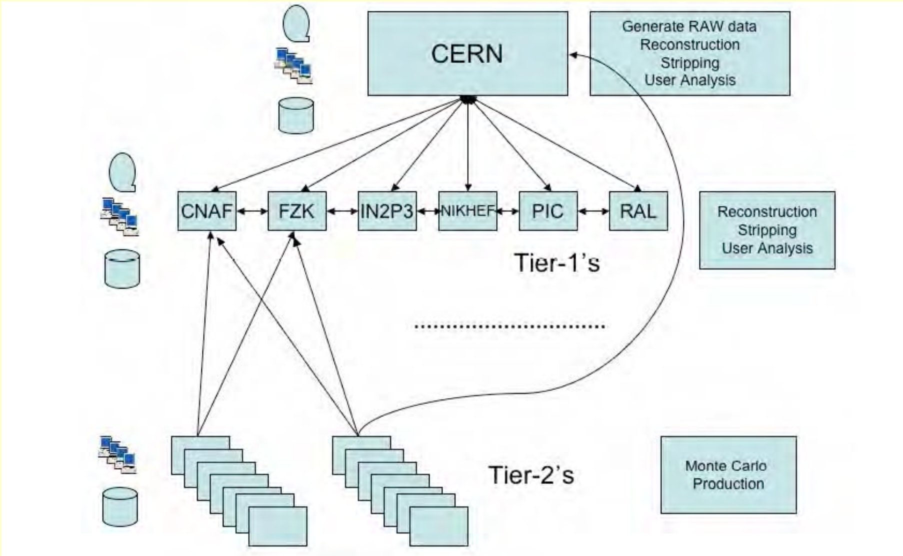
LHCb Experience with LFC Database Replication

Federico BONIFAZI (INFN-CNAF)
Angelo CARBONE (INFN-CNAF / LHCb Bologna)
Barbara MARTELLI (INFN-CNAF)
Gianluca PECO (INFN-Bologna / LHCb Bologna)

Overview

- LHCb LFC replica implementation in collaboration with 3D project.
  - LFC role in LHCb computing model
  - Oracle technologies deployed
  - Production setup
- Single replica setup: functionality, scalability and stability tests
  - Tests description and goals
  - Tests results
- Multi-replicas setup: functionality, scalability and stability tests
  - Tests description and goals: *does the setup scale with more than one replica?*
  - Tests results
- Conclusions

LFC in LHCb Computing model



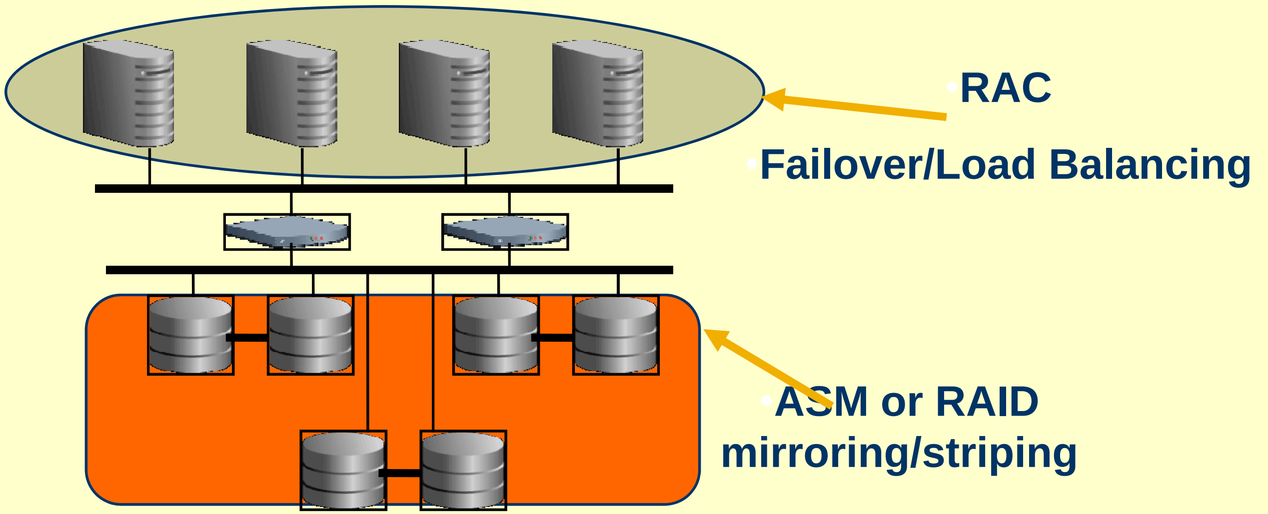
- Every LHCb application running at T0/T1/T2 needs to read/write from the LFC
- Every T1+CERN will run reconstruction using information stored in the conditions database
- The LHCb computing model foresees the LFC and conditions database replication at each T1
- The database replication becomes quite important in order to assure
  - Scalability
  - Geographic redundancy
  - Fault tolerance

LHCb LFC usage and test

- Monte Carlo simulation
  - Transfer output from a MC job to one or more Storage Element and register the file in the catalogue **(write)**
- Data processing (raw data reconstruction, analysis, stripping, etc…)
  - Send the job to the T1 site where the data are available and produce an output to be registered (**read/write**)
- Data transfer
  - find the replica to transfer, perform the transfer and register the new destination (**read/write**)
- Etc…
- In order to efficiently use a replicated database it is mandatory that master and replica database are synchronized with low latency
  - Measure the latency between source a destination databases.
  - *LHCb requirements not dramatically strict: less than 30 minutes*

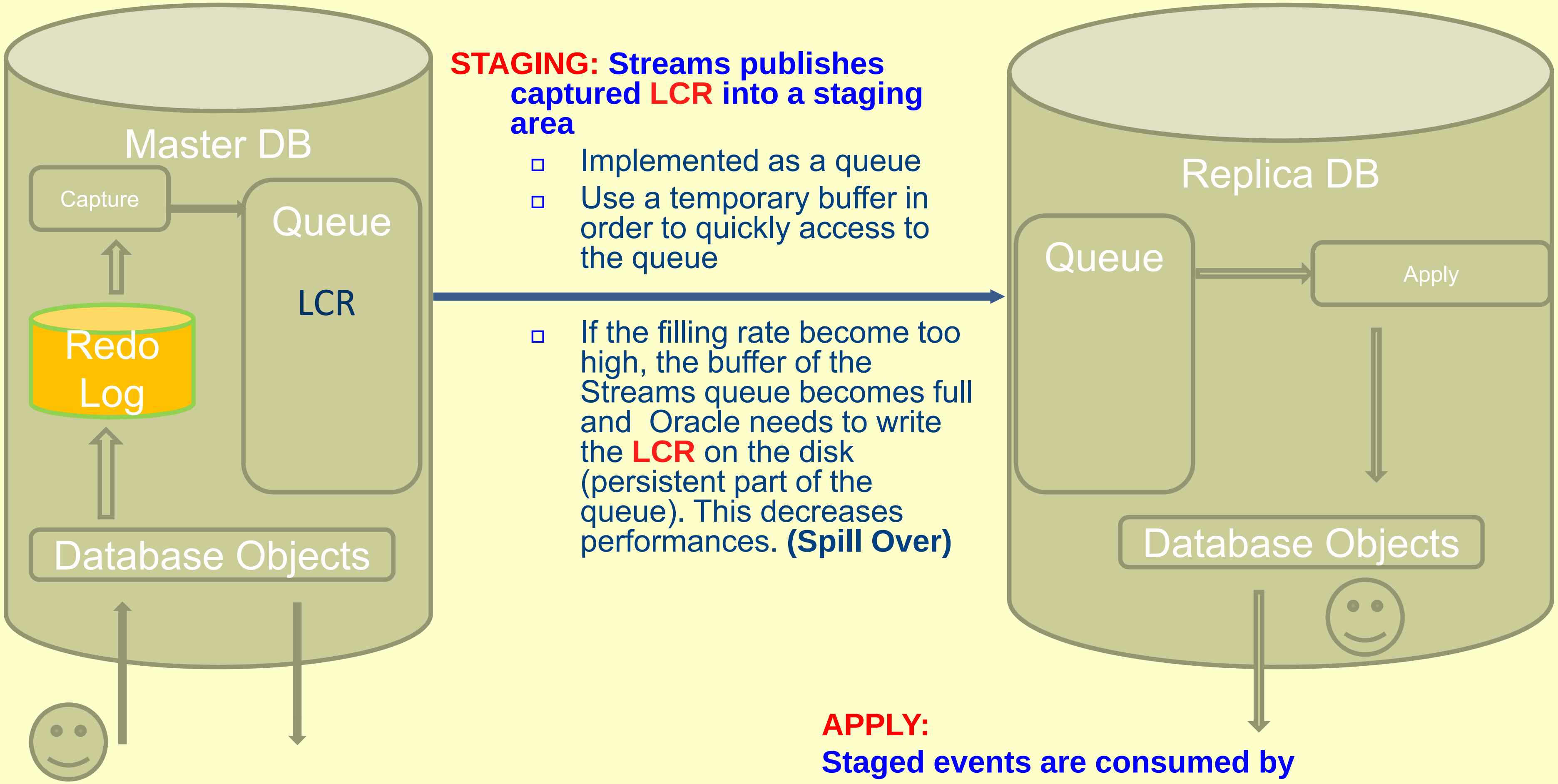
Database Deployment for LFC

- At each site the LFC backend databases are implemented using high availability technologies:
  - **Storage level**: protection from disk failures is achieved using Oracle Automatic Storage Management (ASM) on a Storage Area Network.
  - **Database level**: Oracle Real Application Cluster allows sharing of database across multiple instances.
  - **Replication Level**: Oracle Streams enables the propagation and management of data, transactions and events in a data stream from one database to another



Streams Replication

CAPTURE: Source database events are captured, filtered and stored in LCR (Logical Change Record).

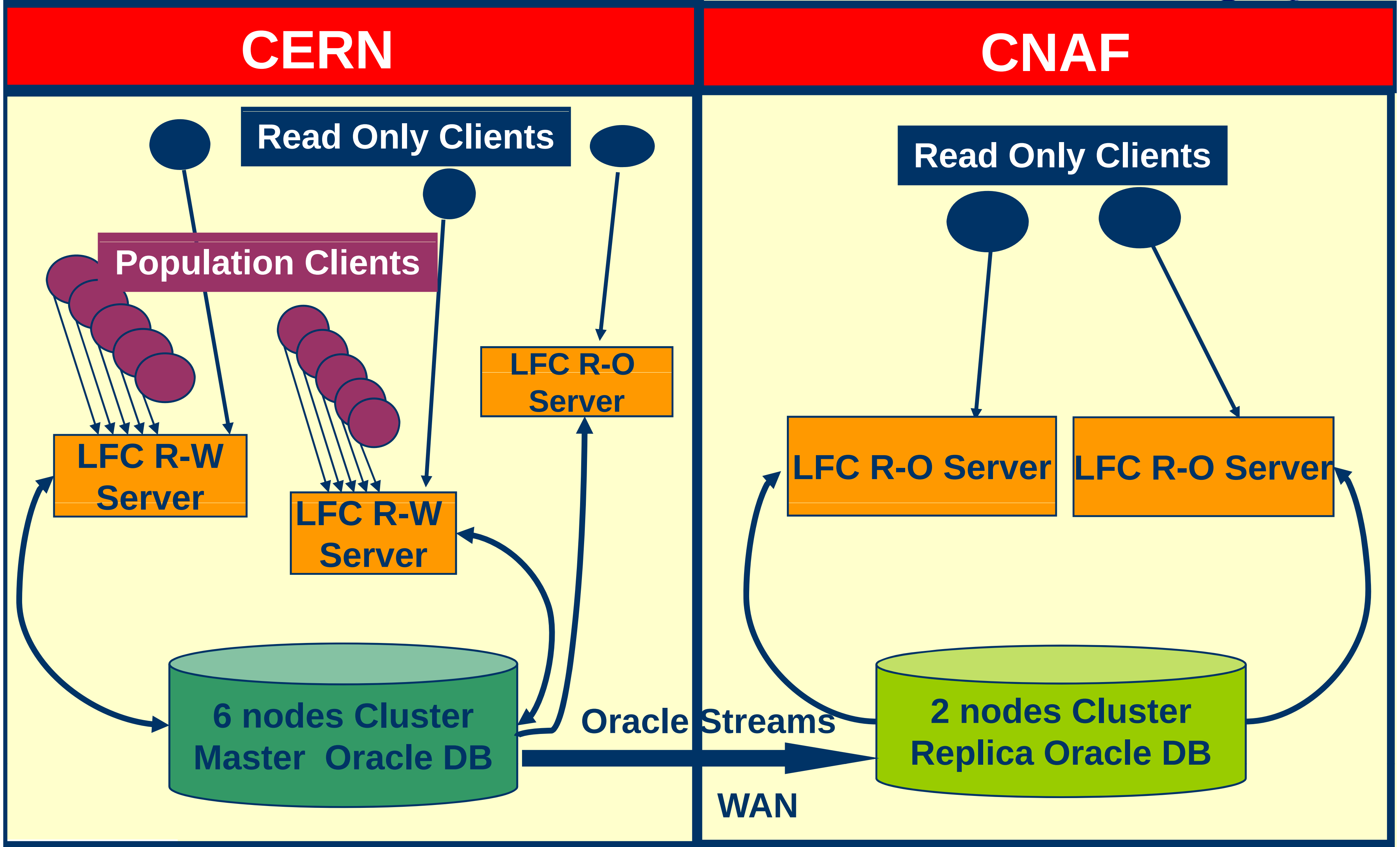


STAGING: Streams publishes captured LCR into a staging area

- Implemented as a queue
- Use a temporary buffer in order to quickly access to the queue
- If the filling rate become too high, the buffer of the Streams queue becomes full and Oracle needs to write the LCR on the disk (persistent part of the queue). This decreases performances. **(Spill Over)**

APPLY:
Staged events are consumed by subscribers to the destination database

LHCb LFC Replication deployment CERN-CNAF



LHCb LFC Replication Tests

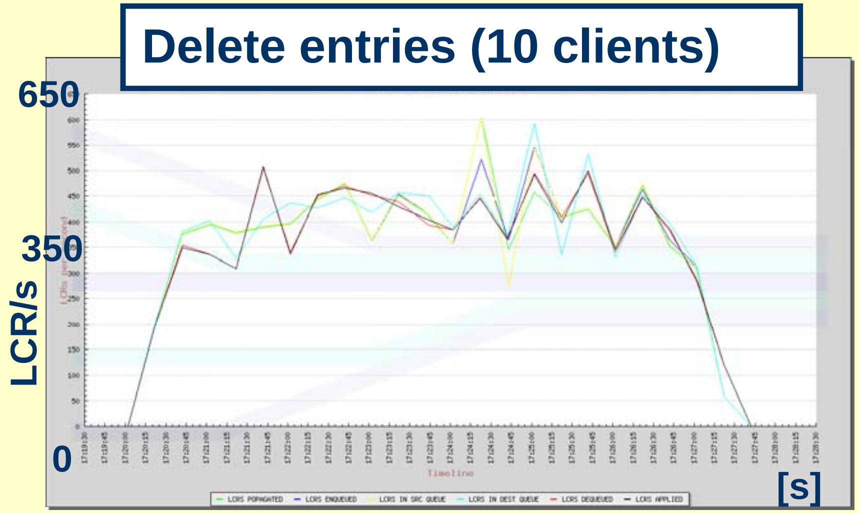
- Two different tests have been realized to evaluate
  - the time latency between the master and replicated database
  - the performance of the LFC front-end with writing/deleting operations as a function of increasing number of clients
- Python scripts using LFC API functions add files and replica to
  - lfc-lhcb.cern.ch (Master database)
  - Tests perform with increasing number of simultaneously writing and deleting clients 10,20,40,76
  - For each number of clients (10,20,40,76) added:
    - 8K files and 10 replica for each file (similar to LHCb usage) → Test I
    - 16K files and 25 replica for each file (beyond LHCb usage) → Test II
    - The load is uniformly distributed over the clients

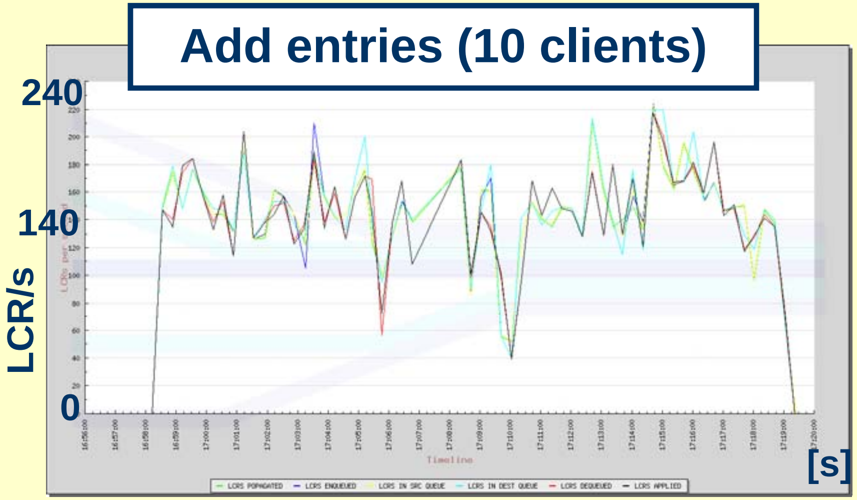
Strmmon web monitoring tool

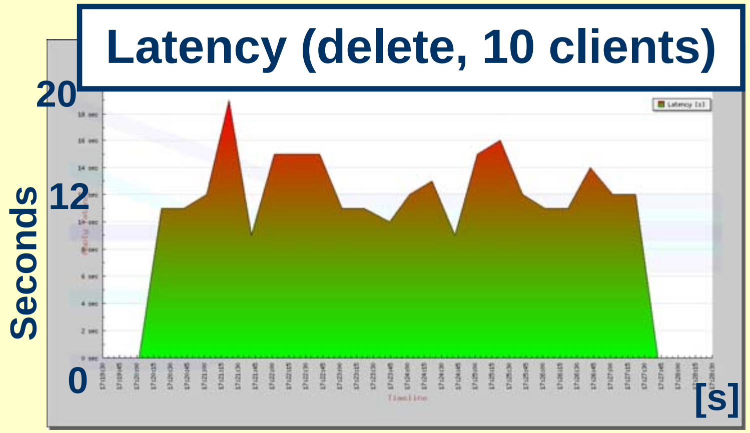
- Most of the measurements and plots shown are taken from Strmmon: the official Streams monitoring tool in the 3D project.
  - http://itrac315.cern.ch:4889/streams/streams.php
  - The tool plots the monitoring streams quantities on web previously stored on a dedicated repository database
  - Very useful to measure
    - The total LCR latency (time elapsed between the creation of the LCR at the master and the apply to the destination database)
    - LCR rate (captured, queued, dequeued, applied)

Test results [TEST I]

- TEST I : add and delete 8K files (10 replicas for each file) with 10,20,40,76 parallel clients (~90K entries)
  - Used two tables
    - CNS_FILE_METADATA
    - CNS_FILE_REPLICA

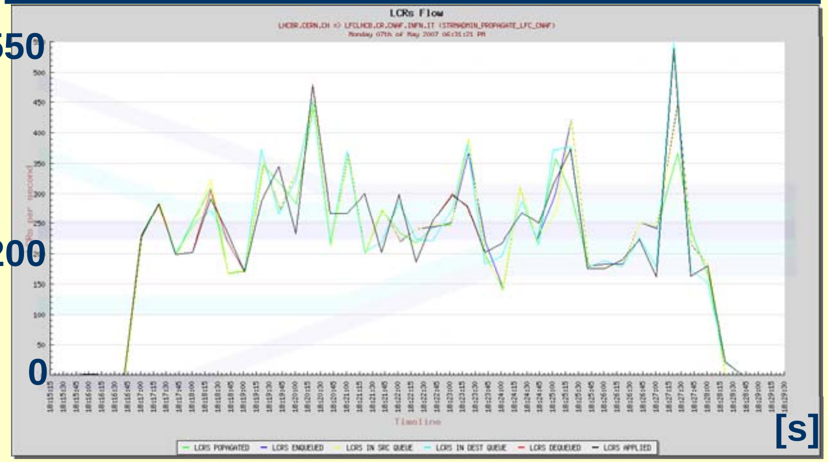




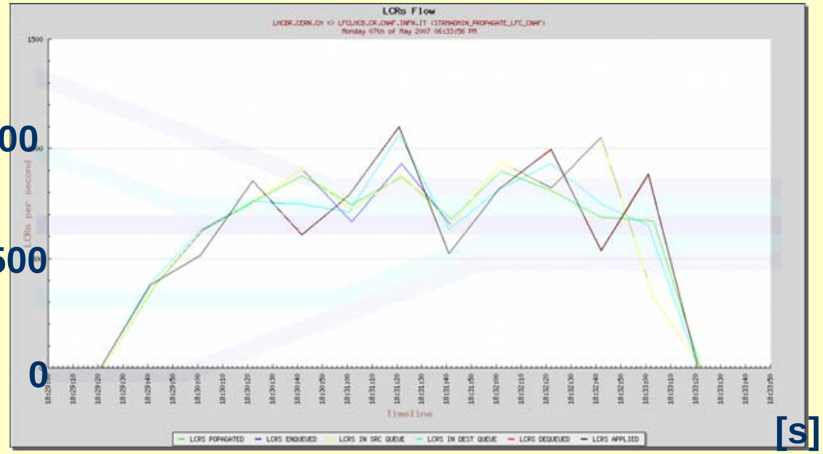


Test results [Test I]

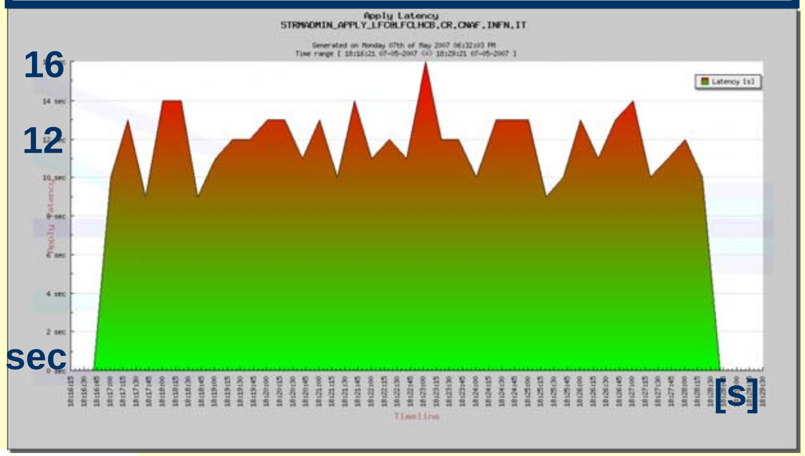
Add entries (76 clients)



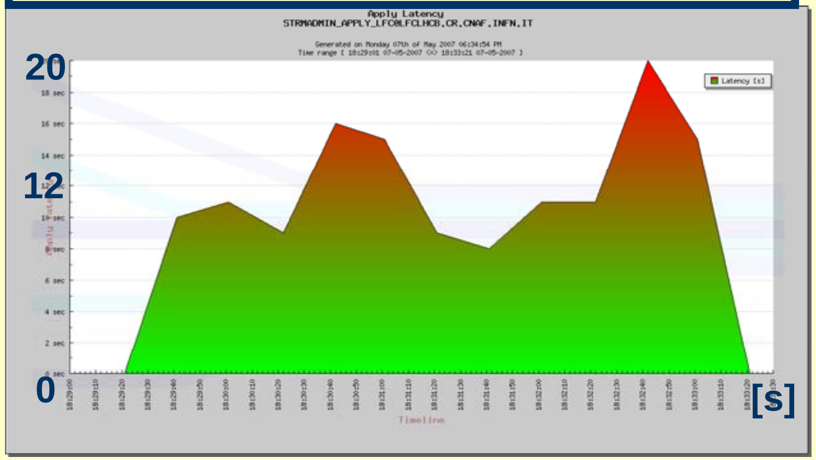
delete entries (76 clients)



Latency (add , 76 clients)

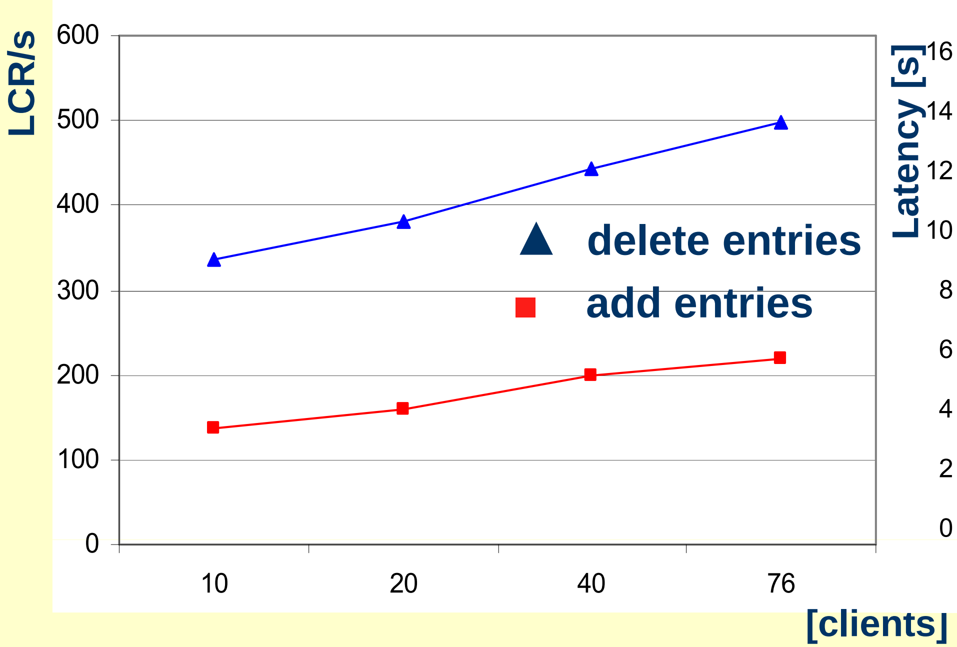


Latency (delete , 76 clients)

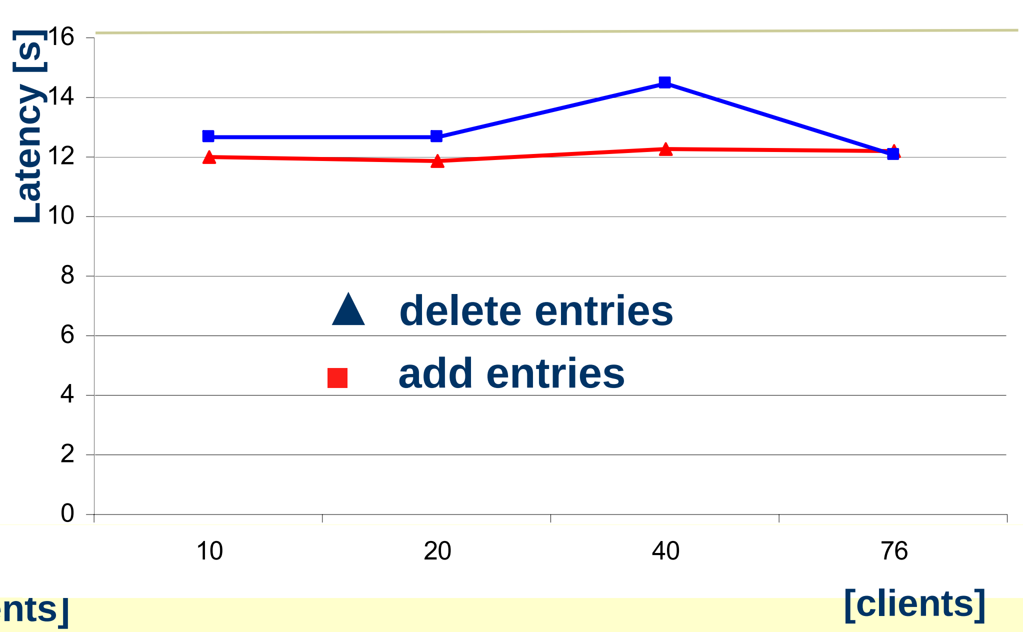


Test results [Test I]

Average LCR/s as function of clients



Average latency as function of clients

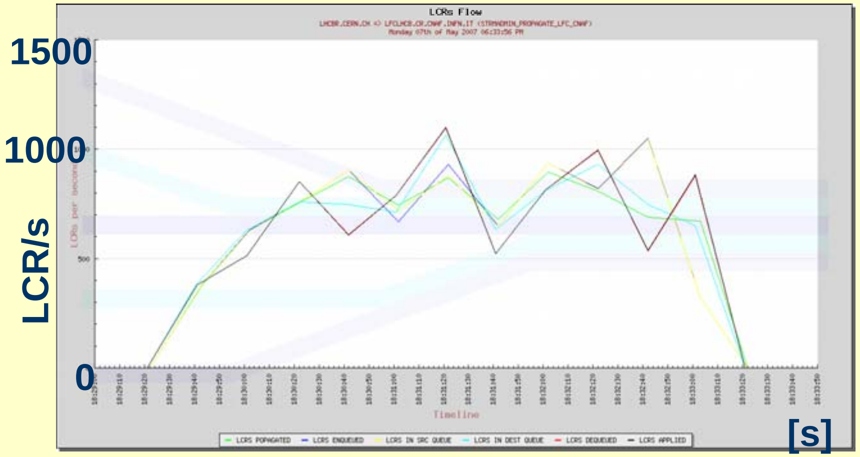


- Linear growing of LCR rate as a function of writing and deleting clients
- Latency, stable (12/13 sec.), independent from the number of clients

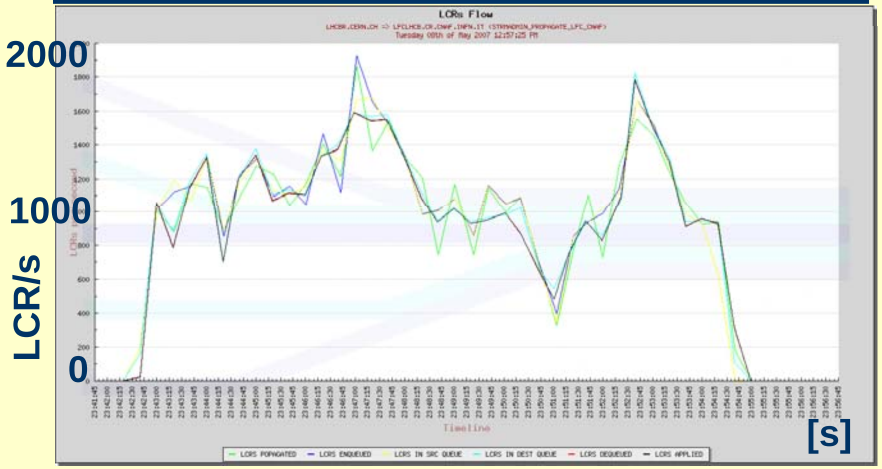
Test result [Test II]

- Test II: add and delete 16K files (25 replicas for each file) with 10,20,40,76 parallel clients (~560K entries)
  - Adding more replica per file increase the LCR rate

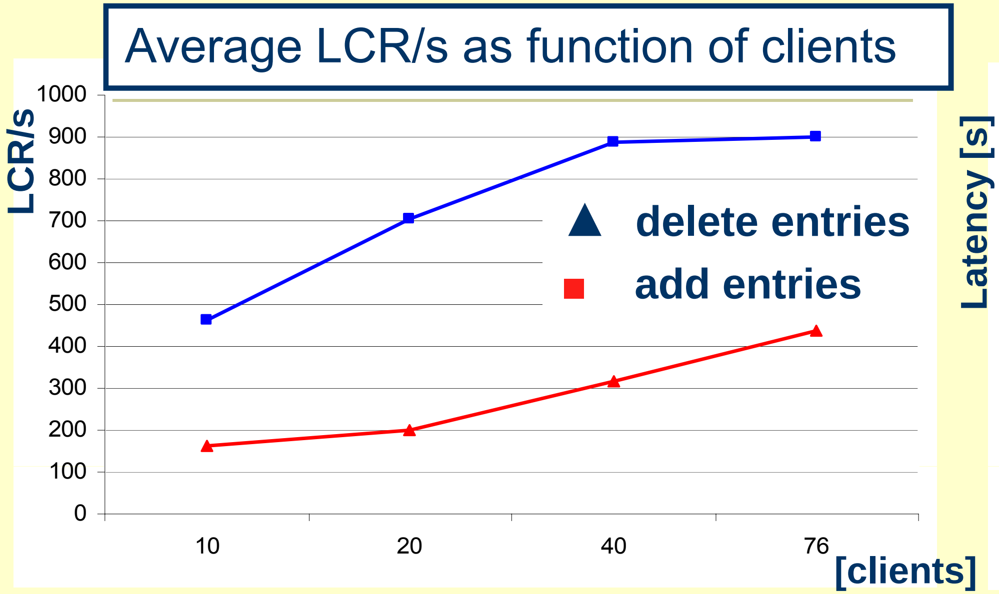
delete entries (76 clients) Test I

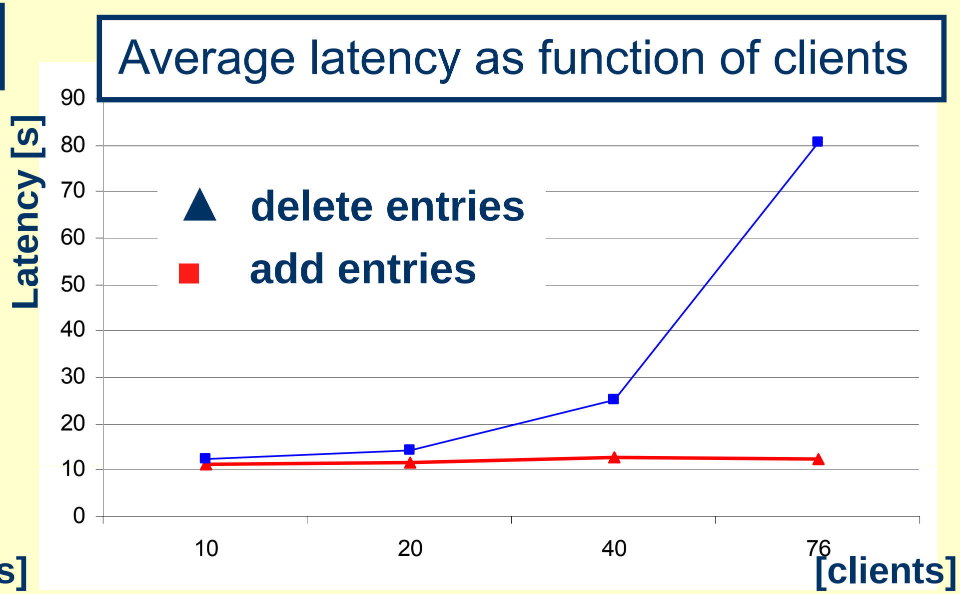


delete entries (76 clients) Test II



Test result [Test II]

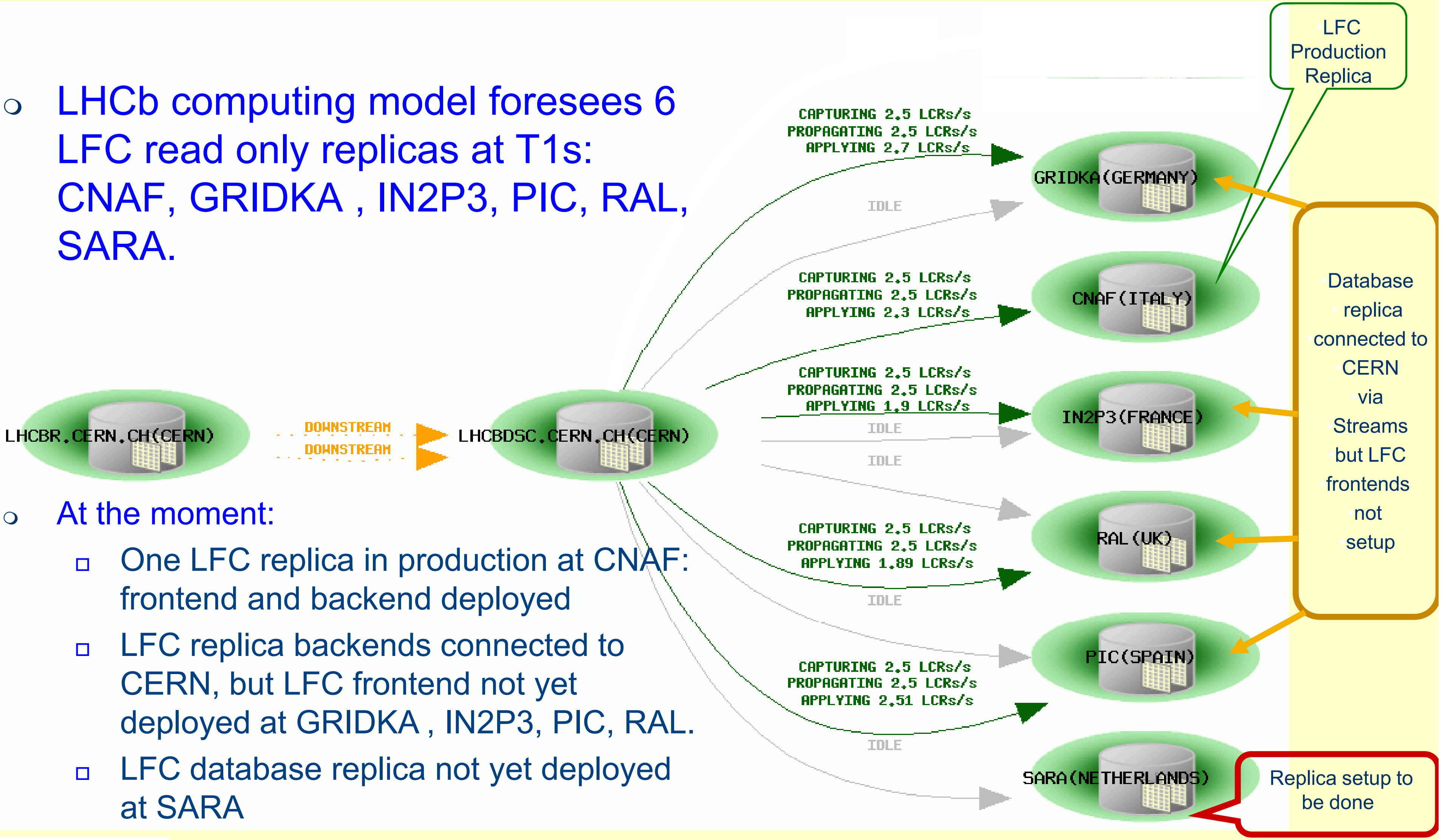




- Linear increasing of LCR rate
  - but with 76 clients the LFC front-end becomes a limit
- At the rate of 900 LCR/s the replication starts to accumulate latency
  - Increase the I/O at the source database due to grow activities
  - the latency is still much better than LHCb requirements

Multi-Replica Setup

- LHCb computing model foresees 6 LFC read only replicas at T1s: CNAF, GRIDKA , IN2P3, PIC, RAL, SARA.



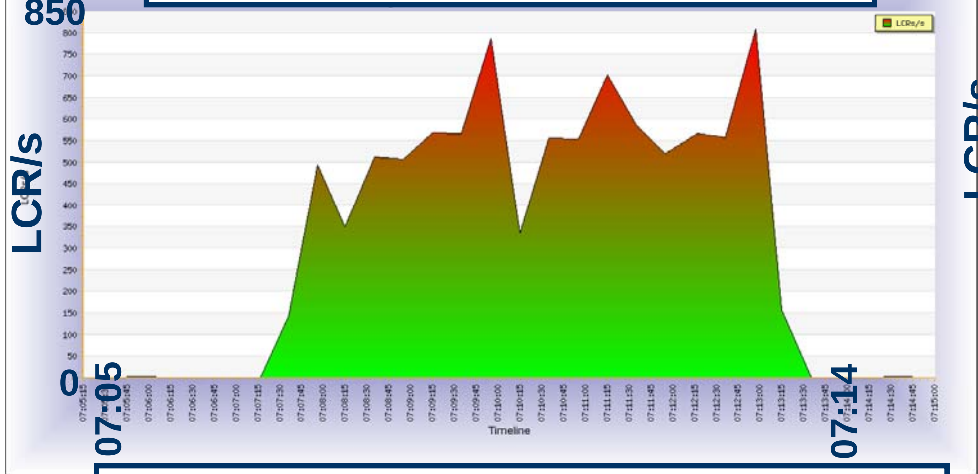
- At the moment:
  - One LFC replica in production at CNAF: frontend and backend deployed
  - LFC replica backends connected to CERN, but LFC frontend not yet deployed at GRIDKA , IN2P3, PIC, RAL.
  - LFC database replica not yet deployed at SARA

Multi-replica Setup: Scalability and Stability Tests

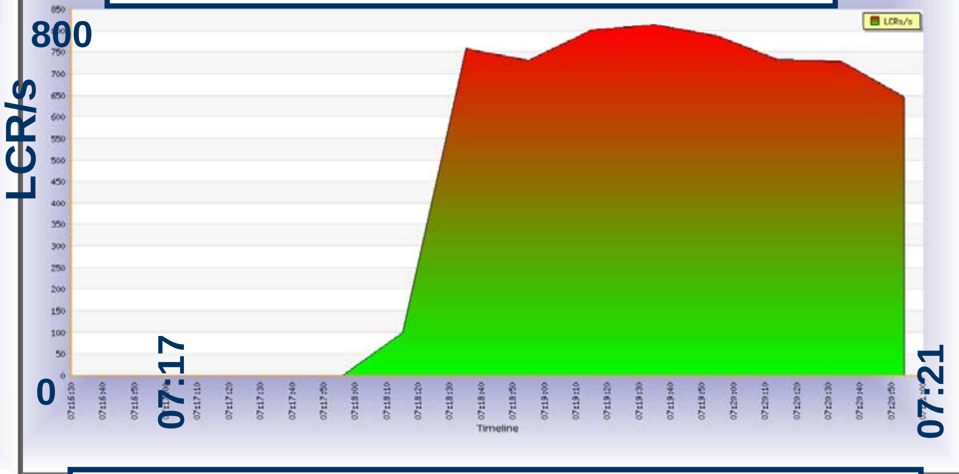
- Scalability and stability tests performed
  - inserting entries in the LFC front-end at CERN;
  - monitoring the replication speed, latency and sinchronization at CNAF, GRIDKA , IN2P3, PIC, RAL, SARA.
- While tests with CNAF replica where performed reading the entries from the LFC front-end at T1, now we need to *read directly from the database back-ends* because LFC front-ends are not yet deployed. *This fact doesn't impact the results at all.*
- The same python test suite written for the single-replica test is used.
- Scalability test:
  - 8K files are inserted plus 10 replicas for each file (similar to LHCb usage).
  - 10, 20, 40, 76 threads per LFC are used: near to the maximum (80 threads) in present deployment.
  - Comparison with previous tests (done in a single replica setup).
- Stability test:
  - the same script is run with 76 clients and 100K file (plus 10 replicas for each file). The files are first added and after a pouse of 5 minutes, removed from the catalog. This operation puts to work the LFC for ~1:30 hours.

Scalability Test Results

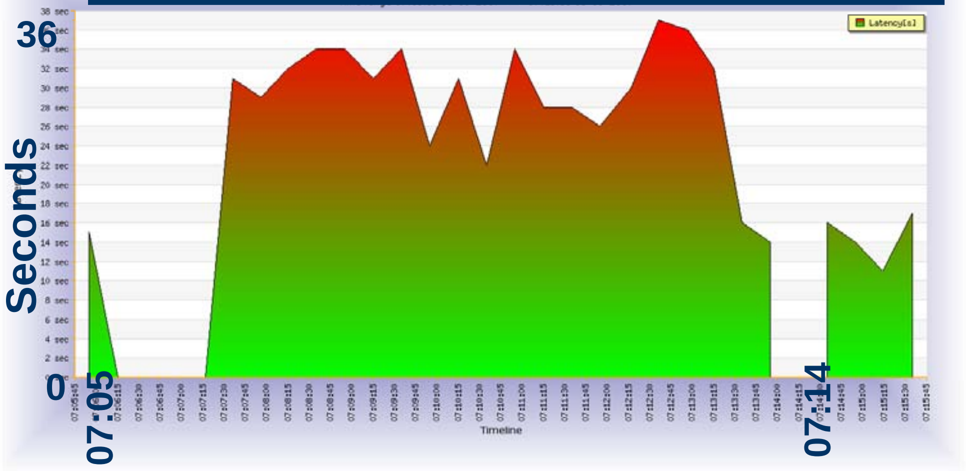
Add entries (76 clients)



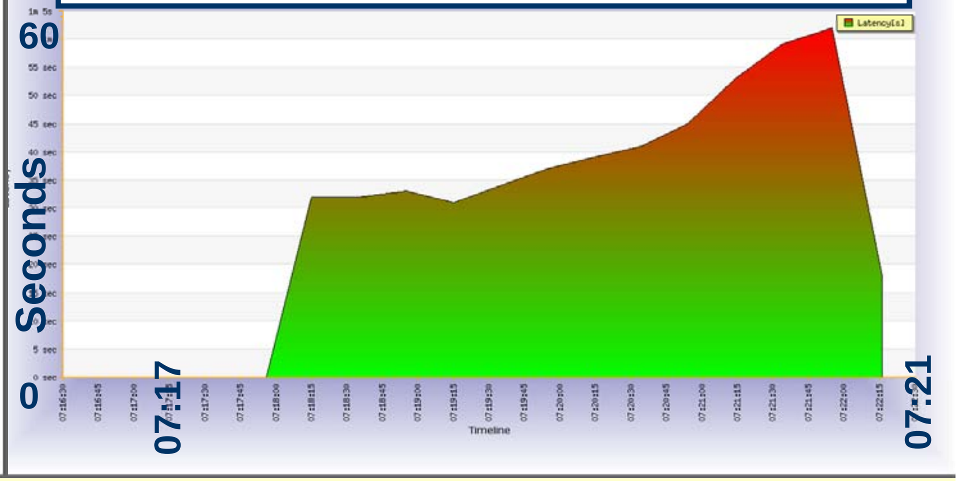
delete entries (76 clients)



Latency (add , 76 clients)

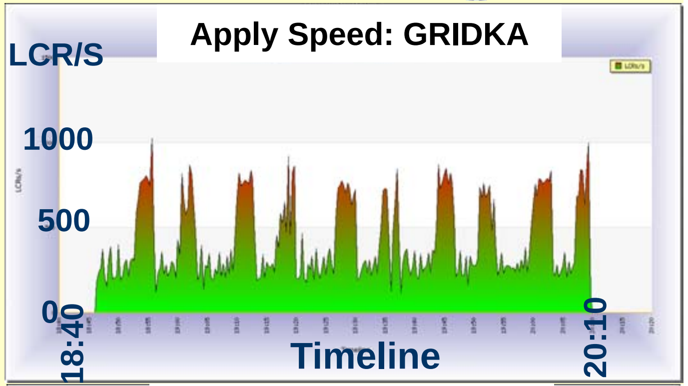


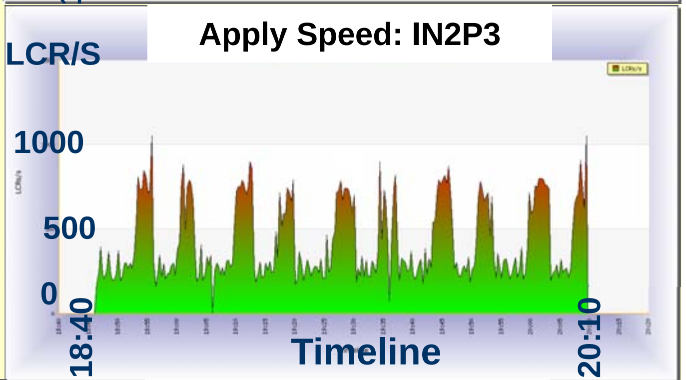
Latency (delete , 76 clients)

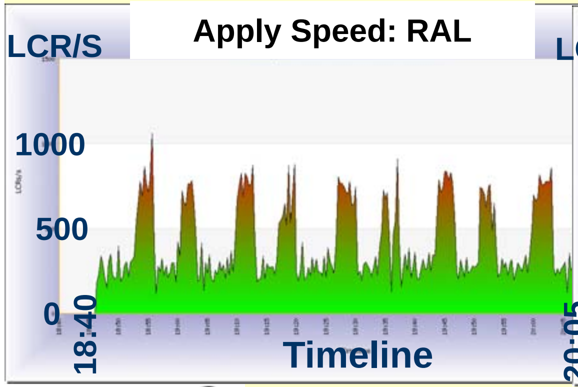


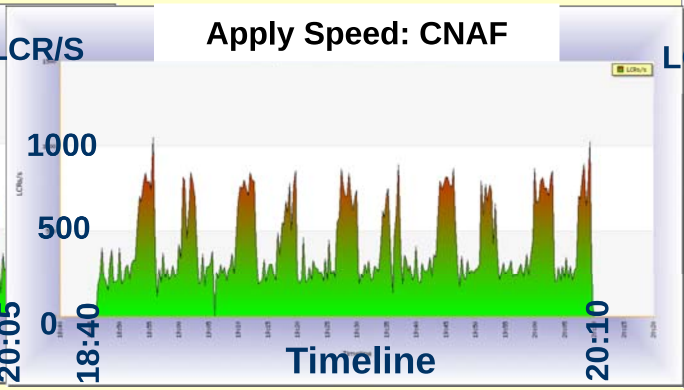
Stability Tests: Apply Speed

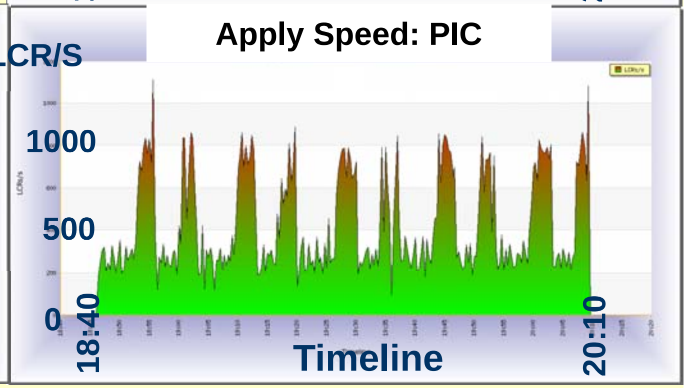
- Reached the speed 1K LCR/s (not less than 1-replica tests).
- No spilling has been detected during the tests.
- Replication rate sustained for 1 hour and half.
- Plots have the same shape: all replicas behave in the same way.
- Peaks are due to queue filling up and emptying.





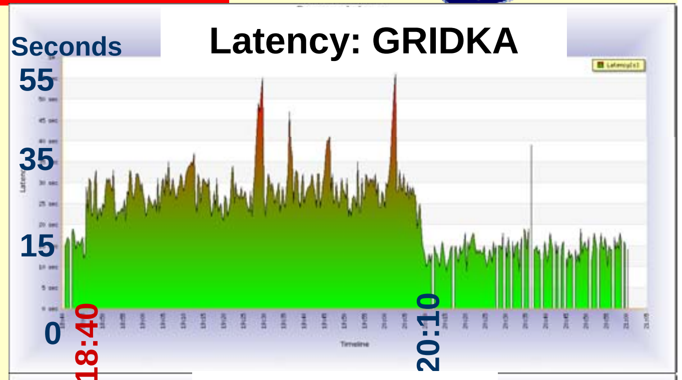


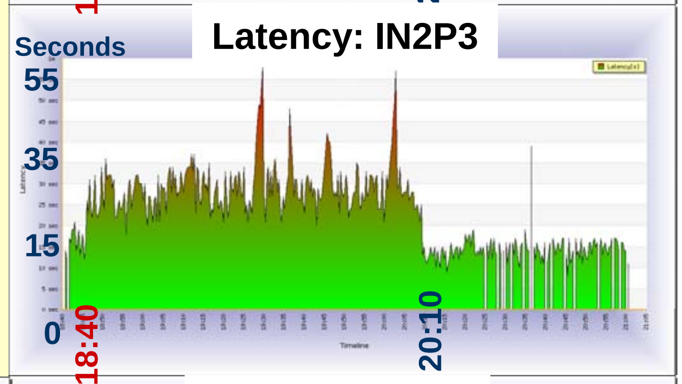


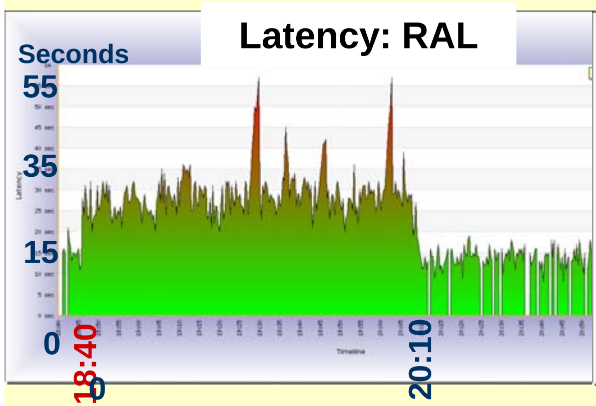


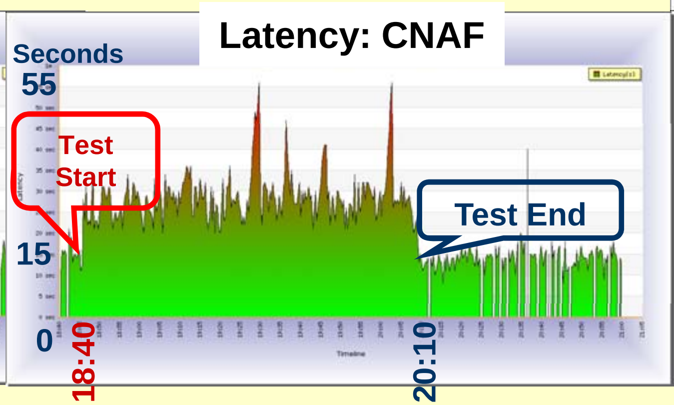
Stability Tests: Replication Latency

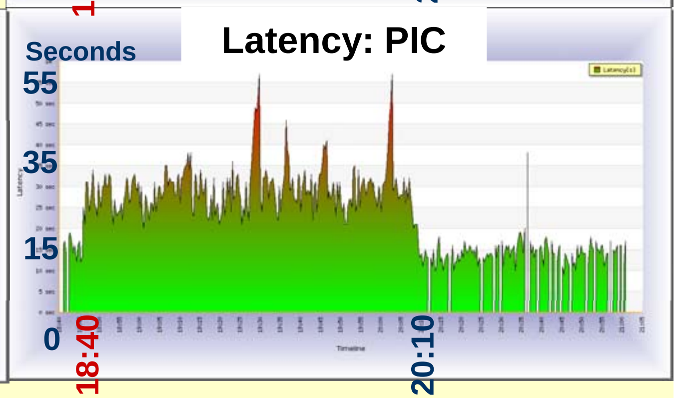
- Latency during the stress test varies from ~15 s to ~55 seconds (peaks).
- Considering that latency during low load periods is about 10-15 seconds, stress tests impact on latency is very low.

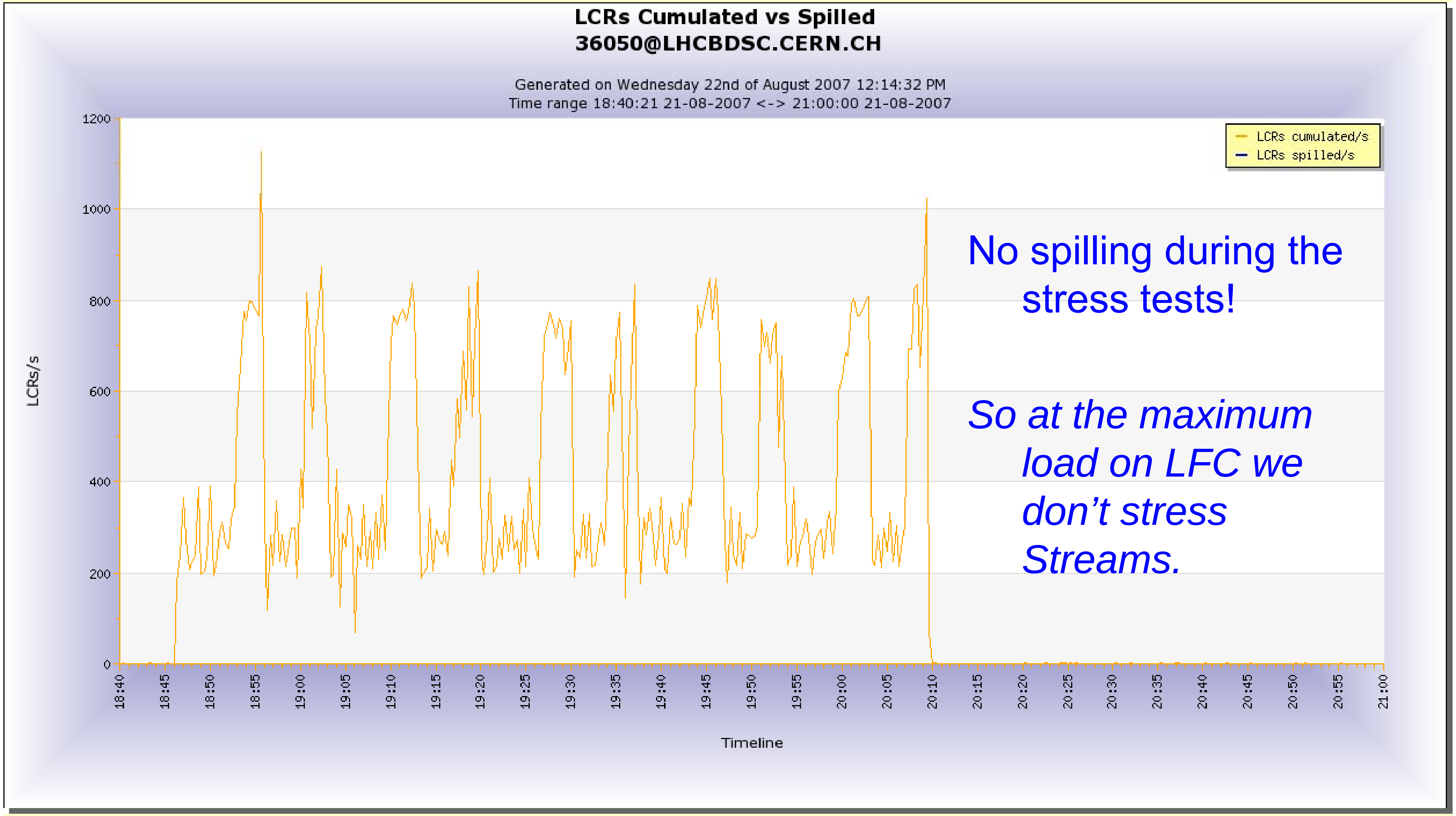












No spilling during the stress tests!

So at the maximum load on LFC we don't stress Streams.

Conclusions

- High Availability is a key issue for database services and is well addressed by present Oracle technologies.
- 3D project has successfully deployed such technologies achieving good stability and reliability of the service at CNAF as a pilot site, now to all the other T1 centres .
- Adding replicas to the setup doesn’t impact Streams replication performances:
  - Latency doesn’t grow.
  - Replication speed doesn’t decrease.
- All T1’s behave in the same way:
  - Plots about replication speed and latency are pretty much the same
- Streams replication is not a bottleneck on LFC performances.
- LHCb requirements about latency and performances are largely met.