Network performance optimization of database replications at WLCG

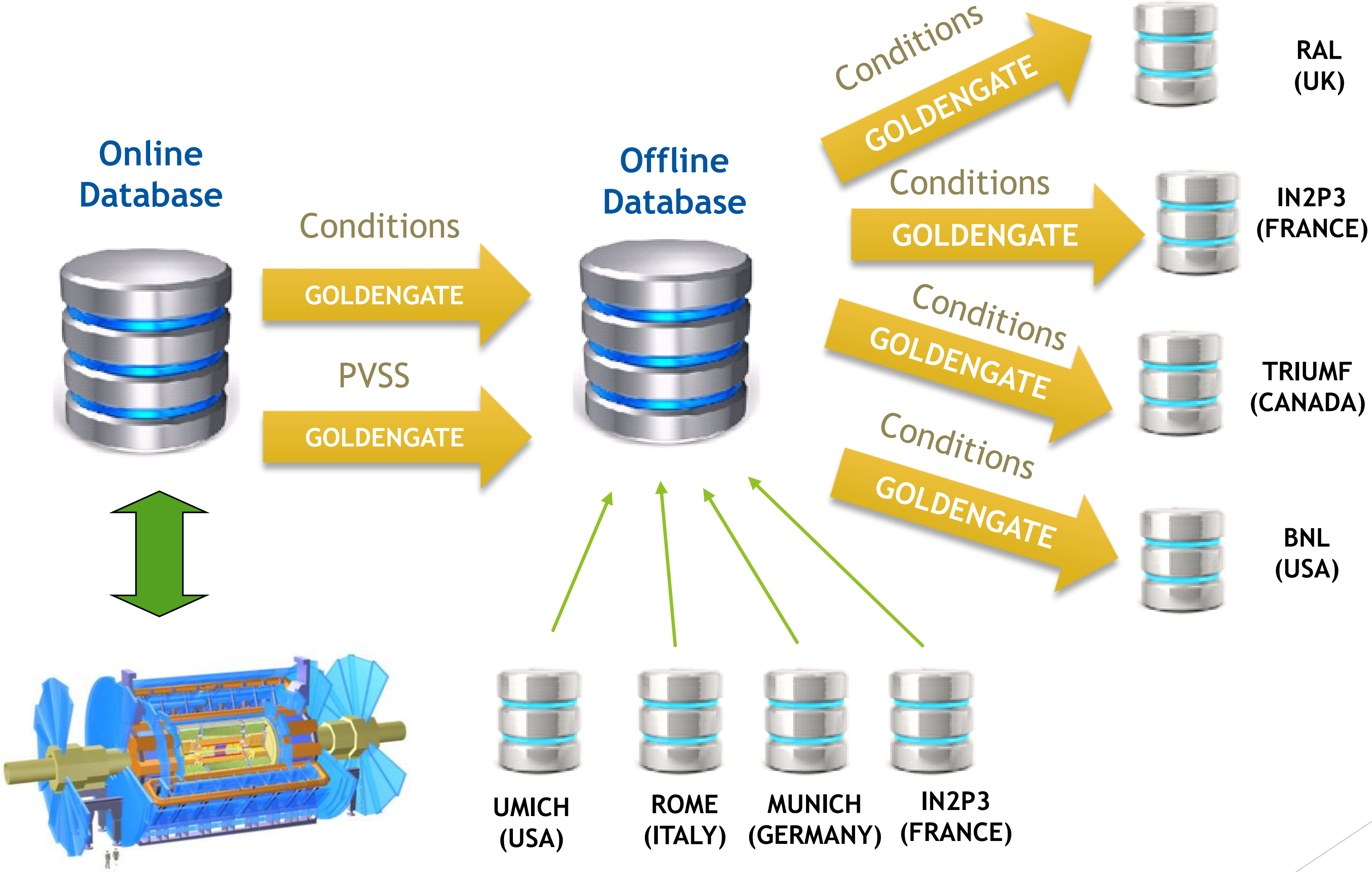
Lorena Lobato Pardavila

Thematic School of Computing 2016

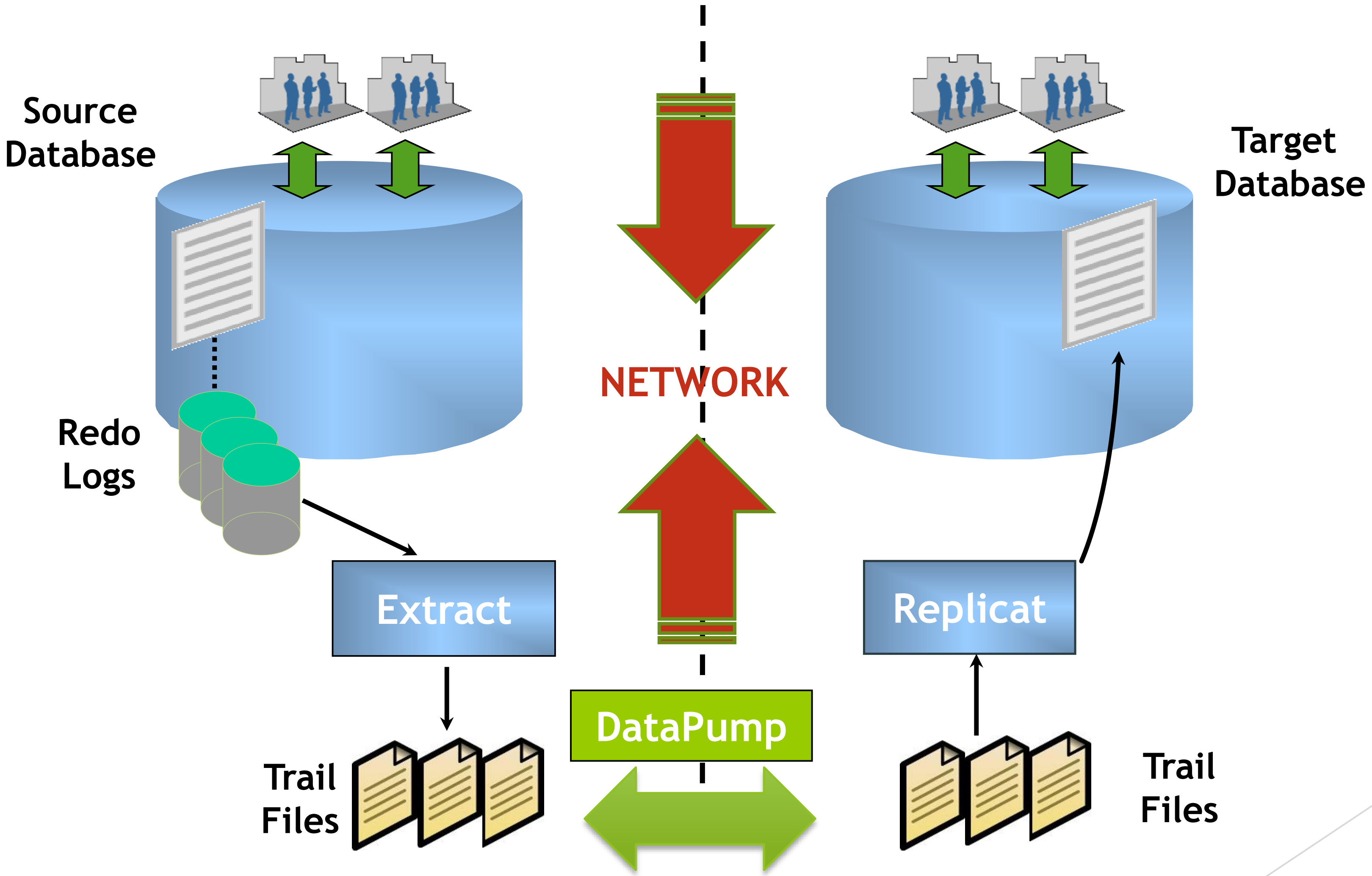
OUTLINE

- Database replication…what is that?
- How do we configure the database replication technology?
- Problem, which problem?
- Which tests were carried out?
- How the problem was solved out?
- Conclusions

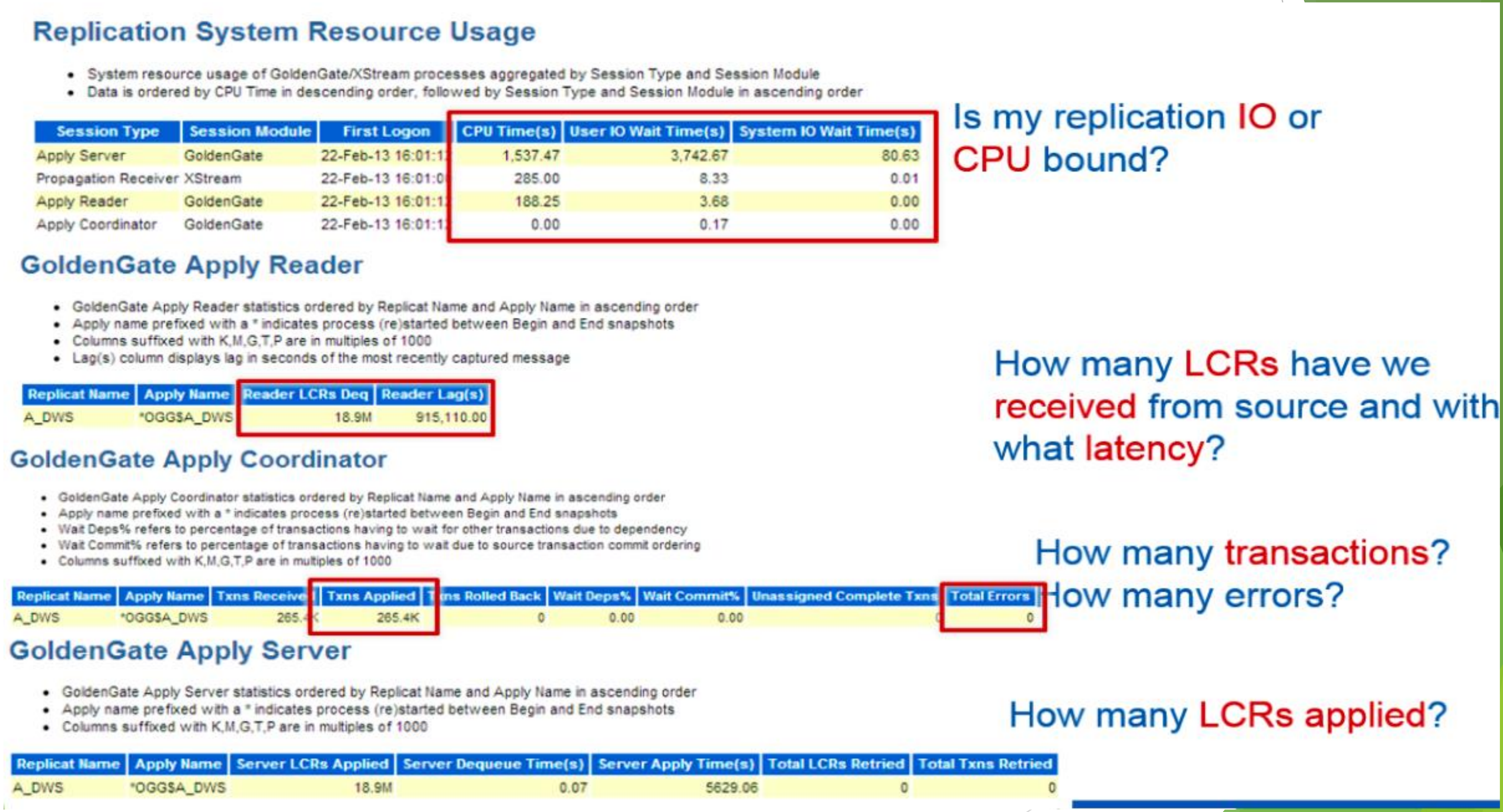
Database Replication…what is that?



How do we configure the database replication technology?



Problem, which problem?



Replication System Resource Usage

- System resource usage of GoldenGate/XStream processes aggregated by Session Type and Session Module
- Data is ordered by CPU Time in descending order, followed by Session Type and Session Module in ascending order

Session Type	Session Module	First Logon	CPU Time(s)	User IO Wait Time(s)	System IO Wait Time(s)
Apply Server	GoldenGate	22-Feb-13 16:01:1	1,537.47	3,742.67	80.63
Propagation Receiver	XStream	22-Feb-13 16:01:0	285.00	8.33	0.01
Apply Reader	GoldenGate	22-Feb-13 16:01:1	188.25	3.68	0.00
Apply Coordinator	GoldenGate	22-Feb-13 16:01:1	0.00	0.17	0.00

Is my replication IO or CPU bound?

GoldenGate Apply Reader

- GoldenGate Apply Reader statistics ordered by Replicat Name and Apply Name in ascending order
- Apply name prefixed with a * indicates process (re)started between Begin and End snapshots
- Columns suffixed with K,M,G,T,P are in multiples of 1000
- Lag(s) column displays lag in seconds of the most recently captured message

Replicat Name	Apply Name	Reader LCRs Deq	Reader Lag(s)
A_DWS	*OGG$A_DWS	18.9M	915,110.00

How many LCRs have we received from source and with what latency?

GoldenGate Apply Coordinator

- GoldenGate Apply Coordinator statistics ordered by Replicat Name and Apply Name in ascending order
- Apply name prefixed with a * indicates process (re)started between Begin and End snapshots
- Wait Deps% refers to percentage of transactions having to wait for other transactions due to dependency
- Wait Commit% refers to percentage of transactions having to wait due to source transaction commit ordering
- Columns suffixed with K,M,G,T,P are in multiples of 1000

Replicat Name	Apply Name	Txns Receive	Txns Applied	Txns Rolled Back	Wait Deps%	Wait Commit%	Unassigned Complete Txns	Total Errors
A_DWS	*OGG$A_DWS	265.4K	265.4K	0	0.00	0.00	0	0

How many transactions?
How many errors?

GoldenGate Apply Server

- GoldenGate Apply Server statistics ordered by Replicat Name and Apply Name in ascending order
- Apply name prefixed with a * indicates process (re)started between Begin and End snapshots
- Columns suffixed with K,M,G,T,P are in multiples of 1000

Replicat Name	Apply Name	Server LCRs Applied	Server Dequeue Time(s)	Server Apply Time(s)	Total LCRs Retried	Total Txns Retried
A_DWS	*OGG$A_DWS	18.9M	0.07	5629.06	0	0

How many LCRs applied?

What tests were carried out?

- Generic workload -> 1000000, 3000000 and 6000000 DML operations
- Two different configurations: Remote and Normal. Both cases playing with BATCHSQL
  - ❑ RC -> (Controlled by CERN, No Data Pump)
  - ❑ NC -> (Replicat on Tier-1s side, Data Pump: configured with COMPRESS parameter)
- Due to **I/O wait times** -> **"trail fails" handling optimization.**
- If reaching maximum **CPU (90-100%)** -> additional Replicat process creation

  for **balancing and avoid bottlenecks**.
- All the Tier-1s have same **Kernel TCP** **parameters configuration**

How the problem was solved out?

Kernel TCP parameters

- **net.core.rmem_max**
- **net.core.wmem_max**
- **net.ipv4.tcp_rmem**
- **net.ipv4.tcp_wmem**
- **net.ipv4.tcp_window_scaling**

DATA PUMP parameter file

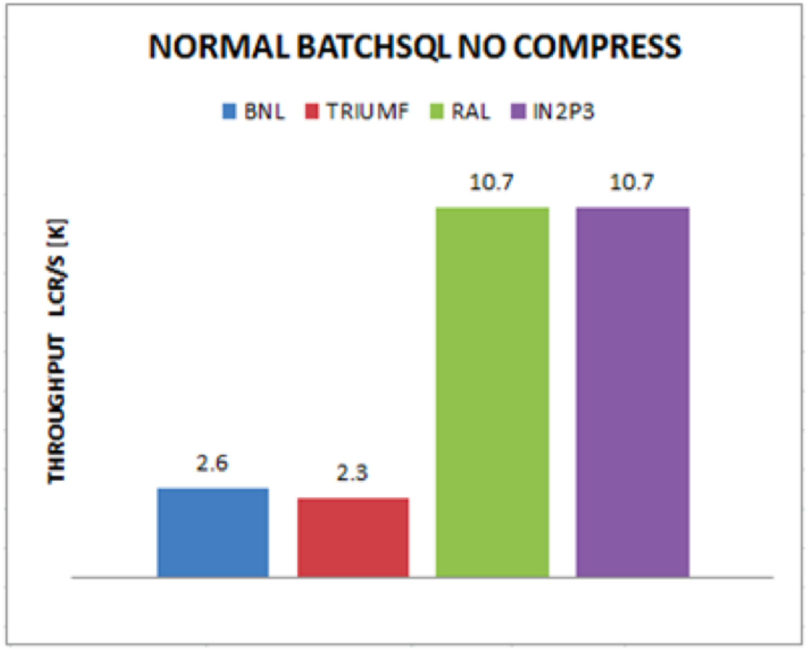
RMTHOSTOPTIONS
[, COMPRESS]
[, COMPRESSTHRESHOLD]
[, ENCRYPT *algorithm* [KEYNAME *key_name*]]
[, PARAMS *collector_parameters*]
[, STREAMING | NOSTREAMING]
[, TCPBUFSIZE *bytes*]
[, TCPFLUSHBYTES *bytes*]
[, TIMEOUT *seconds*]

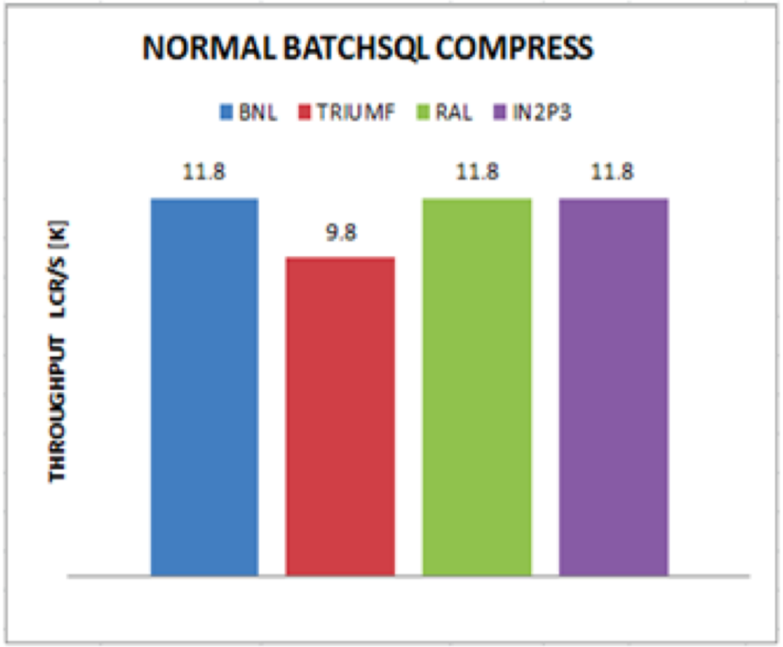
Results

- Example: Summary of Normal Configuration with BATCHSQL

Normal Configuration BACHSQL No Compress			
6000000	TIME (seconds)	LCR/s (K)	txns
BNL	6940	2.6	2.59
TRIUMF	7860	2.3	2.29
RAL	1680	10.7	10.71
IN2P3	1680	10.7	10.71

Normal Configuration BACHSQL Compress			
6000000	TIME (seconds)	LCR/s (K)	txns
BNL	1520	11.8	11.84
TRIUMF	1820	9.8	9.89
RAL	1520	11.8	11.84
IN2P3	1520	11.8	11.84





Conclusions

- Remote configuration was better because of manageability
- For Production workload BATCHSQL does not affect us on the performance
- There was not so much difference on the throughput between configurations
- COMPRESS makes performance faster but on the other hand CPU consumption is high
- RAL and IN2P3 are the faster ones. TRIUMF kept working slow comparing to other Tier-1s. Location is important
- TESTS in different levels are always important. The change improved availability and performance of the data replication services

Questions?

THANKS FOR YOUR ATTENTION!