Oracle CERN openlab Projects Status Review

Eric Grancher

IT-DES

Work from

**Anton Topurov,
Chris Lambert
and Eric Grancher**

Outline

- Achievements since last review
- openlab projects in the context of the service evolution
- Highlights 2007 Q1

Application Design, Development and Scalability on Oracle RAC

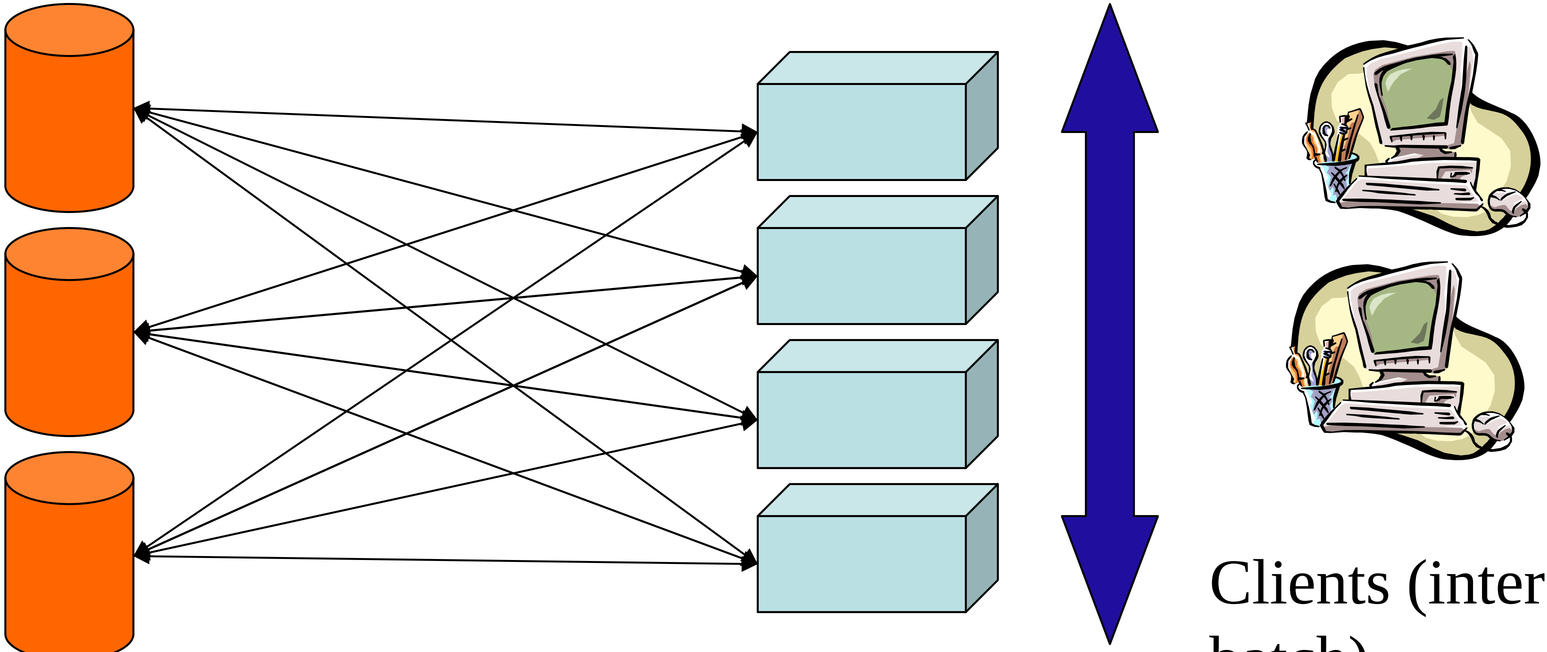
- The process of designing/tuning the applications for RAC scalability is not easy and straightforward
- CERN developers will need recommendations and guidance in order to produce RAC scalable software.

Objectives of the programme:

- To examine real CERN cases and to study RAC scalability
- To design and develop general techniques and recommendations to improve RAC scalability

[RAC scalability] RAC structure

- Shared disk infrastructure, all disk devices accessed from **all** servers



Disks

Database servers

Clients (interactive, batch)

Example of performance gain with RAC

- Commercial control application (critical for LHC and experiments)
- Archiving in Oracle
- Application without modifications: 100 “changes” per second
- CERN needs: 150 000 changes per second (x 1500)

- **Iterative process, based on Oracle’s “wait interface”**

Application RAC (1/6)

- Structure = table EVENTS_HISTORY (ELEMENT_ID, VALUE…)
- Each client “measures” input and registers history with a “merge” operation in the table EVENTS_HISTORY

Application RAC (2/6)

- 100 entries par second
- Initial state observation: database is waiting on the clients “SQL*Net message from client”
- Use of a generic library C++/DB
- Individual insert (one statement per entry)
- Update of a table which keeps “latest state” through a trigger

Application RAC (3/6)

- Changes: bulk insert to a temporary table with OCCI, then call PL/SQL to load data into history table
- From 100 to 2000 changes per second
- awrrpt_1_5489_5490.html
- Now top event: “db file sequential read”

Event	Waits	Time(s)	Percent Total DB Time	Wait Class
db file sequential read	29,242	137	42.56	User I/O
enq: TX - contention	41	120	37.22	Other
CPU time		61	18.88	
log file parallel write	1,133	19	5.81	System I/O
db file parallel write	3,951	12	3.73	System I/O

Application RAC (4/6)

- changes: index usage analysis and reduction, table structure changes. IOT. Replacement of merge by insert. Use of “direct path load” with ETL
- Improvement: from 2000 changes per second to 16 000 changes per second
- Now top event: cluster related wait event
  test5_rac_node1_8709_8710.html

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
gc buffer busy	27,883	728	26	31.6	Cluster
CPU time		369		16.0	
gc current block busy	6,818	255	37	11.1	Cluster
gc current grant busy	24,370	228	9	9.9	Cluster
gc current block 2-way	118,454	198	2	8.6	Cluster

Application RAC (5/6)

- Changes: each “client” receives a unique number. Partitioned table. Use of “direct path load” to the partition with Extracting, Transforming and Loading
- Improvement: from 16 000 changes per second to 150 000 changes per second
- Now top event : “freezes” once upon a while

rate75000_awrrpt_2_872_873.html

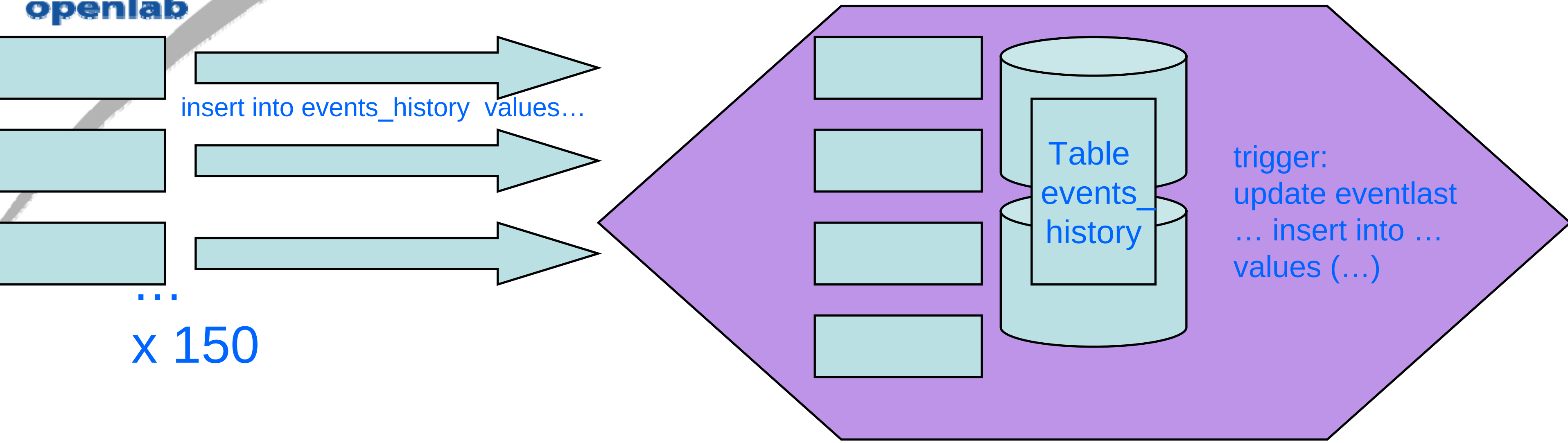
Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
row cache lock	813	665	818	27.6	Concurrency
gc current multi block request	7,218	155	22	6.4	Cluster
CPU time		123		5.1	
log file parallel write	1,542	109	71	4.5	System I/O
undo segment extension	785,439	88	0	3.6	Configuration

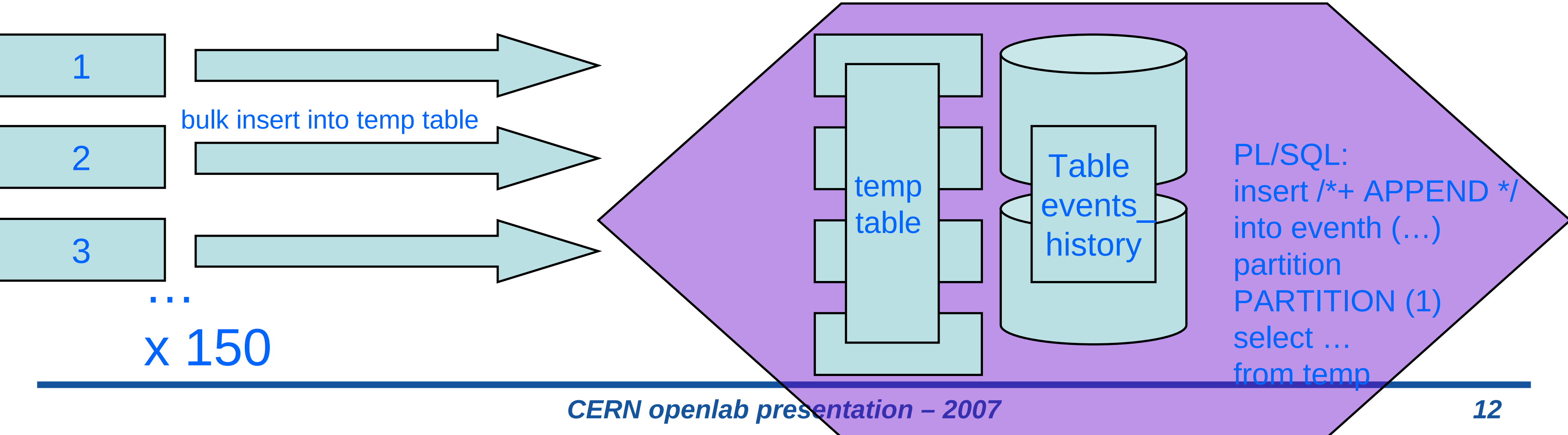
Application RAC (6/6)

- Identification: ASM space allocation is blocking some operations
- Changes: space pre-allocation, background task.
- Allows to keep steady rate

- Conclusion: from 100 changes per second to 150 000 changes per second
- 6 nodes RAC (dual CPU, 4GB RAM), 32 disks SATA with FCP link to host
- 4 months, re-writing of part of the application with changes interface (C++ code), changes of the database code (PL/SQL), schema change, numerous work sessions
- Initial help of an Oracle consultant

Application RAC / summary





Oracle Data Guard Automatic Failover

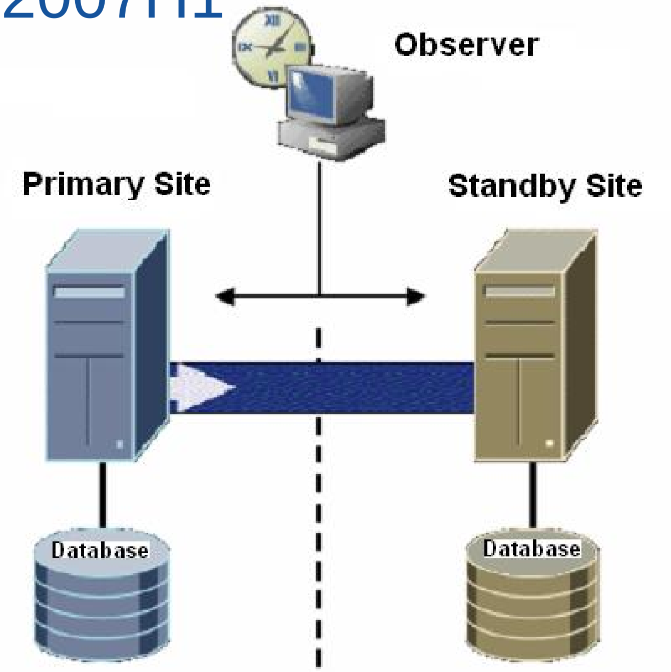
Objectives of the programme:

- To test following Data Guard solutions on RDBMS 10gR2 :
  - ✓ Automatic Failover
  - ✓ Inter patchset SQL Apply

  - To test Data Guard Automatic Failover mechanism with focus on:
    - ✓ Data size
    - ✓ Time to switch
- To deploy in production Data Guard Automatic Failover mechanism on selected CERN service, in order to reduce downtime implied by major software / hardware issues and upgrades

[DataGuard] Status / results

- DataGuard automatic failover is now well understood, interaction with Oracle helped to identify how to set connect time failover / and use DB_ROLE_CHANGE event
- Will be implemented in production as a core building block for database servers in 2007H1



Enterprise Manager

- New openlab subject
- Managing a large number of Oracle targets coupled with a comparatively small number of personnel
- January 2007 achievements:
  - Global report of all CERN Oracle installation which require installation of the latest CPU patch
  - Report of databases for which one or more datafiles have not been backed up in the past x days

Service evolution in the openlab context

- We have started the move of administrative applications to “AIS RAC”, applications migration started December 2006
- 10gR2 with 4 nodes (4 CPUs, 16GB per node)
- 64 bits Linux (RedHat Enterprise Linux 4 x86-64).
- No issue with RHEL4 64bits / Oracle 64 bits Linux

New hardware

- CASTOR (CERN mass storage system) uses Oracle database as central architectural component
- Multi-core systems (Woodcrest) in pre-production
- Relatively high-performance Network Attached Storages in pre-production

2007 Q1-Q2 plans

- Oracle Enterprise Manager (CERN coordinator: Chris Lambert)
- Oracle DataGuard (CERN coordinator: Anton Topurov)
- Application Design, Development and Scalability on Oracle RAC (CERN coordinator: Anton Topurov)