run by small sites
  - Dependency on externally provided database software but opensource alternative was supported (MySQL)
  - Not an option with CASTOR2
    - Strong dependency on commercial components (Oracle, LSF)
    - Not supported for disk cache only installations
  - Recommendation is to use dCache or dpm
- A subset of the LCG Tier-1 institutes use CASTOR
  - CNAF, PIC, ASCG: CASTOR1 in production and now also migrating to CASTOR2
  - RAL is about to start with CASTOR2 in production
- Support model
  - Data management is mission critical for Tier-1 centers
    - Best effort support not appropriate
  - Require one person to participate in CASTOR developments for some period
    - Build up local expertise with good knowledge of the software
    - CNAF, RAL and PIC have all contributed

Outline

- Scope and constraints
- History
  - SHIFT: users managing tapes
  - CASTOR1: managed storage
  - CASTOR2: scalable disk cache
  - CASTOR outside CERN
- Main requirements driving CASTOR1 → CASTOR2
  - Data recording and reconstruction
  - Data analysis: what was known at the time
- New requirements
  - Grid
  - Data analysis: what changed with time
- Conclusions

Data recording and reconstruction (1)

❖ LCG project proposal appendix 3 in http://lcg.web/lcg/PEB/Documents/c-e-2379Rev.final.doc (Sep 5 2001):

Summary of the approximate LHC computing requirements

- **General Parameter**
  - Size of a recorded p-p event: 1 MB (up to 40MB for a Pb-Pb event in ALICE)
  - Data taking rate: 102 (Hz), down from 109 Hz p-p collisions, after several trigger levels.
  - Recorded p-p events per day: 107 (out of 1014)
  - Data taking: 107 seconds/y, or ~116 days/y (except Ion runs for ALICE ~15 days/y).
  - Recorded p-p events per year: 109
- **Storage per experiment**
  - 3 to 10 PB on tape, total ~28 PB (with 2/3 more per year beyond), Raw Data storage is ~ 1/3 of this total.
  - 1 to 6 PB of disk, total ~11 PB (with 1/3 more per year beyond)
- **CPU (off-line) per Experiment**
  - Best guesses today range from ~1 M SI-95 in LHCb, to ~2 M SI-95 for each of ALICE, ATLAS and CMS. Uncertainties are at least a factor 2. Estimates are the sum of Tier0, all Tier1 and all Tier2.
- **Networking**
  - 6-12 Gbps between main centres (1.5-3 Gbps per experiment)

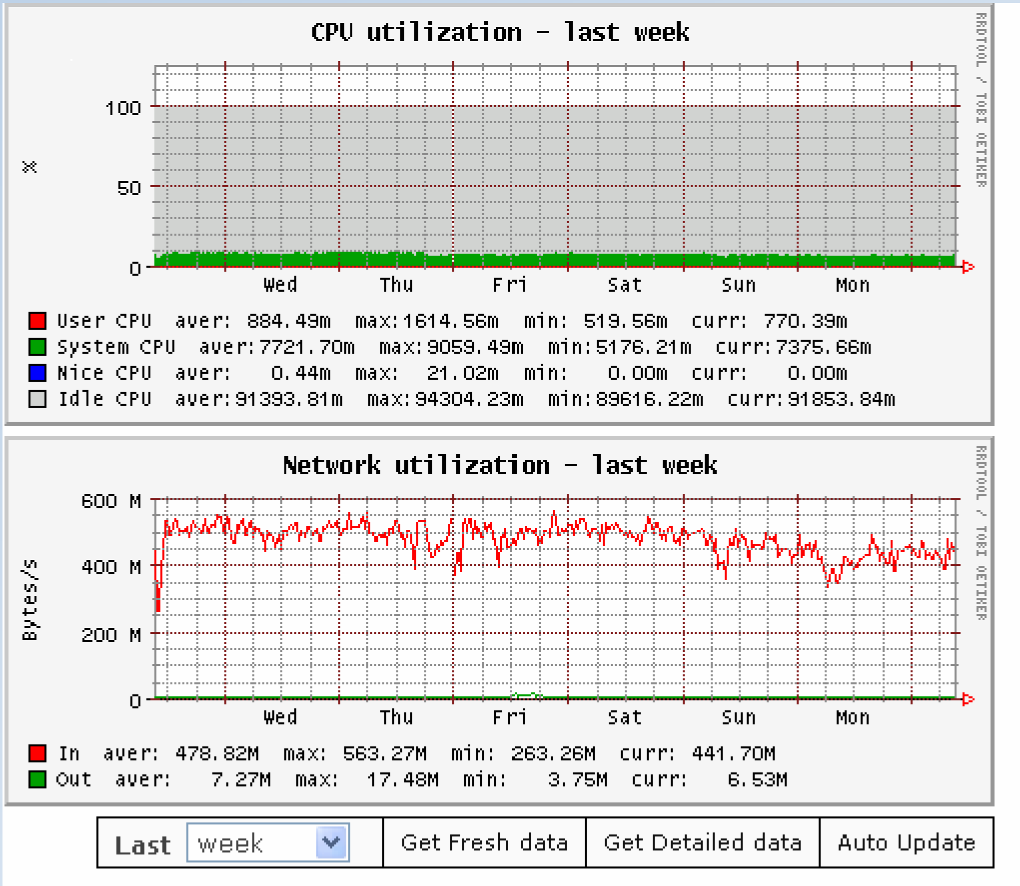
Data recording and reconstruction (2) ALICE Mock Data Challenges (MDC)

- MDC-1
  - March 1999
  - HPSS
  - Goal ???
  - Result: 14MB/s sustained for a week
- MDC-2
  - April 2000
  - HPSS and CASTOR1 prototype
  - Goal 100MB/s sustained for a week
  - Result: Failed to sustain a weeks operation
- MDC-3
  - 2001
  - CASTOR1
  - Goal: 100MB/s sustained for a week
  - Result: 85MB/s sustained for a week
- MDC-4
  - December 2002
  - CASTOR1
  - Goal: 200MB/s sustained for a week
  - Result: ~300MB/s sustained for a week
- MDC-5
  - January 2004
  - CASTOR1
  - Goal: 300MB/s sustained for a week
  - Result: Failed to sustain a weeks operation. 280MB/s for day
- MDC-6
  - March 2005
  - CASTOR2
  - Goal: 450MB/s sustained for a week
  - Result: 450MB/s sustained for a week
- MDC-7
  - July 2007?
  - CASTOR2
  - Goal: 1GB/s sustained for a week

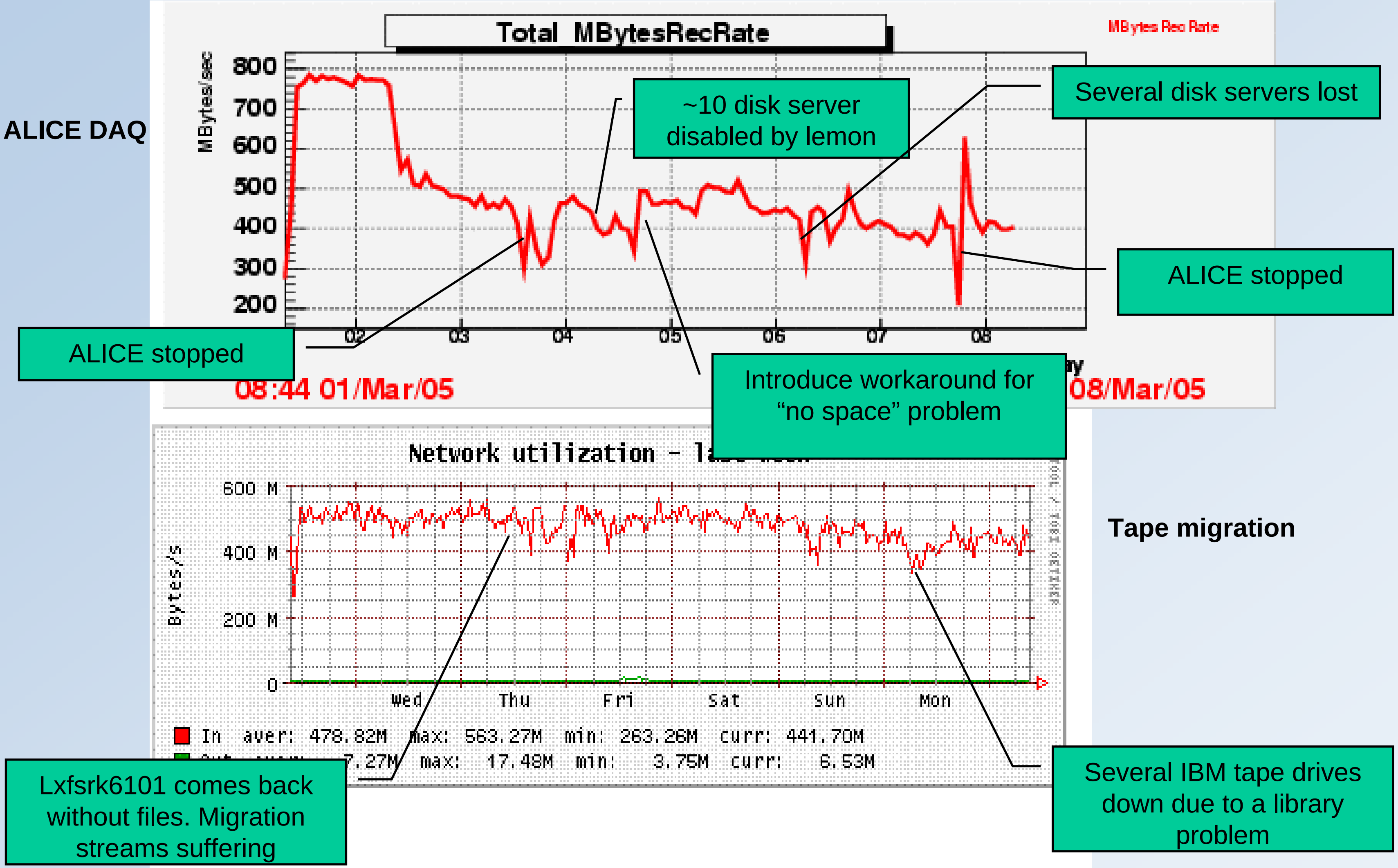
Data recording and reconstruction (3)

- The ALICE MDC 4 and 5 were in particular useful for understanding limitations in CASTOR1
  - Lack of robustness against hardware failures
    - Drive failure of one stream hold back all the others
    - Disk server failures could hang the stgdaemon or the tape migration
  - Lack of I/O stream scheduling → file-system contention
    - Static migration streams
      - All migration candidates and their internal ordering were decided before the migration started
    - No intput/output stream scheduling
      - Round-robin filesystem selection
  - Single threaded stgdaemon handling all requests
    - Even if child processes were forked for long lived requests such as migrations and garbage collection, all catalogue updates had to go through the stgdaemon
  - RFIO used for disk server houskeeping
    - Stgdaemon calls rfio for removing remote files, checking filesystem and file sizes
    - Stuck RFIO → stuck stgdaemon (although timeouts help)
- CASTOR2 addresses all those problems
  - Further tuning of policies is still required

ALICE MDC-6


CPU utilization - last week
%
100
50
0
Wed
Thu
Fri
Sat
Sun
Mon
User CPU aver: 884.49m max:1614.56m min: 519.56m curr: 770.39m
System CPU aver:7721.70m max:9059.49m min:5176.21m curr:7375.66m
Nice CPU aver: 0.44m max: 21.02m min: 0.00m curr: 0.00m
Idle CPU aver:91393.81m max:94304.23m min:89616.22m curr:91853.84m
Network utilization - last week
Bytes/s
600 M
400 M
200 M
0
Wed
Thu
Fri
Sat
Sun
Mon
In aver: 478.82M max: 563.27M min: 263.26M curr: 441.70M
Out aver: 7.27M max: 17.48M min: 3.75M curr: 6.53M
Last
week
Get Fresh data
Get Detailed data
Auto Update

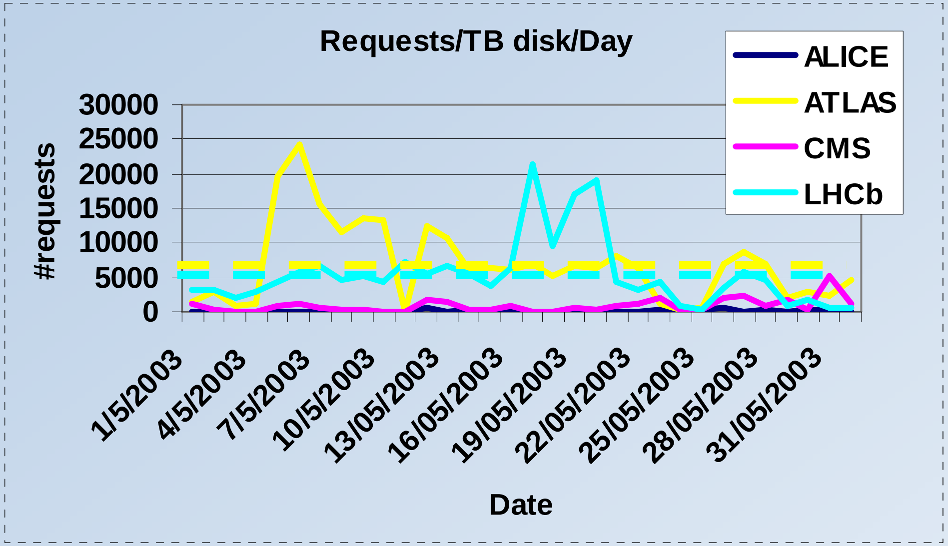
ALICE MDC-6 (details)



Data analysis
What was known at the time

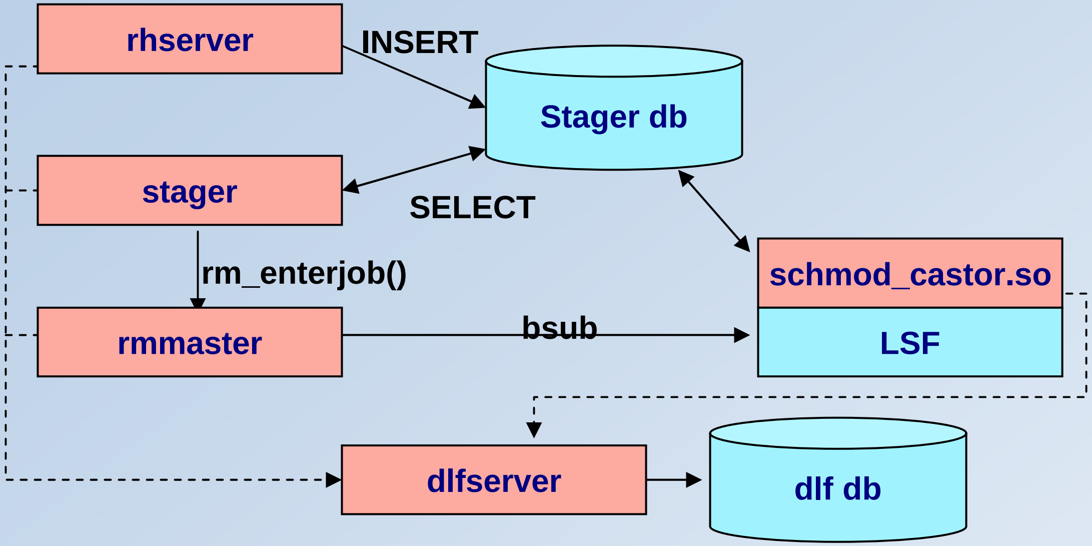
- Not very much…
- Only viable option: look at existing analysis workload and extrapolate
  - Assumption 1: workload scale linearly with the size of the disk cache
    - Normalize the request load on the LHC CASTOR1 stagers to the size of their disk pools used for physics analysis
  - Assumption 2: LHC disk pools ~PB size
- LHC experiments' CASTOR1 stager accounting data for one month (May 2003) was used to evaluate the current workload

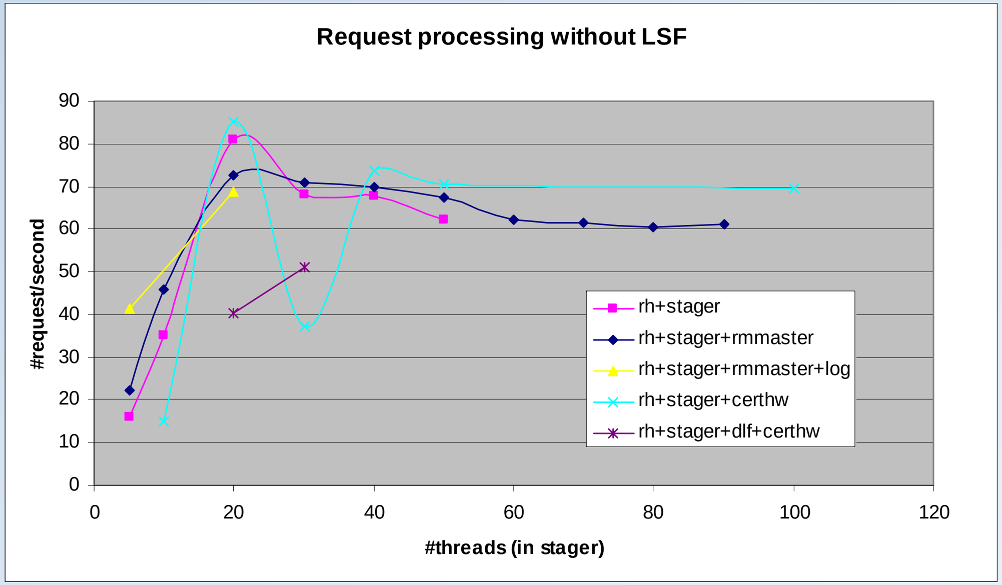
CASTOR1 activity



- ATLAS
  - Peak rate 25k reqs/TB/day = 0.3reqs/TB/s
  - Average rate 7k reqs/TB/day = 0.08reqs/TB/s
- Scale linearly to PB disk pool
  - Peak rate: 300 reqs/PB/s
    - Not yet met: CASTOR2 request handler has been measured to handle peaks up to 150 reqs/s
  - Average rate: 80 reqs/PB/s
    - Not yet met: CASTOR2 request processing has been measured to handle average rates up to 5-10 reqs/s. Bottleneck is in LSF CASTOR plugin

CASTOR2 timing without LSF





Outline

- Scope and constraints
- History
  - SHIFT: users managing tapes
  - CASTOR1: managed storage
  - CASTOR2: scalable disk cache
  - CASTOR outside CERN
- Main requirements driving CASTOR1 → CASTOR2
  - Data recording and reconstruction
  - Data analysis: what was known at the time
- New requirements
  - Grid
  - Data analysis: what changed with time
- Conclusions

Grid

- In parallel with the deployment of CASTOR2 there has been a growing emphasize on SRM requirements
  - SRM v1.1 has been in production for CASTOR1 since 2004
    - Stable but not very well defined interface
  - LCG SRM baseline WG decided in 2005 that v1.1 was not sufficient to meet the experiments' requirements. SRM v2.1 implementation was requested at a tight timescale
    - Much more complex interface than v1.1 but still not well defined
    - CASTOR implementation delivered by RAL
- Grid mapping & VOMS integration
  - Support for ACL in CASTOR name server
  - Extensions to the RFIO TURL definition to meet requirement for different storage types
- Tier-0 – Tier-x connectivity
  - Not really a CASTOR requirement but it is the place where the problem was discovered

Data analysis
What changed with time

- The time to open a file…
  - Still unclear what the exact requirement
  - A single file open in CASTOR2 takes ~0.5 – 1s
    - Limited by LSF scheduling and the CASTOR LSF plugin
    - Bulk processing allows for up to ~10 requests/s but it appears difficult to go much above without changing the CASTOR LSF plugin
- xrootd integration may help
  - Ongoing development by xrootd developer with some integration work on Castor side
    - Not yet agreed as production service; future maintenance of xrootd to be agreed (SLAC…)

Conclusions

- CASTOR2 is the latest development of in a 15 years history of storage software developments at CERN
  - SHIFT (1991 – 1999): tape software, rfio
  - CASTOR1 (1999 – 2006): name server, VDQM, VMGR, rtcopy
  - CASTOR2 (2006 -): stager components
- CASTOR2 is specifically designed to meet the LHC requirements for the CERN Tier0,1 disk cache management
  - Large (PB) disk caches
  - Tight stream control
  - Robustness against hardware failures
  - Data export to Tier-1 centers
- More work is probably required for meeting the CERN Tier-2 requirements
  - Time to open files
  - Xrootd integration