Oracle Streams Performance

Eva Dafonte Pérez

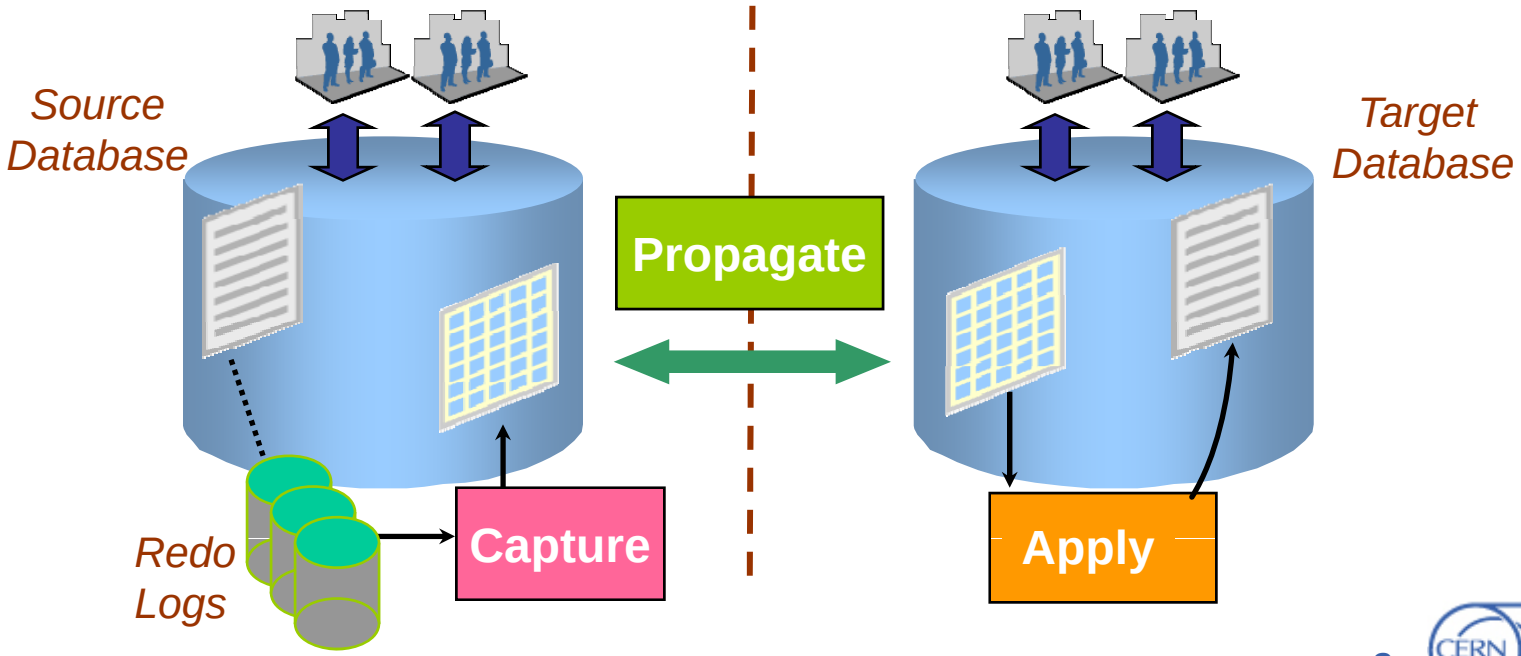
CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Outline

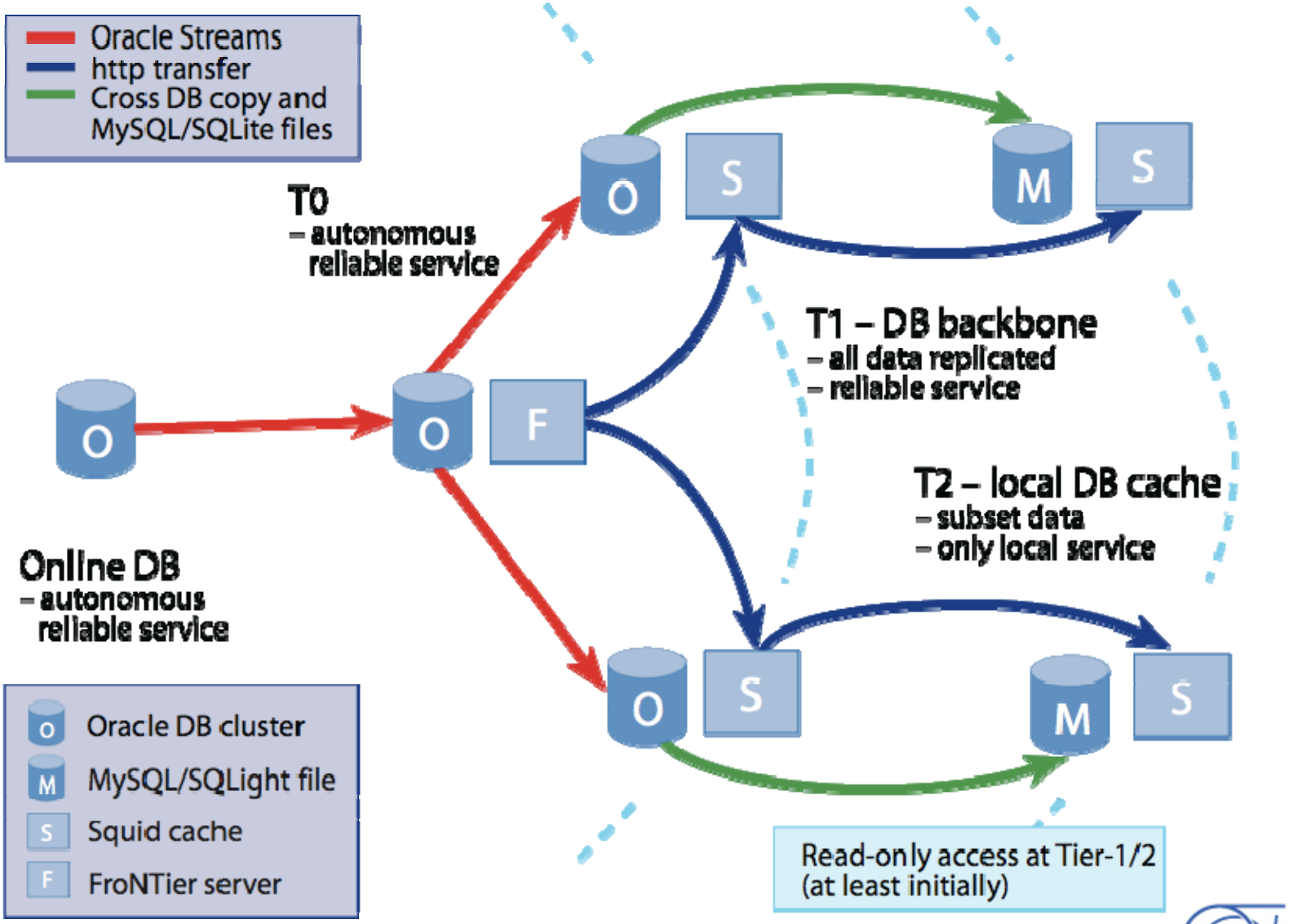
- Oracle Streams Replication
- Replication Performance
- Streams Optimizations
  - Downstream Capture
  - Split & Merge Procedure
  - TCP and Network
  - Rules
  - Flow Control
  - Lessons Learned
  - Transportable Tablespaces for Scalable Resynchronization
- Summary

Oracle Streams Replication

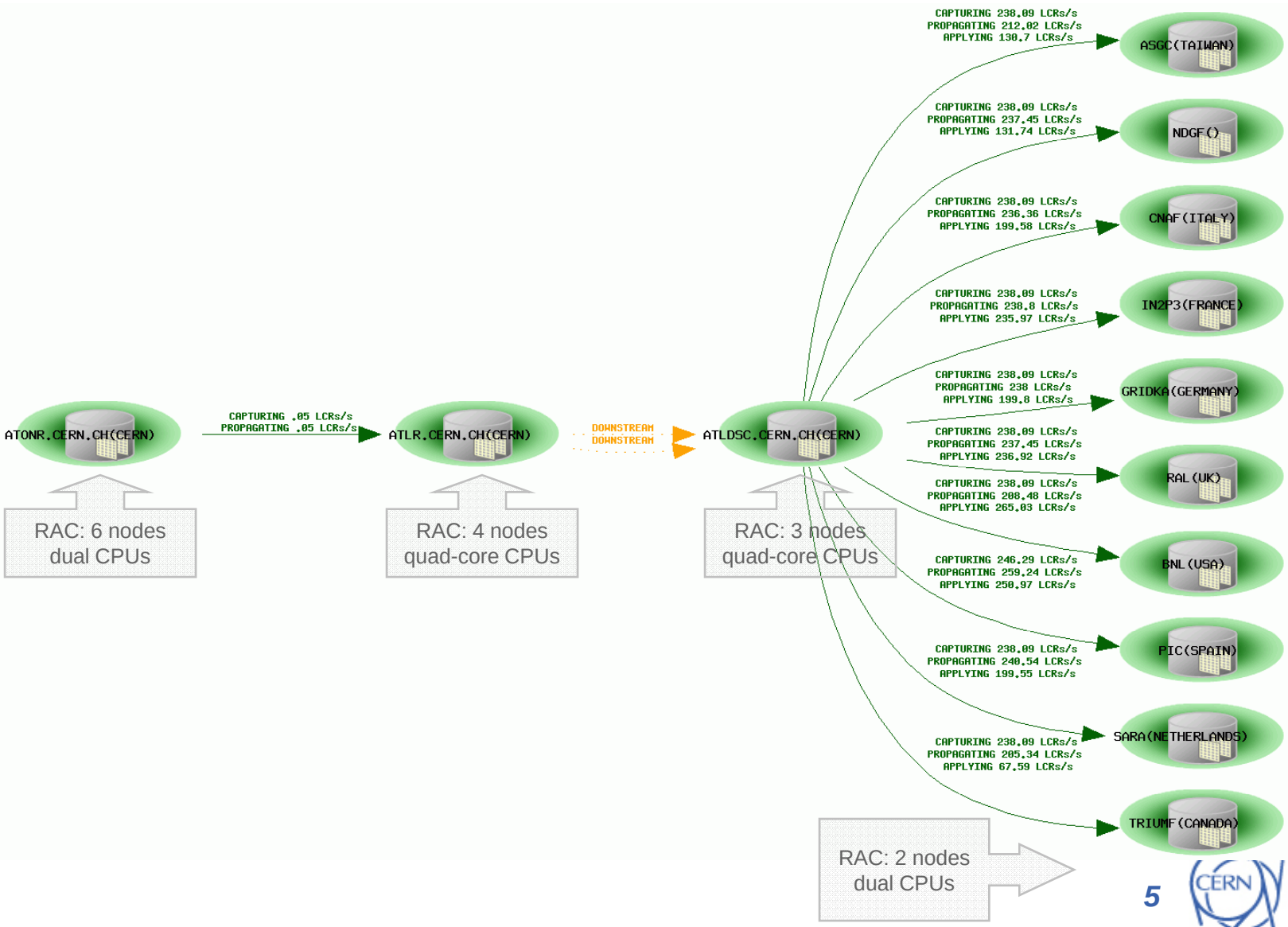
- Technology for sharing information between databases
- Database changes captured from the redo-log and propagated asynchronously as Logical Change Records (LCRs)



LCG Distributed Architecture

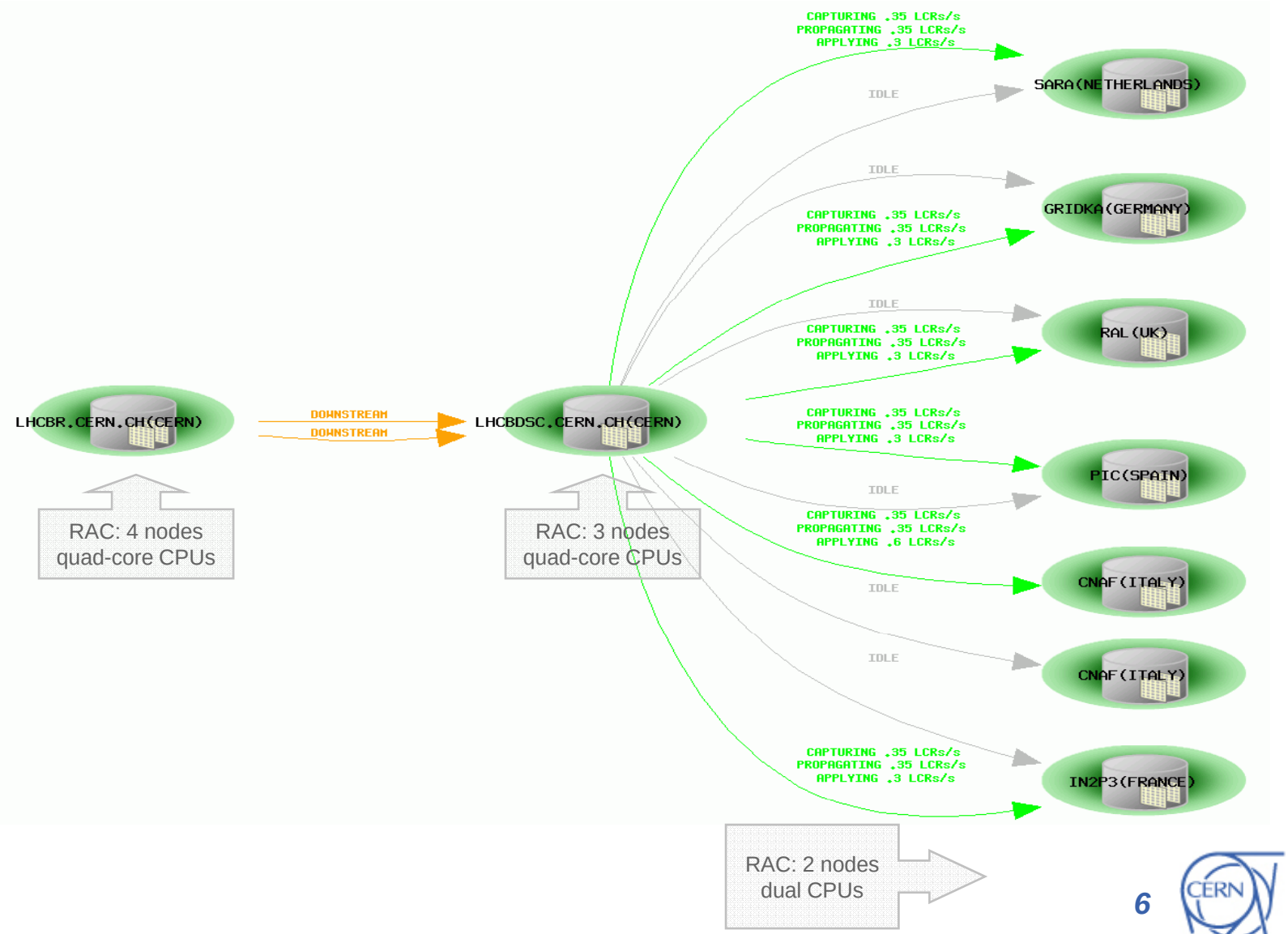


Streams Setup for ATLAS



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Streams Setup for LHCb (Conditions and LFC)



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Replication performance

- The atomic unit is the change record: LCR
- LCRs can vary widely in size
  - ➔Throughput is not a fixed measure

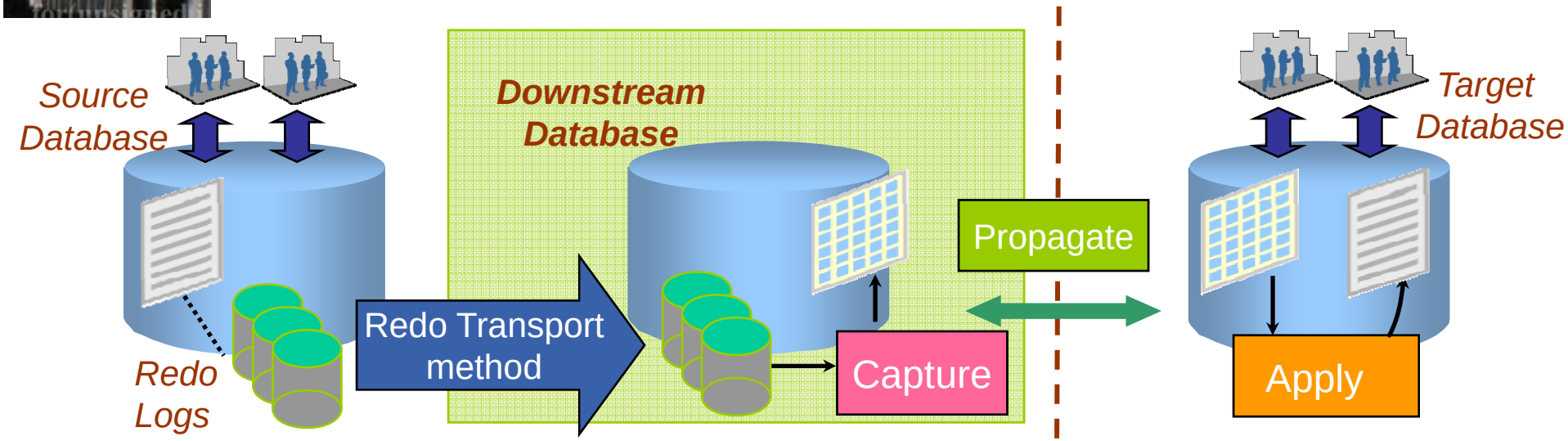
- Capture performance:
  - Read LCRs information from the redo
    - From redo log buffer (memory - much faster)
    - From archive log files (disk)
  - Convert changes into LCRs
    - depends on the LCR size and number of columns
  - Enqueue the LCRs
    - Independent of the LCR size

Replication performance

- Propagation performance:
  - Browse LCRs
  - Transmit LCRs over the network
  - Remove LCRs from the queue
    - Done in separate process to avoid any impact
- Apply performance:
  - Browse LCRs
  - Execute LCRs
    - Manipulate the database is slower than the redo generation
    - Execute LCRs serially => apply cannot keep up with the redo generation rate
  - Remove LCRs from the queue

Downstream Capture

- Downstream capture to de-couple Tier 0 production databases from destination or network problems
  - source database availability is highest priority
  - more resources allocated to the Streams processes
- Optimizing redo log retention on downstream database to allow for sufficient re-synchronisation window
  - we use 5 days retention to avoid tape access
- Optimizing capture parameters
  - retention time and checkpoint frequency



Split & Merge Procedure

- If one site is down
  - LCRs are not removed from the queue
  - capture process might be paused by flow control

    ➔ impact on replication performance

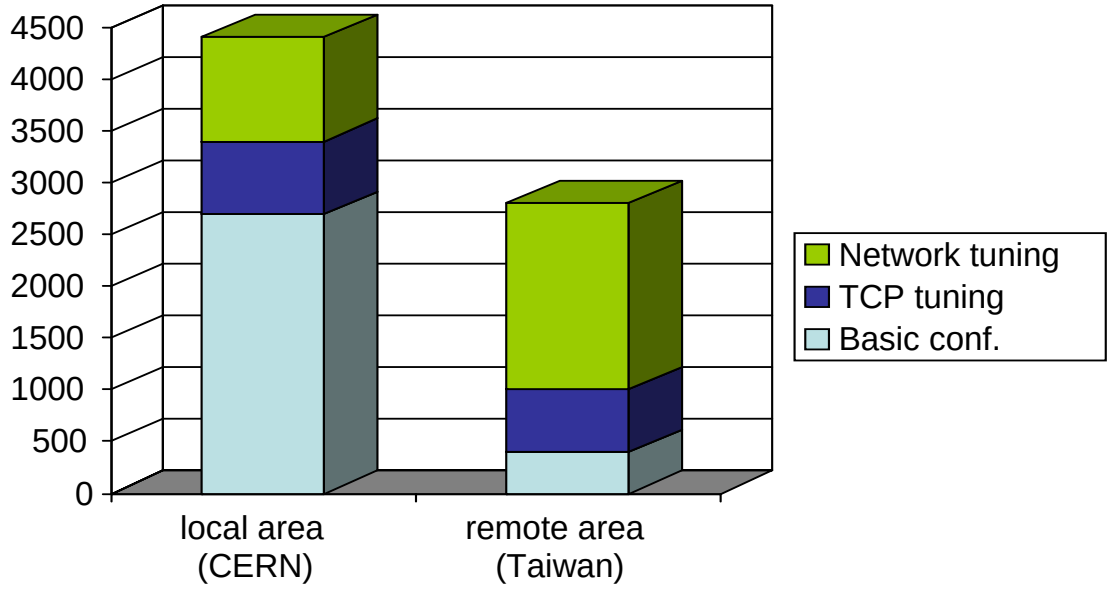
- Objective: isolate replicas against each other
  - **split** the capture process
    - (original) Real-Time capture for sites “in good shape”
    - (new) normal capture for site/s unstable/s
      - new capture queue and propagation job
      - original propagation job is dropped

      →spilled LCRs are dropped from the original capture queue
  - **merge** the capture processes
    - resynchronization

suggested by Patricia McElroy (Principal Product Manager Distributed Systems/Replication)

TCP and Network Optimizations

- TCP and Network tuning
  - adjust system max TCP buffer (/etc/sysctl.conf)
  - parameters to reinforce the TCP tuning
    - DEFAULT_SDU_SIZE=32767
    - RECV_BUF_SIZE and SEND_BUF_SIZE
      - Optimal: 3 * Bandwidth Delay Product
- Reduce the Oracle Streams acknowledgements
  - alter system set events '26749 trace name context forever, level 2';



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Streams Rules

- Used to control which information to share
- Rules on the capture side caused more overhead than on the propagation side
- Avoid Oracle Streams complex rules

Complex Rule

```
condition =>  '( SUBSTR(:ddl.get_object_name(),1,7)  IN ("COMP200", "OFLP200", "CMCP200",
                 "TMCP200", "TBDP200", "STRM200")
               OR SUBSTR (:ddl.get_base_table_name(),1,7) IN ("COMP200", "OFLP200", "CMCP200",
"TMCP200", "TBDP200", "STRM200") ) '
```

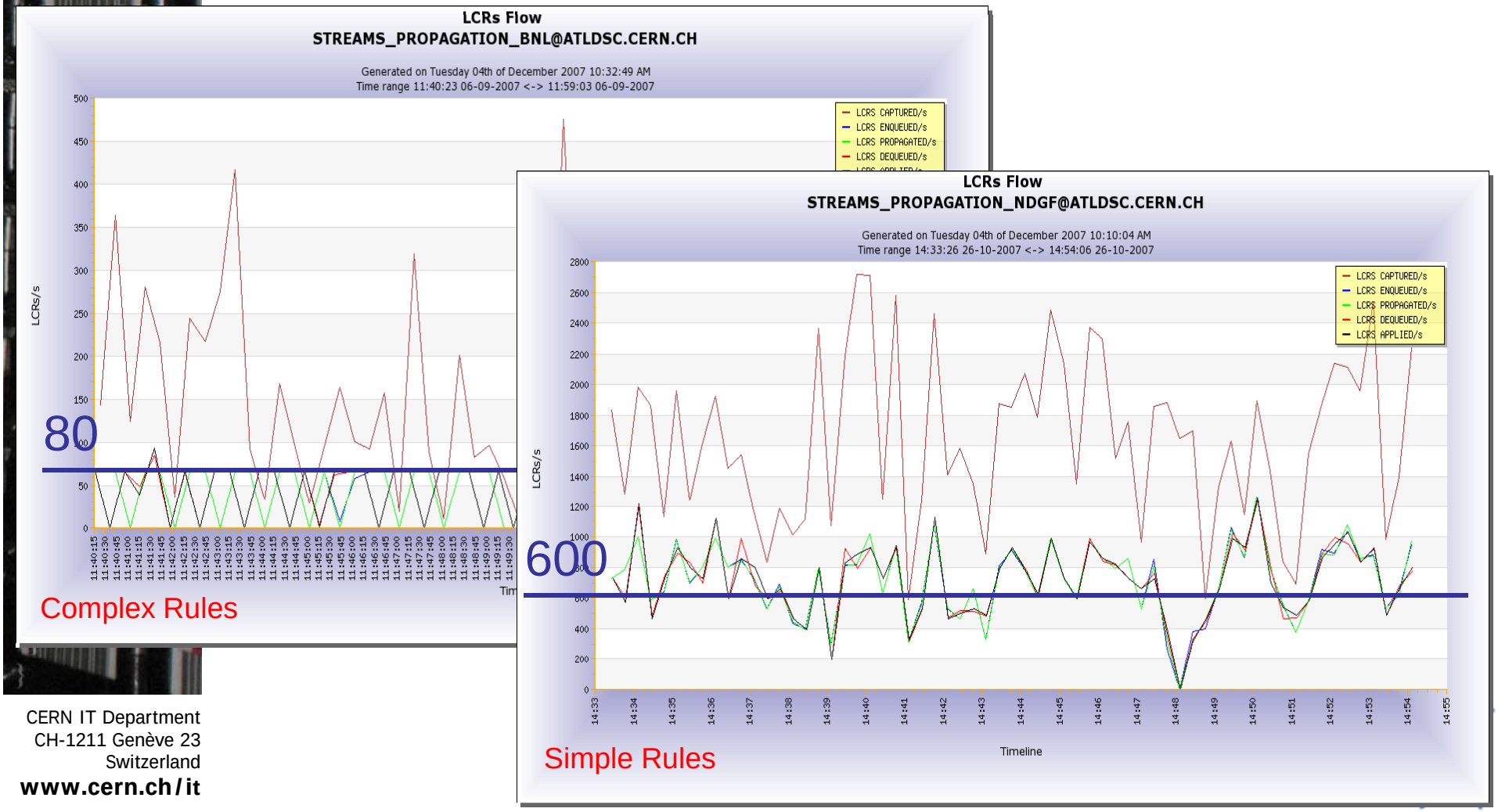
Avoid complex rules:
- LIKE
- Functions
- NOT

Simple Rule

```
condition => '(((:ddl.get_object_name() >= "STRM200_A" and :ddl.get_object_name() <= "STRM200_Z") OR
      (:ddl.get_base_table_name() >= "STRM200_A" and :ddl.get_base_table_name() <= "STRM200_Z"))
      OR ((:ddl.get_object_name() >= "OFLP200_A" and :ddl.get_object_name() <= "OFLP200_Z") OR
      (:ddl.get_base_table_name() >= "OFLP200_A" and :ddl.get_base_table_name() <= "OFLP200_Z"))
```

Streams Rules

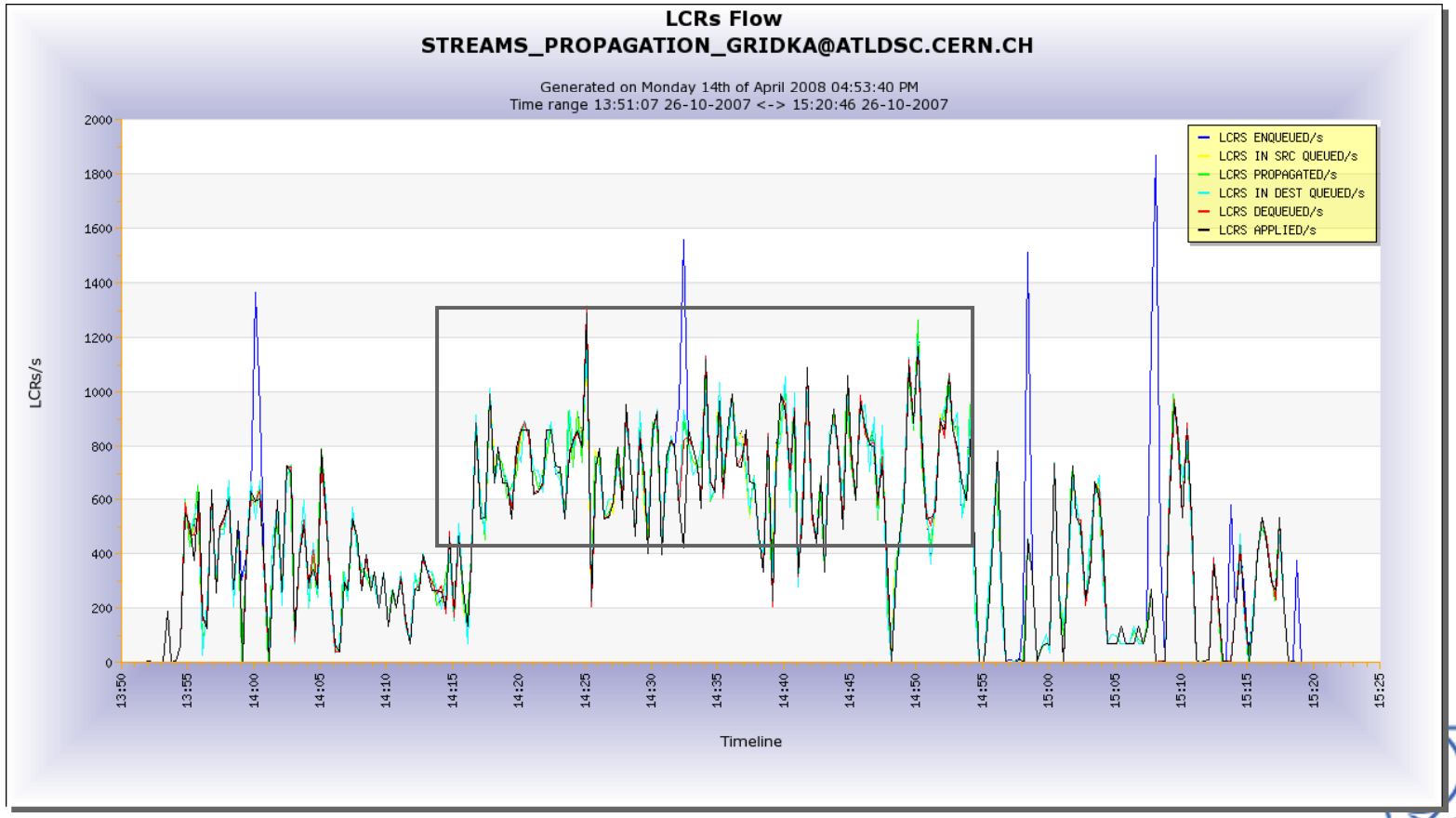
- Example: ATLAS Streams Replication
  - filter tables by prefix



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Streams Rules

- Example: ATLAS Conditions
  - 2 GB per day
  - Offline (CERN) to Tier1 (Gridka)



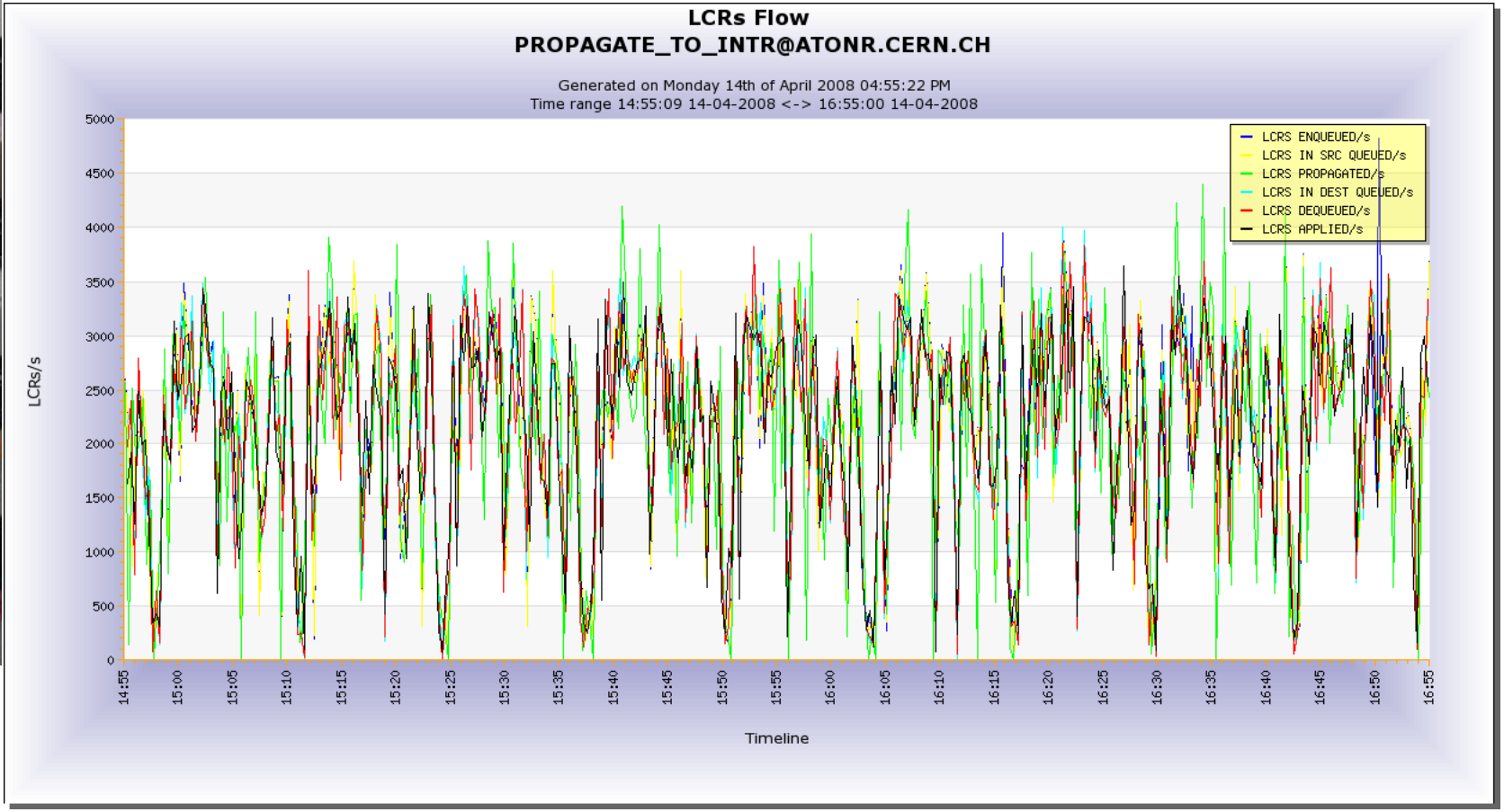
CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Flow Control

- By default, flow control kicks when the number of messages is larger than the threshold
  - Buffered publisher: 5000
  - Capture publisher: 15000
- Manipulate default behavior
- 10.2.0.3 + Patch 5093060 = 2 new events
  - 10867: controls threshold for any buffered message publisher
  - 10868: controls threshold for capture publisher
- 10.2.0.4 = 2 new hidden parameters
  - "_capture_publisher_flow_control_threshold"
  - "_buffered_publisher_flow_control_threshold"

Flow Control

- Example: ATLAS PVSS tests
  - Online to Offline (CERN), 6 GB per day
  - default: 2500 - 3000 LCRs per second
  - new thresholds: 3500 - 4000 LCRs per second



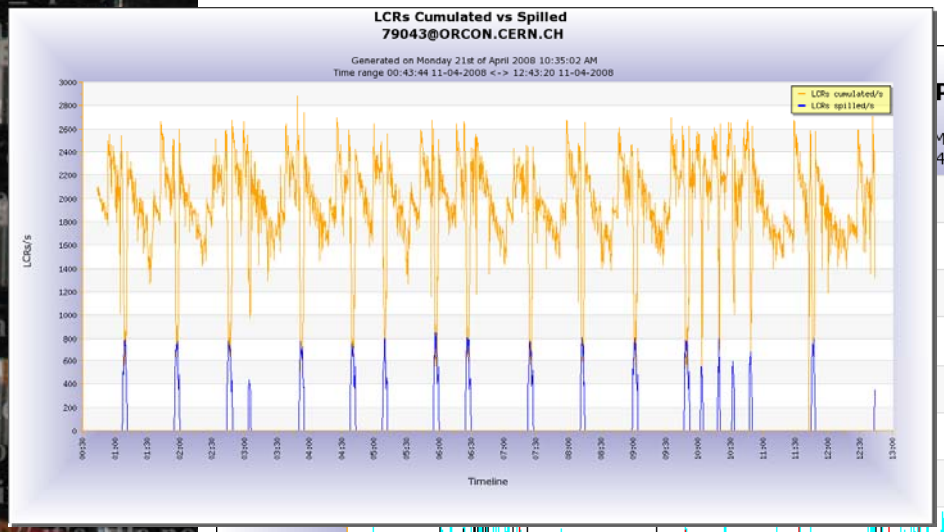
CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

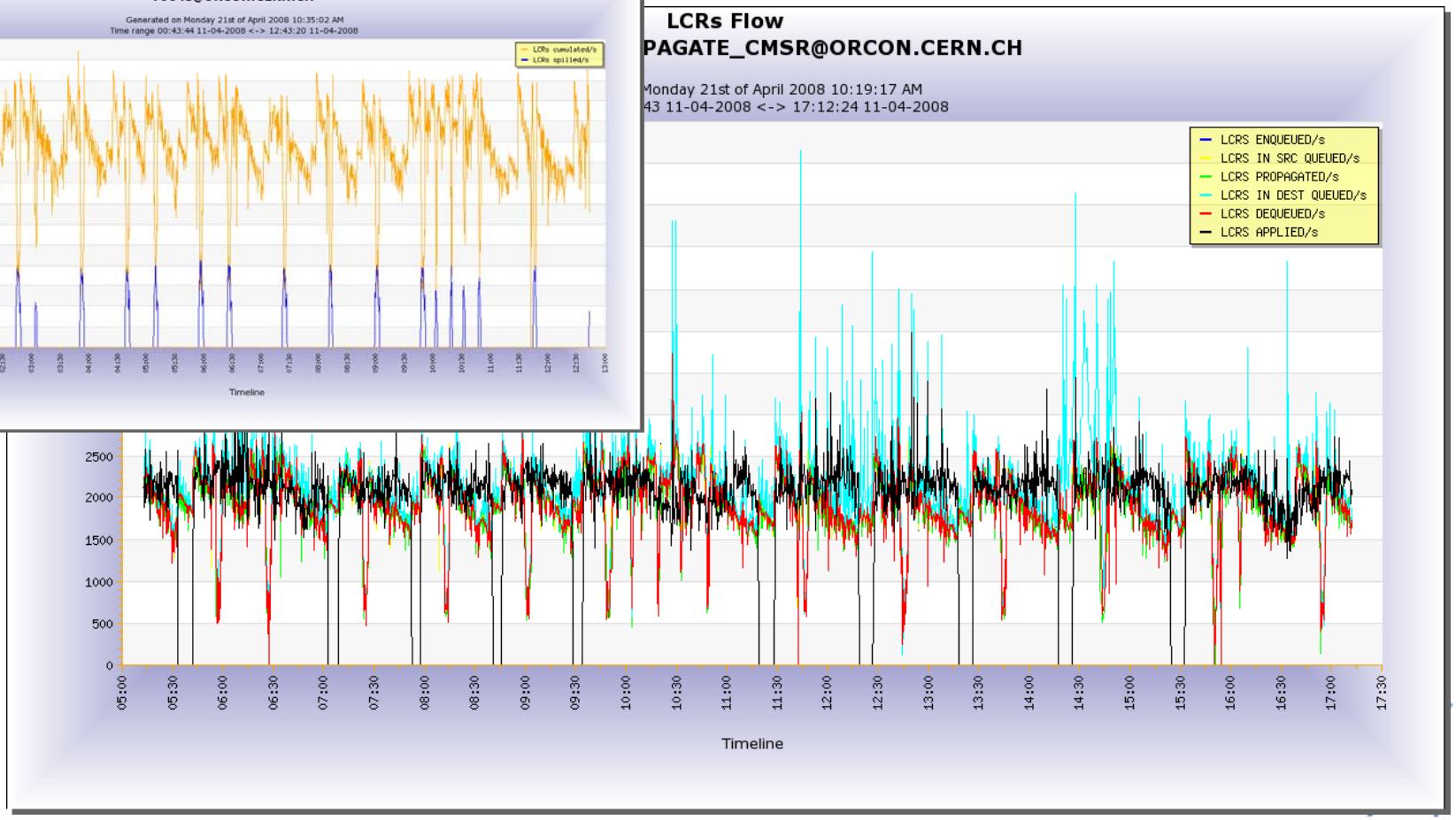
Lessons Learned

- SQL bulk operations (at the source db)
  - May map to many elementary operations at the destination side
  - Need to control source rates to avoid overloading
- Long transactions (non-frequent commits)
  - Total number of outstanding LCRs is too large
  - LCRs are in memory too long
    - ➔LCRs are spilled over to disk
    - ➔Apply performance is impacted
  - All LCRs in a single transaction must be applied by one apply server
    - ➔Parallel servers cannot be used efficiently
  - Too many unbrowsed messages enables flow control
    - ➔Streams processes are paused

Lessons Learned

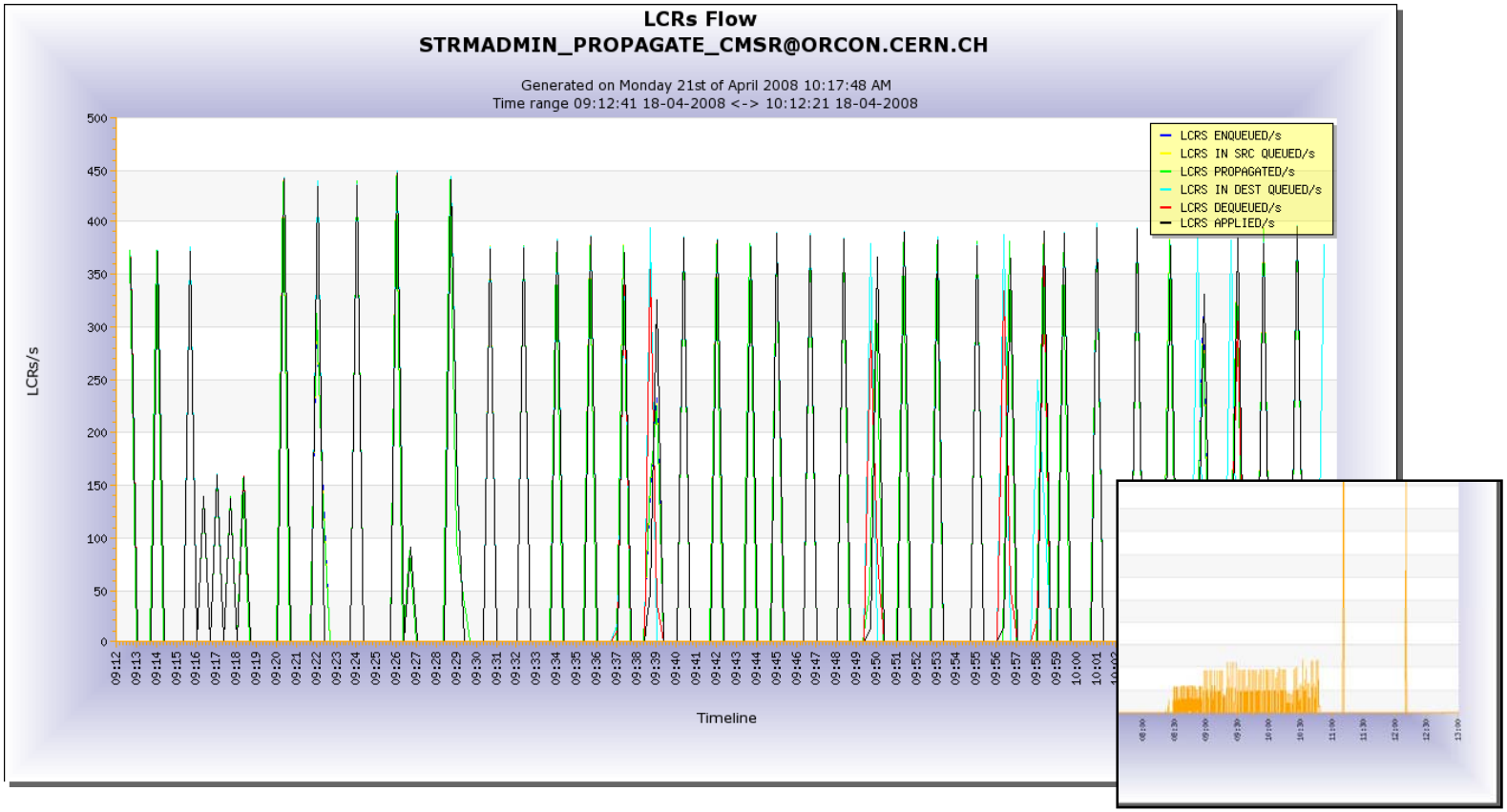
- Example: CMS replication
  - Online to Offline (CERN)
  - Single transaction mapping 428400 LCRs





Lessons Learned

- Example: CMS replication
  - Use BLOB objects: single transaction mapping 3600 LCRs

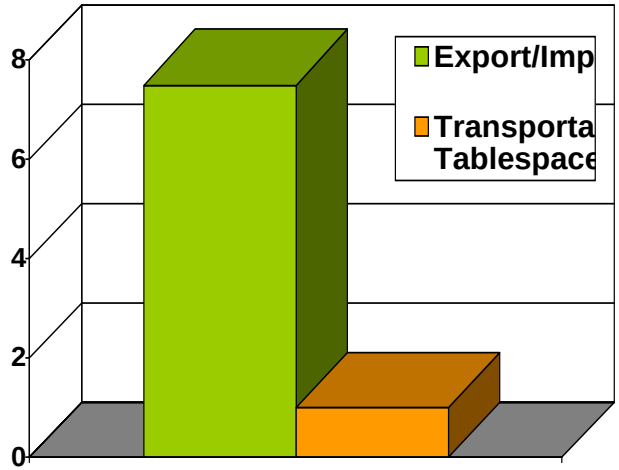


- Performance depends on the number of LCRs per transaction, rather than LCR size

Scalable Resynchronization

- Target site out of the Streams recovery window
- Complete transfer of data (schemas and tables) using Oracle Data Pump utility to destination database may take days
  - Example ATLAS Conditions data
- Transportable Tablespaces: move a set of tablespaces from one Oracle database to another
  - Export metadata of tablespace instead of data in tablespace

Example: ATLAS COOL test



Moving data using transportable tablespaces is much faster than Data Pump export/import

Scalable Resynchronization

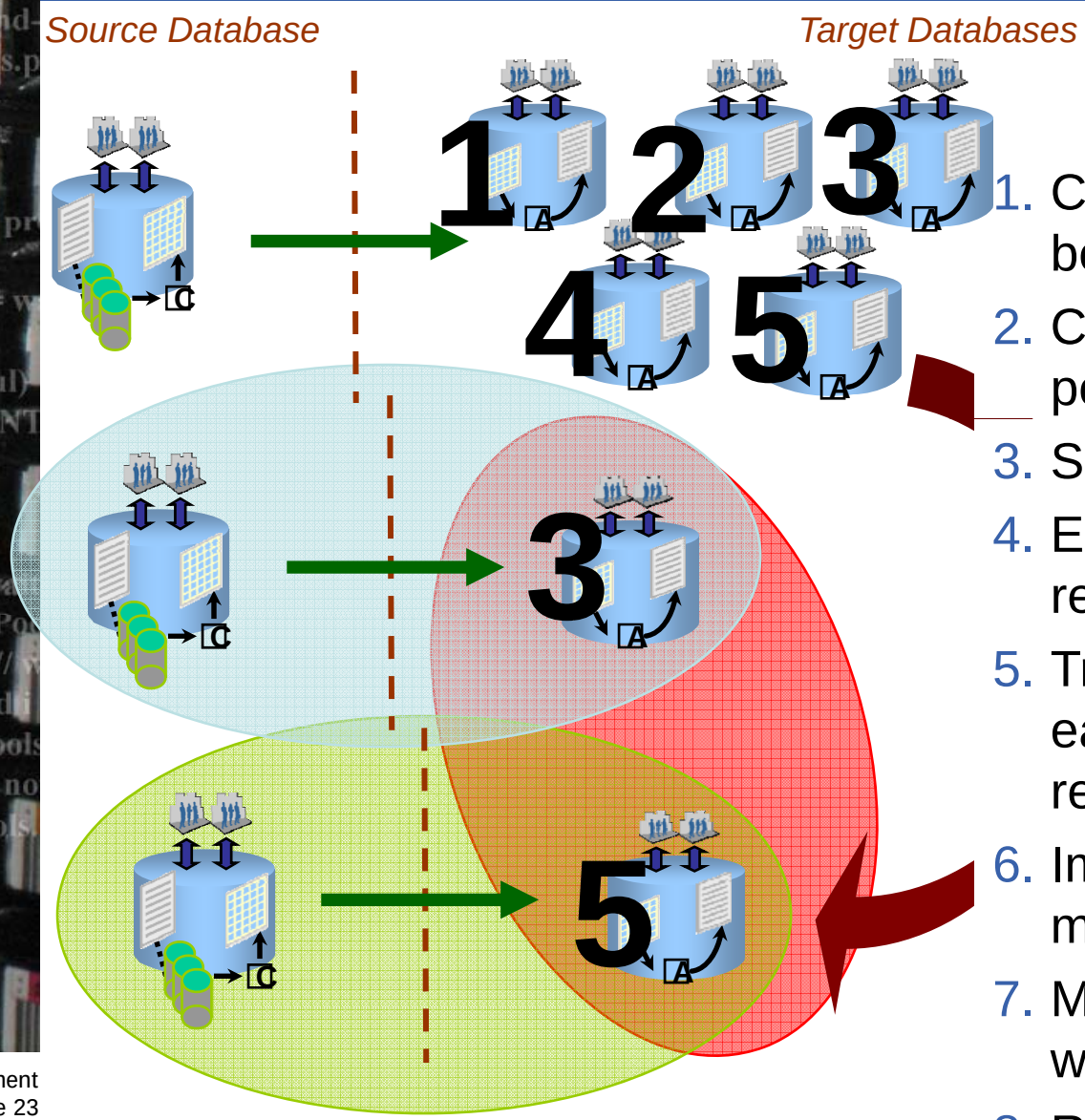
- Restrictions:
  - Database block size and character set must be the same at source and target
  - The Oracle release and the compatibility must be the same or higher at target
  - Tablespaces must be self contained
  - Tablespaces need to be set to read-only while the files are copied

It is NOT possible to use tablespaces from Tier0

BUT we can use tablespaces from Tier1

- Cross-Platform since Oracle 10g
  - Oracle provides a byte-order-conversion solution that uses Oracle Recovery Manager (RMAN)

Scalable Resynchronization



1. Create database links between databases
2. Create directories pointing to datafiles
3. Stop replication to site 5
4. Ensure tablespaces are read-only
5. Transfer the data files of each tablespace to the remote system
6. Import tablespaces metadata in the target
7. Make tablespaces read-write
8. Reconfigure Streams

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Summary

- The LCG Database Deployment Project (LCG 3D) has set up a world-wide distributed database infrastructure for LHC
  - some 124 RAC nodes = 450 CPU cores at CERN + several tens of nodes at 10 partner sites are in production now
- Monitoring of database & streams performance has been implemented building on grid control and strmmon tools
  - key to maintain and optimise any larger system
- Important optimizations have been implemented in order to increase the Streams performance
  - collaboration with Oracle as part of the CERN openlab has been essential and beneficial for both partners

Questions?