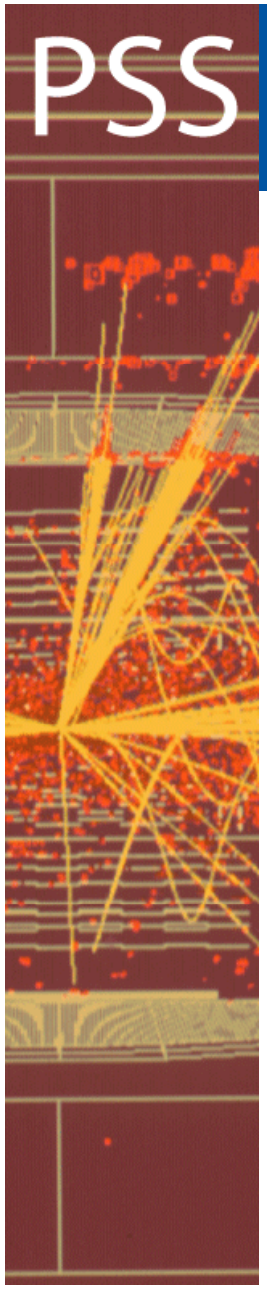


PSS Physics Services Support

CERN IT Department

DB Performance Tuning in a RAC Environment for Administrators

Luca Canali, CERN IT

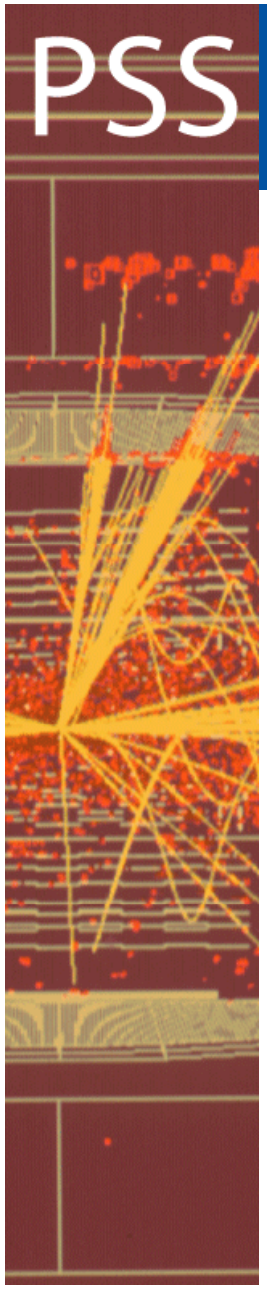
WLCG Service Reliability Workshop,

26th-30th Nov 2007

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

LCG

CERN



Outline

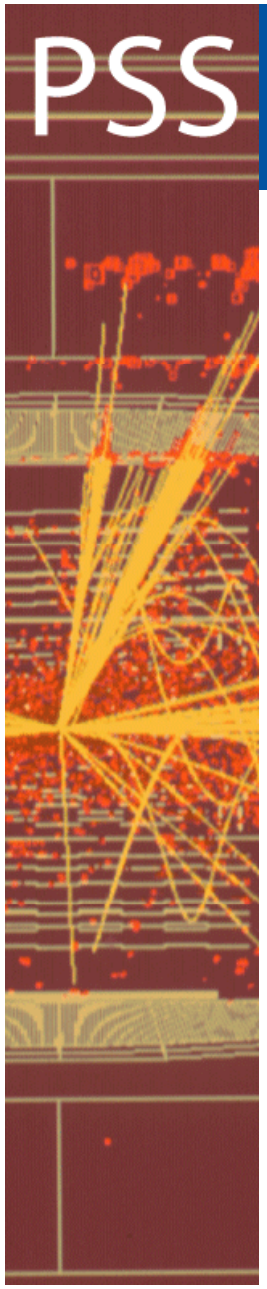
- Discussion driven by production experience
- DB Service architecture and deployment
  - Being proactive against performance issues
- Monitoring and reporting
  - Quickly identify performance issues
  - Possibly stop the issue before it impacts SL
- Reactive performance tuning
  - Case studies from CERN production
- Capacity planning and new HW assessment
  - Our experience

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

Architecture and Deployment:

design for performance

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



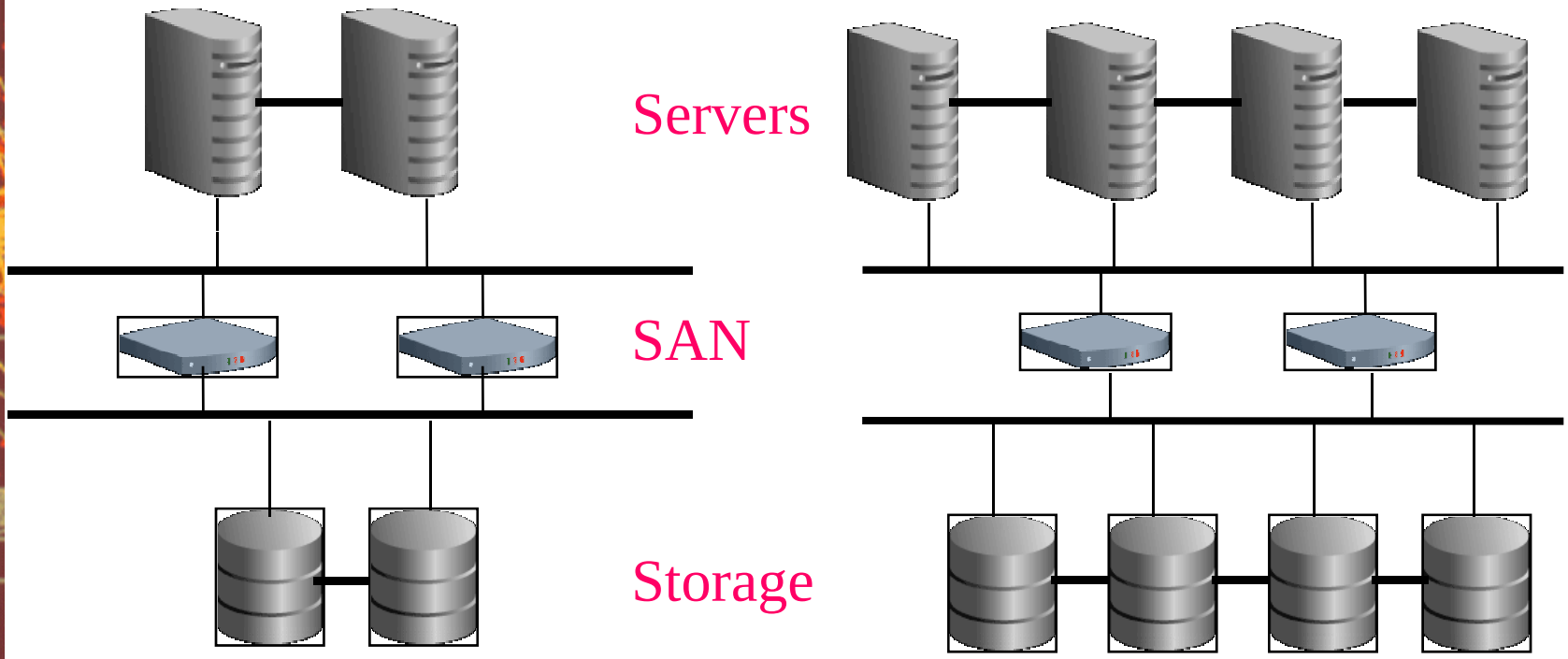
Architectural Goals

- Operational requirements:
  - Performance, High Availability, Scalability, Manageability, Security
- A good DB architecture
  - Avoids single points of failure
  - Avoids serialization points
  - Allows for change
  - Allows for growth

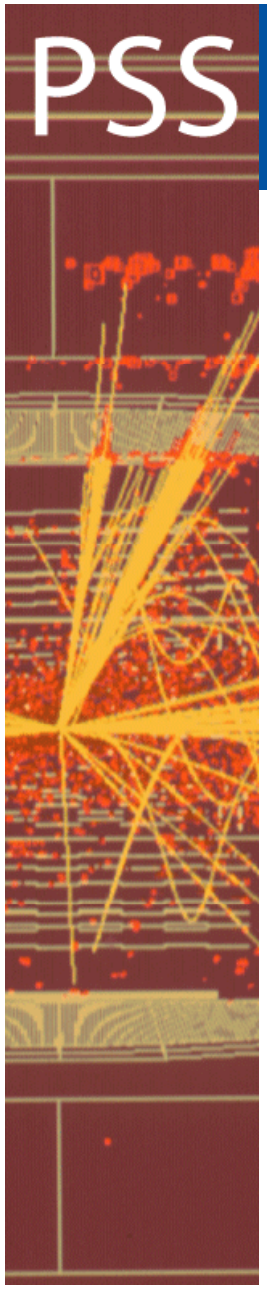
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Clustered DBs for Physics

- Clusters of redundant low-cost components



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

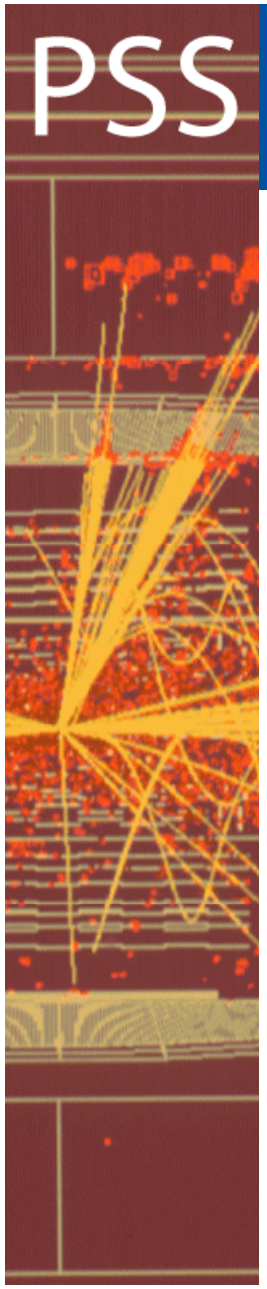


Keep it simple: homogeneity

CERN IT Department

- Keep your environments as homogeneous as possible
  - Same type of HW
    - 2 x Xeon 4GB RAM
  - Same version OS
    - Currently RHEL 4 Update 5, 32 bit
  - Same Oracle version
    - Currently 10.2.0.3 32 bit
  - Yet allow for change
    - Quadcore architecture, 64 bit Oracle, etc
    - Changes need validation before going to production

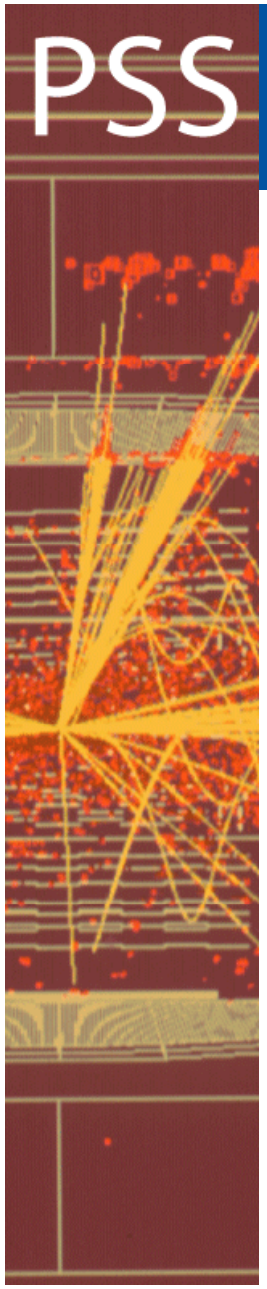
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Keep It Simple: Database Setup

- Use defaults for DB config when possible
  - Then reactively change when necessary
  - Example: spfile parameters, statistics collection
- Standardize the installation procedure
  - Publish, share and evolve with as wiki
  - Example: Installation procedure: http://twiki.cern.ch/twiki/bin/view/PSSGroup/Installation_verbose
  - Similar pages for streams, etc

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



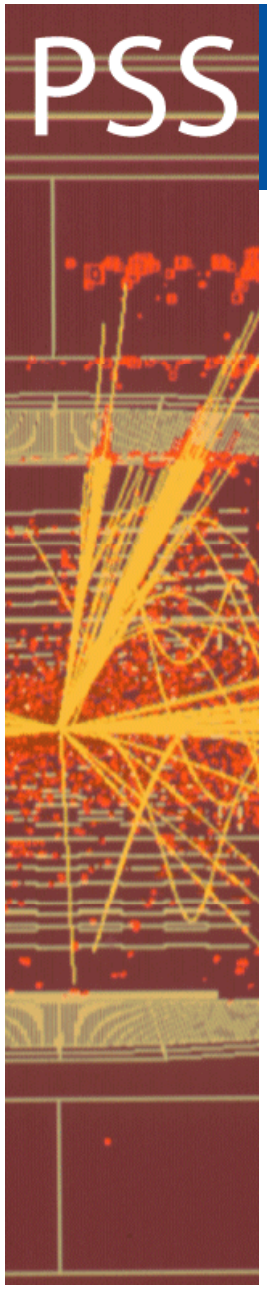
Keep It Simple: Consolidation

CERN IT Department

- Consolidation of DBs (schemas) in a single Oracle DB
  - RAC cluster and ‘Oracle Services’
  - One cluster consolidates DBs per experiment
- Open question: what are the limits of the model?
  - Technology limitations (move forward in time)
    - When is a DB ‘too big’?
    - When is the consolidation prevented by Application behaviour (DSS and OLTP mixture)?

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

CERN



Application Lifecycle Model

- Development DB
- Pre-production tests (integration DB)
- Performance and concurrency tests (test DB)
- Production

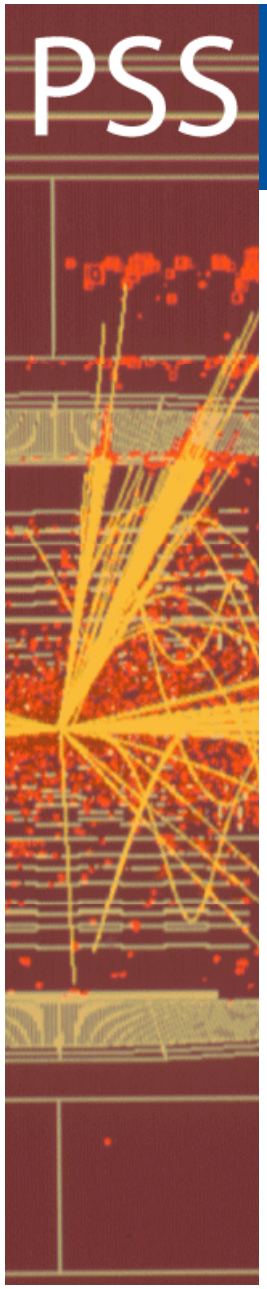
- In other words:
  - We don’t allow users to develop in production
  - We promote extensive tests
  - Tests done in a production-like environment

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

Monitoring and Reporting,

monitor performance

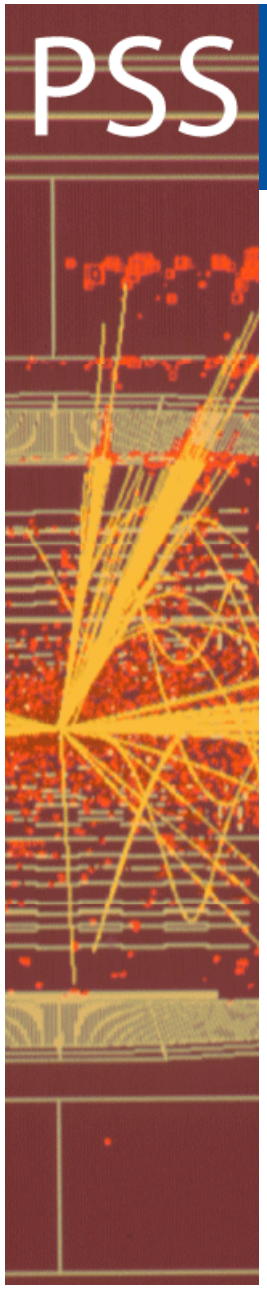
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Performance Rules

- Andrew Holdsworth's two simple performance rules:
  - The key to good performance
    - **You run good execution plans**
    - **There are no serialization points**
  - Without these all bets are off!

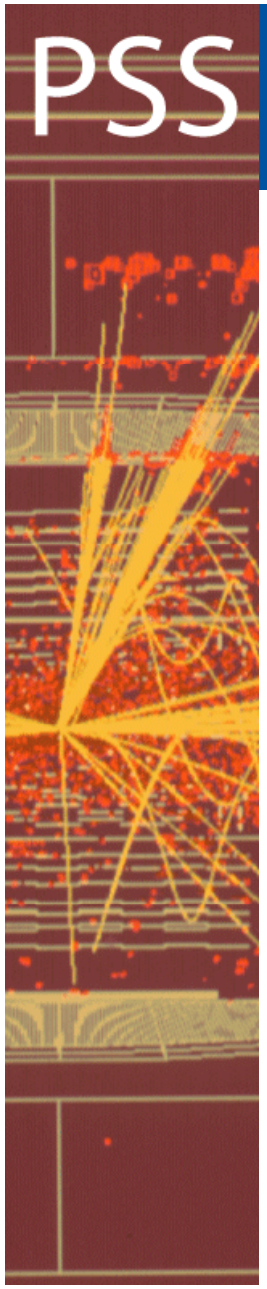
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



What to Monitor

- Gather data to identify potential bottlenecks
  - OS
  - DB
  - Application specific metric
- Identify DB operations that are too slow
  - inefficient SQL
  - Use the ‘response time’ characterization (i.e. wait events) to identify where DB time is spent
- Unfortunately there is no single DB ‘speed indicator’

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



Systems Bottlenecks

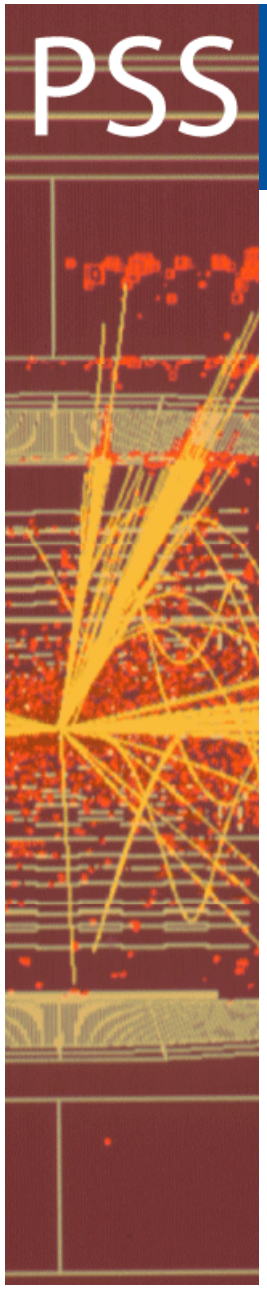
- Simple and often useful indicators of performance
  - CPU
  - I/O (IOPS and throughput)
  - Network
  - Swap (memory)
- Tools
  - OEM
  - Lemon
  - Sqlplus

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

CPU and Load

- The amount of CPU and load on the system are the simplest metric to check
  - OEM performance page will show the cluster load
- NOTE: on RAC system bottlenecks have to be checked on each node
  - The cluster average can be misleading

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



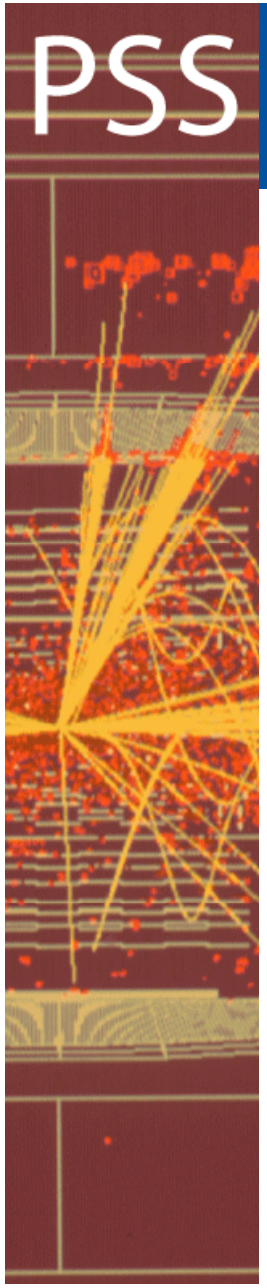
Case: High Load on One Node

```
select
   ins.instance_name,ins.host_name,round(os.value
   ,2) load
from gv$osstat os, gv$instance ins
where os.inst_id=ins.inst_id and
   os.stat_name='LOAD'
order by 3 desc
```

INSTANCE_NAME	HOST_NAME	LOAD
test1	**host1.cern.ch**	**16.73**
test2	host2.cern.ch	.37
test3	host3.cern.ch	.20
test4	host4.cern.ch	.49
test5	host5.cern.ch	.69
test6	host6.cern.ch	.79

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

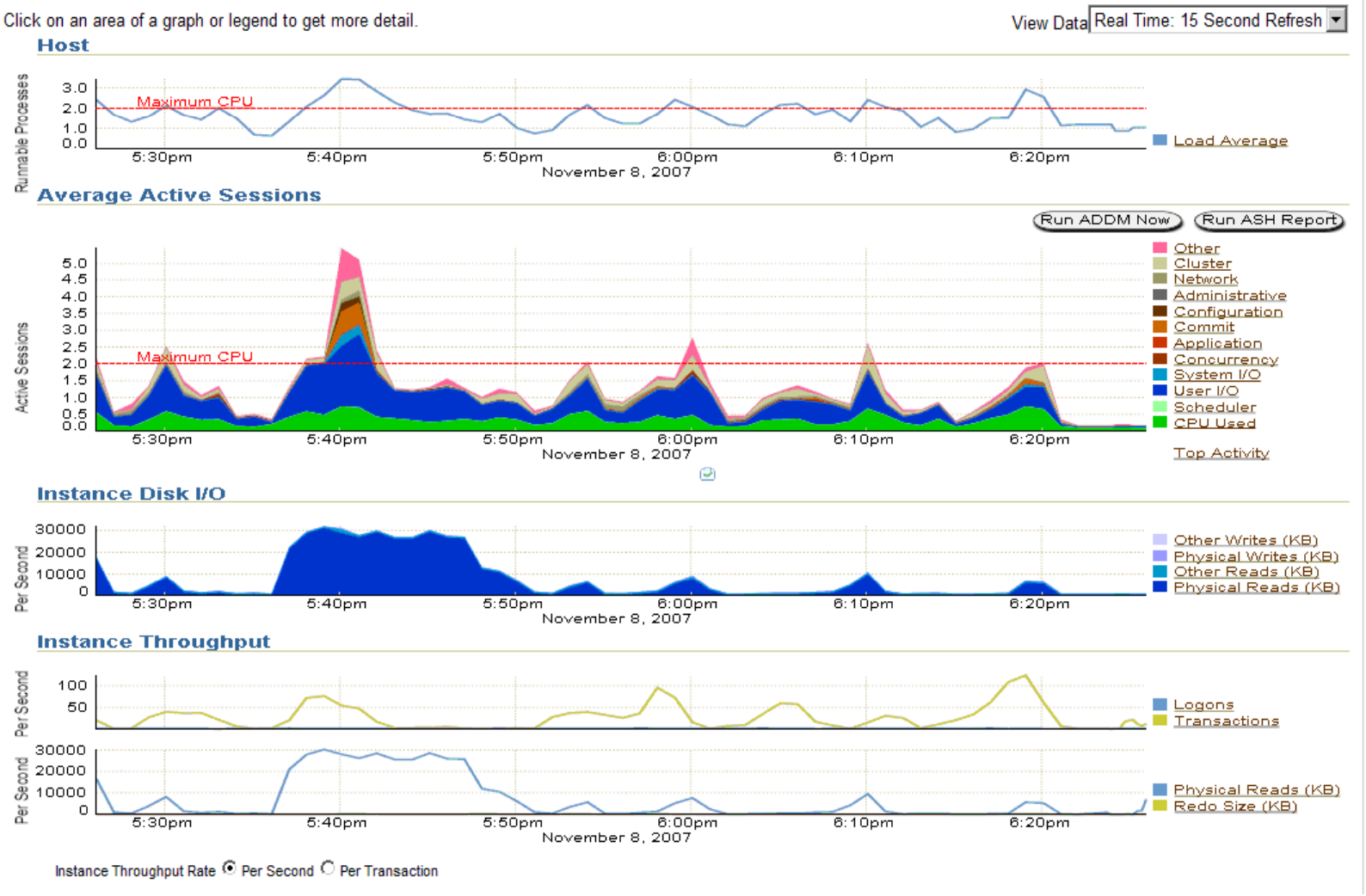
DB Performance Investigations



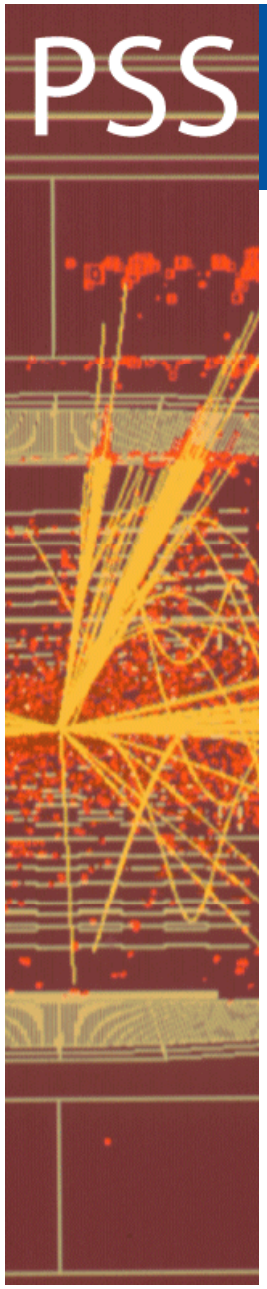
- Structured approach
- Use the response-time method
  - Active session history (ASH)
  - OEM
  - Bottlenecks appear at the instance level: query GV$SESSION
- For each active session identify
  - Response time characteristic (CPU, wait time)
  - Most active sql statements

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

OEM Performance Tab



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



Case: One Slow Query Executed Multiple Times

- Most of the times there is one sql or application that is responsible for bad performance
  - Can be identified from the 'top activity' OEM page

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

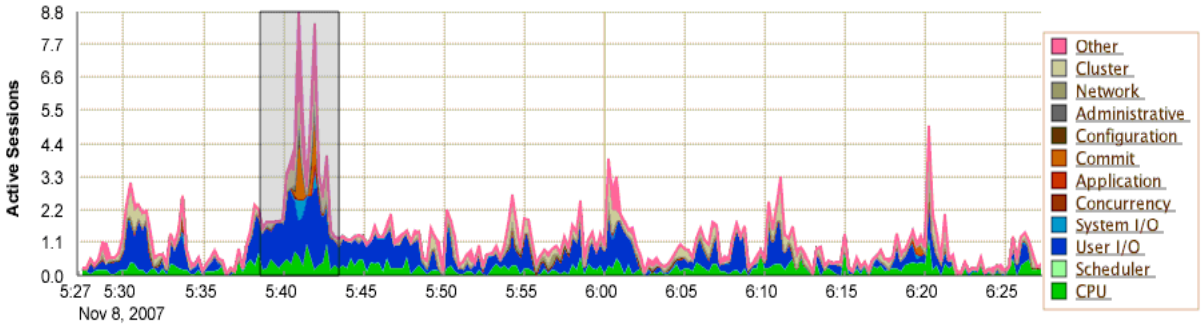
OEM Top Sessions

Top Activity

Switch Database Instance atlr.cern.ch_atlr1 Go

Drag the shaded box to change the time period for the detail section below.

View Data Real Time: 15 Second Refresh



Detail for Selected 5 Minute Interval

Start Time **Nov 8, 2007 5:38:34 PM MET**

Run ASH Report

Top SQL

Schedule SQL Tuning Advisor | Create SQL Tuning Set

Select All | Select None

Select	Activity (%)	SQL ID	SQL Type
☐	42.78	3t4zyft1ycmp7	SELECT
☐	11.30	aq8utg96sc8kq	SELECT
☐	8.30	0pcmq7hryn2x0	SELECT
☐	5.87	aahvgmuar9z07	SELECT
☐	4.15	7yhvgrsw7x8d7	SELECT
☐	2.72	7grmrsp8dy3yd	SELECT
☐	1.72	2f3ddpjygkxhv	SELECT
☐	1.72	9scc29h43m5ap	INSERT
☐	1.29	7hzn20h90y9rk	SELECT
☐	1.00	6khrc3pvyw9j6	SELECT

Schedule SQL Tuning Advisor | Create SQL Tuning Set

Total Sample Count: 699

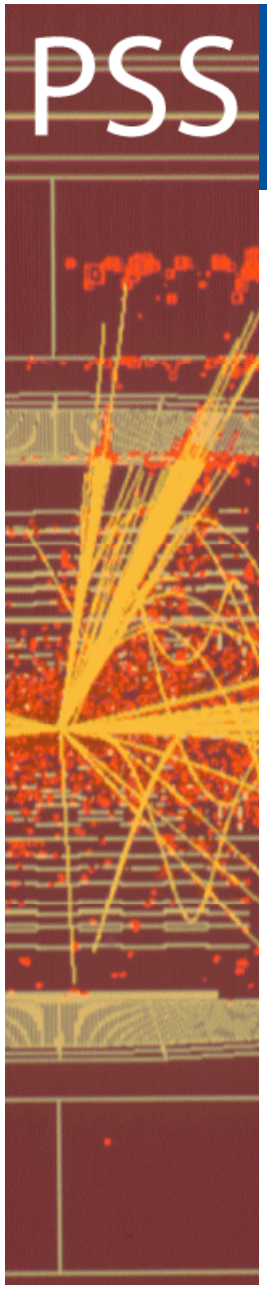
Top Sessions

View Top Sessions

Activity (%)	Session ID	User Name	Program
29.72	635	ATLAS_PVSS_READER	root.exe@roata01 (TNS V1-V3)
5.77	579	ATLAS_PS_W3	httpd@ccwbsn01.in2p3.fr (TNS V1-V3)
4.77	675	ATLAS_DASHBOARD_DM_WRITER	data.stats.collection@lxarda11.cern.ch (TNS V1-
4.08	673	ATLAS_PS_W3	
3.48	661	ATLAS_PS_W1	python@atlas002.uta.edu (TNS V1-V3)
3.38	874	SYS	oracle@itrac21.cern.ch (LGWR)
3.28	545	ATLAS_DQ2_R	httpd.worker@lxb7239.cern.ch (TNS V1-V3)
3.28	859	SYS	oracle@itrac21.cern.ch (ARC0)
3.18	858	SYS	oracle@itrac21.cern.ch (ARC1)
2.78	636	ATLAS_DQ2_W	httpd.worker@lxb7238.cern.ch (TNS V1-V3)

Total Sample Count: 1,006

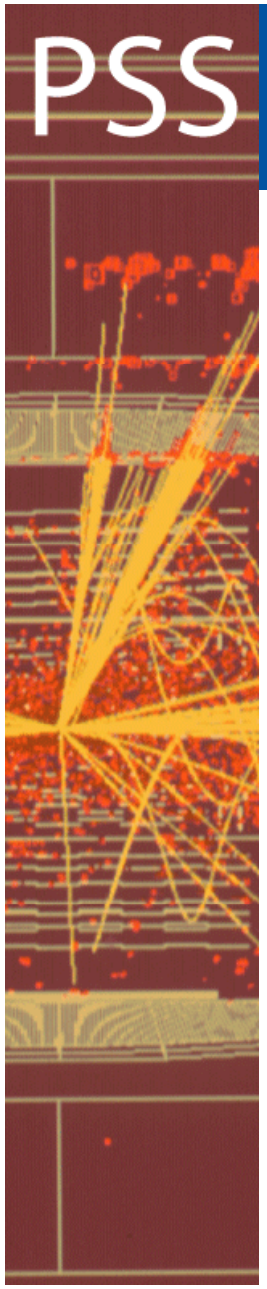
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Environment Specific Tips

- For an OLTP system
  - Avoid full scans
    - Use GV$SESSION_LONGOPS to identify them
  - Avoid parallel full scan operations
  - Beware of long-running SQL
  - Beware of DML lock contention
- With experience find rules to quickly identify other unwanted activities

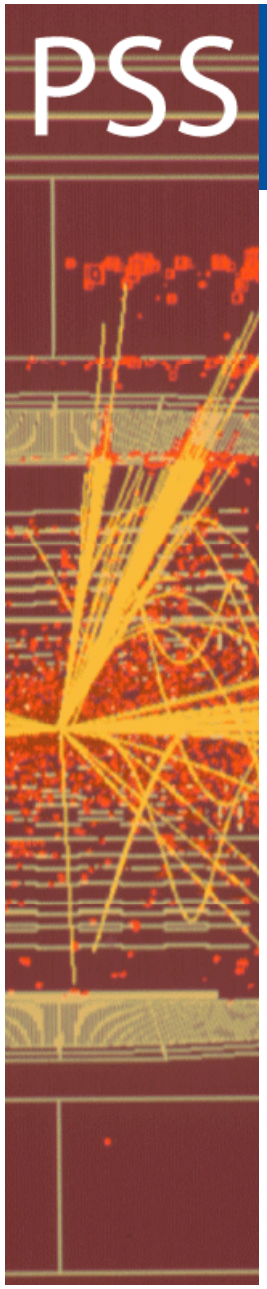
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Monitoring Performance with SQL Scripts

- Monitoring performance ‘with SQL scripts’:
  - GV$SESSION -> list the active sessions, their elapsed time and wait events
  - GV$SESSION_LONGOPS -> find full scans
  - GV$OSSTAT
- Quickly give an overview of the performance
  - Identify serialization and slow SQL
  - Details on each RAC node

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



Result of the analysis

- Determine if
  - The bottleneck is at the CPU level
  - The bottleneck is with serialization ‘wait events’ (I/O bound or cluster-event bound, etc)
  - Can you identify an SQL statement / User that is causing the high load problem?

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

Weekly Experiment Report

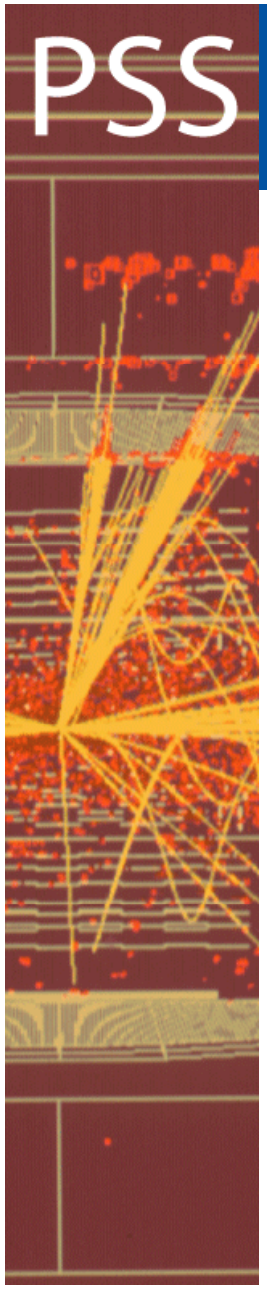
- A weekly report is produced and sent to the experiments on the activity of their DBs:
  - Reports on the number of connections and system resources used (from SYS.AUD$)
  - Reports on the number of active sessions per hour (workload) for each service (using DBA_ACTIVE_SESSION_HISTORY)
  - Space Usage per application
  - Work in progress: graphs of systems resources and utilisation metrics

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

Reactive Performance Tuning:

Case studies

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



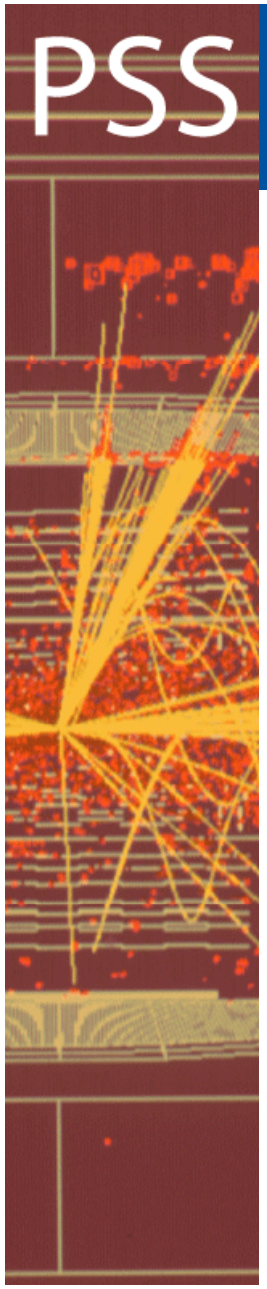
Typical IO-related wait events

CERN IT Department

- Selected IO related events (the ‘blue events’ in OEM):
  - db file sequential read
  - db file scattered read
  - log file synch

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

CERN



Case: I/O bound, full scan 1/2

- Long running query because of a large table full scan
  - db file scattered read is the main wait event
  - Shows up in v$session_longops
  - Execution plan show full scan operations
  - Often an index is missing or cannot be used

```
SELECT TO_CHAR(current_timestamp AT TIME ZONE
   'GMT', 'YYYY-MM-DD HH24:MI:SS TZD') AS
   curr_timestamp, COUNT(username) AS
   failed_count
FROM  sys.dba_audit_session
WHERE returncode != 0 AND TO_CHAR(timestamp,
   'YYYY-MM-DD HH24:MI:SS') >=
   TO_CHAR(current_timestamp - TO_DSINTERVAL('0
   0:30:00'), 'YYYY-MM-DD HH24:MI:SS')
```

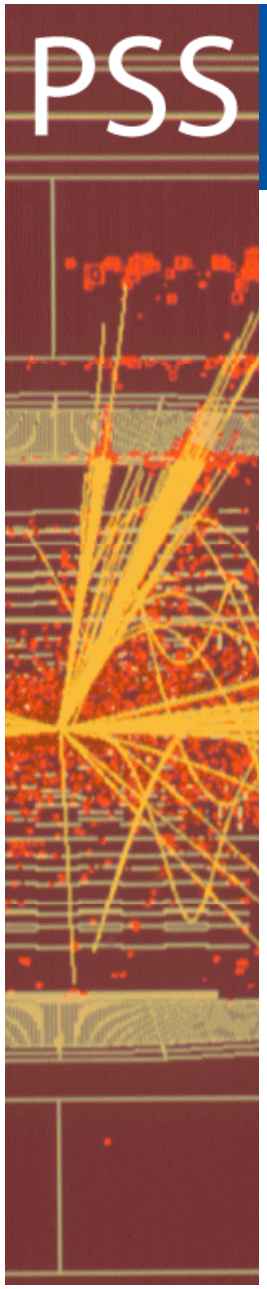
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Case: I/O bound, full scan 2/2

```
-------------------------------------------------------------
| Id  | Operation                    | Name                   |
-------------------------------------------------------------
|   0 | SELECT STATEMENT             |                        |
|   1 |  SORT AGGREGATE              |                        |
|*  2 |   HASH JOIN RIGHT OUTER      |                        |
|   3 |    INDEX FULL SCAN           | I_STMT_AUDIT_OPTION_MAP|
|*  4 |    HASH JOIN RIGHT OUTER     |                        |
|   5 |     INDEX FULL SCAN          | I_SYSTEM_PRIVILEGE_MAP |
|*  6 |     HASH JOIN RIGHT OUTER    |                        |
|   7 |      INDEX FULL SCAN         | I_SYSTEM_PRIVILEGE_MAP |
|*  8 |      HASH JOIN RIGHT OUTER   |                        |
|*  9 |       INDEX RANGE SCAN       | I_AUDIT_ACTIONS        |
|* 10 |       TABLE ACCESS FULL      | AUD$                   |
-------------------------------------------------------------
```

- What's wrong with this query, and how to fix it?

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



Case: I/O bound, index read 1/3

- A query that reads data via an index
  - db file sequential read is the main wait event
  - Does not show up in v$session_longops
  - Execution plan shows index range scan operations
  - Often better than full scans but not always

```
select  distinct("UserId")
from JOB,TASK where
JOB."TaskId"=TASK."TaskId" and
  TASK."TaskTypeId"=1 and
  "DboardFirstInfoTimeStamp">='31-DEC-06
12.00.00.000000 AM' and
  "DboardFirstInfoTimeStamp" < '01-JUL-
  07 12.00.00.000000 AM';
```

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

Case: I/O bound, index read 2/3

```
-----------------------------------------------------
| Id  | Operation                          | Name
-----------------------------------------------------
|   0 | SELECT STATEMENT                   |
|   1 |  HASH UNIQUE                       |
|*  2 |   FILTER                           |
|*  3 |    HASH JOIN                       |
|   4 |     TABLE ACCESS BY INDEX ROWID| JOB
|*  5 |      INDEX RANGE SCAN              | ..TIMESTAMPIDX..
|*  6 |     TABLE ACCESS FULL              | TASK
-----------------------------------------------------
```

..takes more than 1 hour

Job is a table of 18M rows

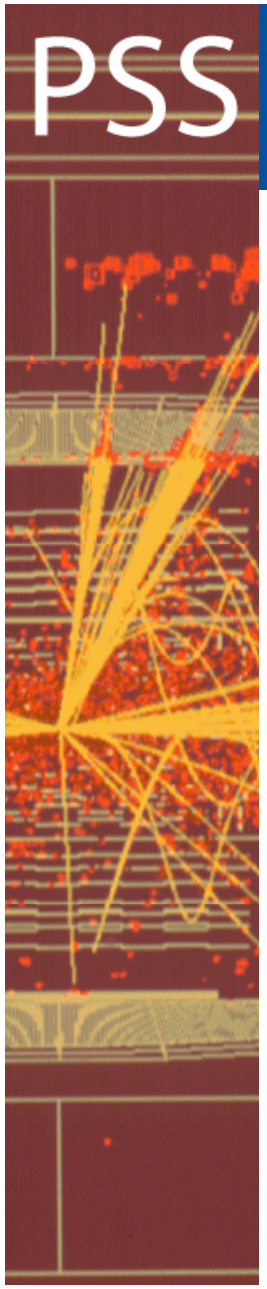
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Case: I/O bound, index read 3/3

```
-------------------------------------
| Id  | Operation             | Name |
-------------------------------------
|   0 | SELECT STATEMENT      |      |
|   1 |  HASH UNIQUE          |      |
|*  2 |   FILTER              |      |
|*  3 |    HASH JOIN          |      |
|*  4 |     TABLE ACCESS FULL| JOB  |
|*  5 |     TABLE ACCESS FULL| TASK |
-------------------------------------
```

..forcing a full scan brings the execution time down to a few minutes

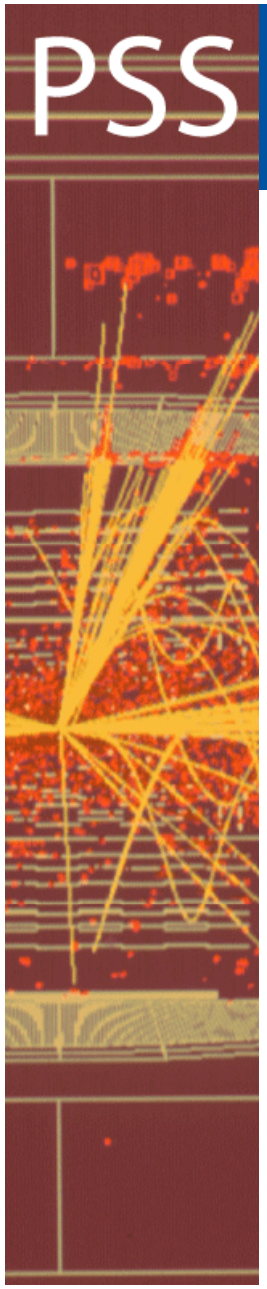
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Cluster wait events

- Cluster events. Example:
  - gc current request, gc cr request
  - gc current/cr buffer busy
  - gc buffer busy, etc
- Contention with the global cache
  - Global cache is faster then IO
  - RAC global cache is a distributed cache with requires locks and in general has a measurable overhead
  - The interconnect can be a bottleneck but it is rare in OLTP (it's mainly global lock handling)

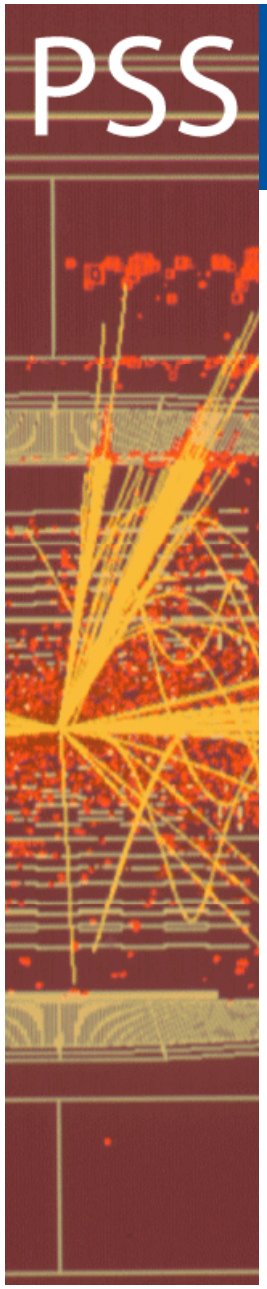
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Case: tuning cache contention

- A file transfer application performs cleanup of the queues (large DML operations) on a regular basis
  - Several performed DML concurrently
  - Serialization point: ‘current’ blocks shipped across RAC nodes
- Solution 1: restrict the application to 1 node
  - Not scalable but works
- Solution 2: change the application such that only one ‘daemon’ is in charge of the DML

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



Case: tuning cache contention

- PVSS needs to concurrently write data into the ‘event history’ table from multiple clients
  - Contention due to current blocks shipped back and forth cluster nodes
- Solution 1: partition the event history table by client id
  - each client connects to a preferred node and uses 1 partition
- Solution 2: perform a larger number of installations
  - Each application instance has a preferred node

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

DBA actions against cluster contention

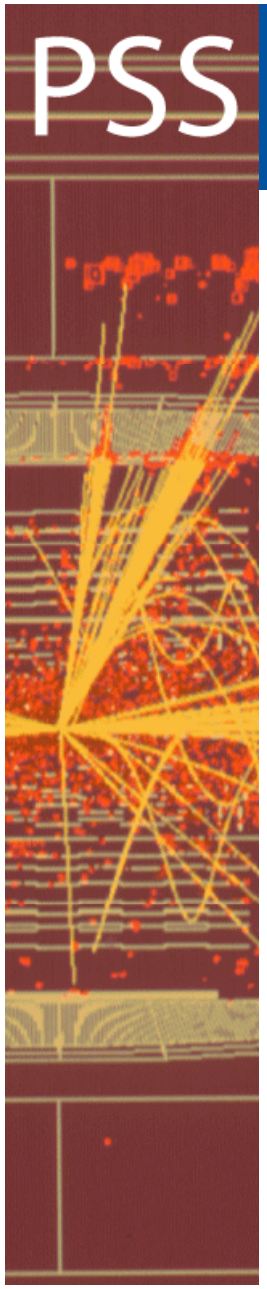
- The reactive tuning (short term) solution is to run applications suffering from cluster contention on 1 node only
  - Use RAC Services
  - Deployment of ‘quadcores’ will make this easier
- Use RAC service ‘preferred’ nodes to manually load balance
  - For large clusters one node can be designated as ‘empty’ and the first failover node of non load balanced services

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

Transactional Locks

- Non different than in single instance
  - It's an application issue
  - Reactive tuning should be focused on understanding 'the lock tree' and then on selectively killing blocking sessions
- From production:
  - Often the locking issues can be solved by creating indexes on foreign keys

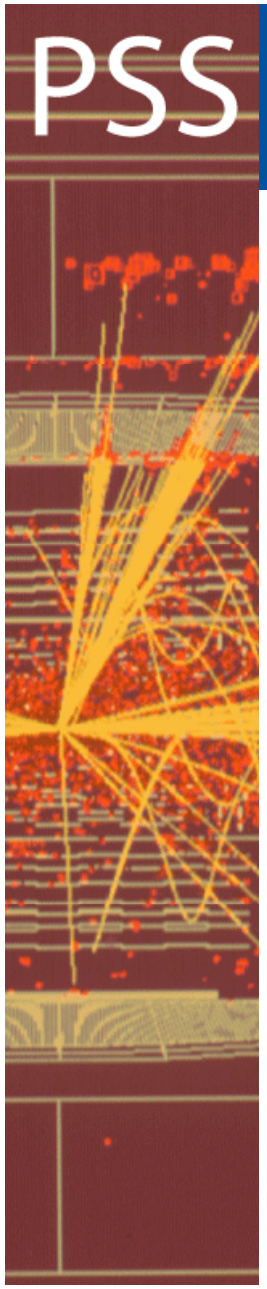
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Case: execution plan change

- An SQL statement in production suddenly becomes very slow and brings a RAC node to saturation
- Upon investigation it is found that its execution plan has changed
- A reparsing of the statement is forced and performance are back to normal
  - ‘alter system flush shared_pool’ is one possible method of forcing reparsing.

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



Best-laid plans..

- Plans can change because of
  - Data change and statistics collection
  - Bind variable peeking
  - Small schema or data changes
  - Oracle version change/upgrade
- It appears often has a ‘time bomb’ since a new plan is picked up only when the statement is (re)parsed

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

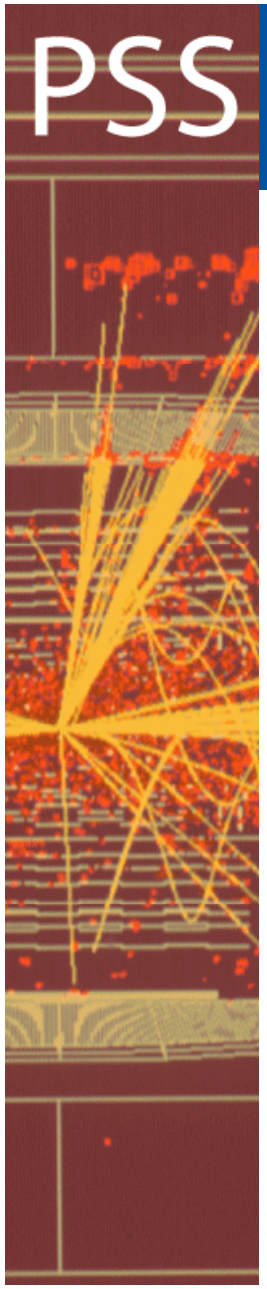
Case: changing because of bind variable peeking

```
select <query text> from tab1, tab2,…
where id = :var1
```

The first time this statement is parsed it will get an execution plan consistent with the bind variable value used

Result: a plan with a full scan can be chosen where instead an index is more appropriate for the general case (ex: cool task #4402)

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



Case: changing because of parameter change

A query execution plan can change because of init.ora parameters

Example: ALTER SYSTEM SET STAR_TRANSFORMATION_ENABLED=TRUE;

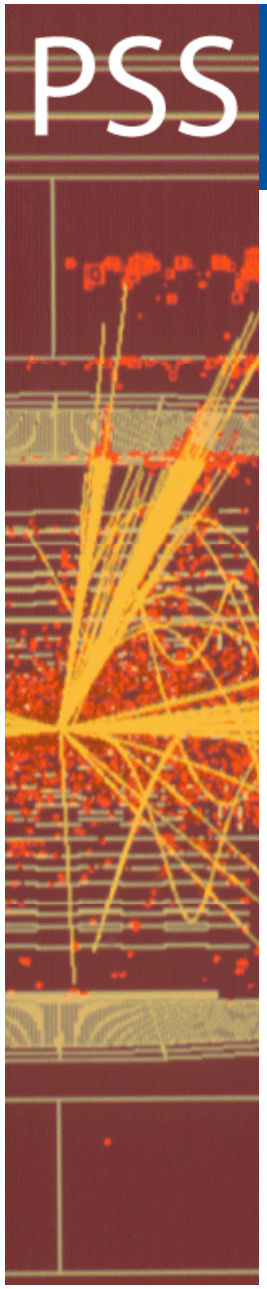
Bug:5397482 Wrong results from Star transformation with transitive join predicate fixed in 10.2.0.4 patchset.

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

Investigation of execution plans

- Main tools
  - Explain plan and dbms_xplan.display()
  - dbms_monitor.session_trace_enable (aka 10046 trace)
  - 10053 trace (cost based optimizer trace)

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



10046 trace

- The DBA is involved since it produces a trace file in ‘udump’
  - alter session set tracefile_identifier=‘…';
  - Exec dbms_monitor.session_trace_enable
  - ..run sql and pl/sql
  - Exec dbms_monitor.session_trace_enable
  - Identify the trace file in udump
  - Process the trace file with tkprof or orasrp

http://twiki.cern.ch/twiki/bin/view/PSSGroup/SQLTraceAnalysis

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

10046 trace file

- What is in the 10046 trace file:
  - All SQL that has been executed
  - Time spent in CPU and wait events
  - N# of DB calls
  - Allows for identify bottlenecks (response time analysis)

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

10053 trace

- Similar to the 10046 trace with a few differences
  - Enable with:

alter session set events '10053 trace name context forever, level 1' --level can be 1 or 2

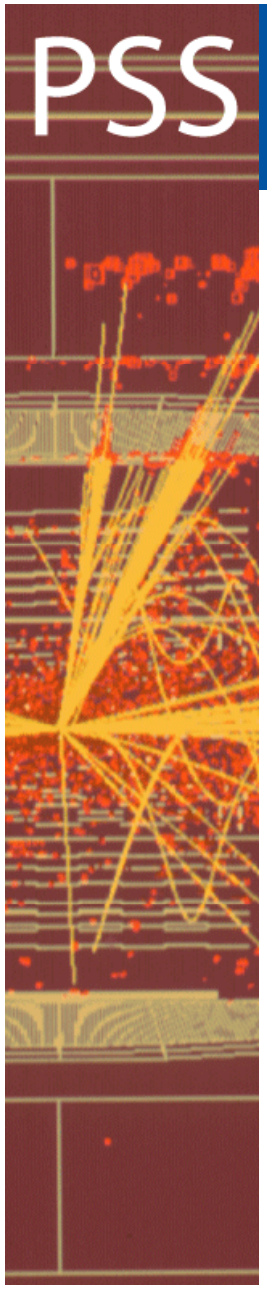
  - Parse the SQL statement
  - alter session set events '10053 trace name context off
  - Read the trace file

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

10053 trace file

- What is in the 10053 trace file:
  - Details of why the CBO has chosen a given execution plan as the best one
    - Parameters affecting the CBO
    - Access paths
    - Possible join orders
  - Compare 2 trace files line by line to understand why a query is fast in 1 environment and slowin another

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



10053 trace file example

CERN IT Department

- From 10053 on DB A (fast):

```
Best:: AccessPath: IndexRange  Index: I1_TEST1
Cost:4.00 Degree: 1 Resp: 4.00 Card: 2.20  Bytes: 0
```

- From 10053 on DB B (slow):

```
Best:: AccessPath: TableScan
Cost: 103.23 Degree:1 Resp:103.23 Card: 2.20 Bytes: 0
```

- Reason (from trace file 10053 DB B):

```
Index: I1_TEST1  Col#: 2
LVLS: 2  #LB: 306  #DK: 28035  LB/K: 1.00  DB/K: 1.00
   CLUF: 19713.00
UNUSABLE
```

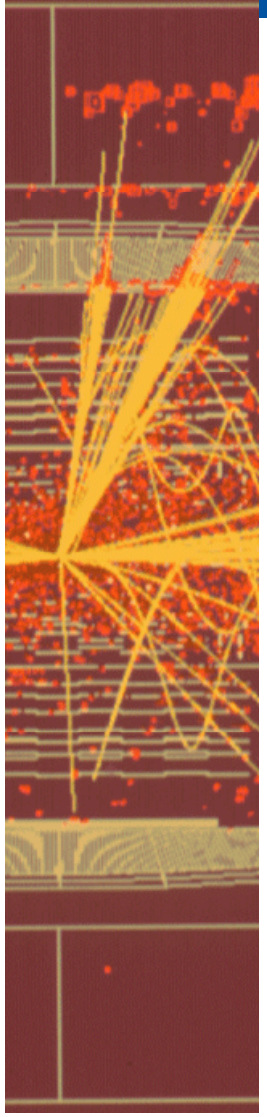
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

DB segment stats

- Best practices
  - Collect statistics regularly
  - Don’t fiddle with CBO parameters
    - Example: don’t set optimizer_index_caching, etc
- Open questions:
  - When to use the default statistics gathering and when to do more (like collecting histograms)?
  - Up-to-date statistics are often good but they can cause plans to change
  - Hints should be avoided, but several times they solve the problem

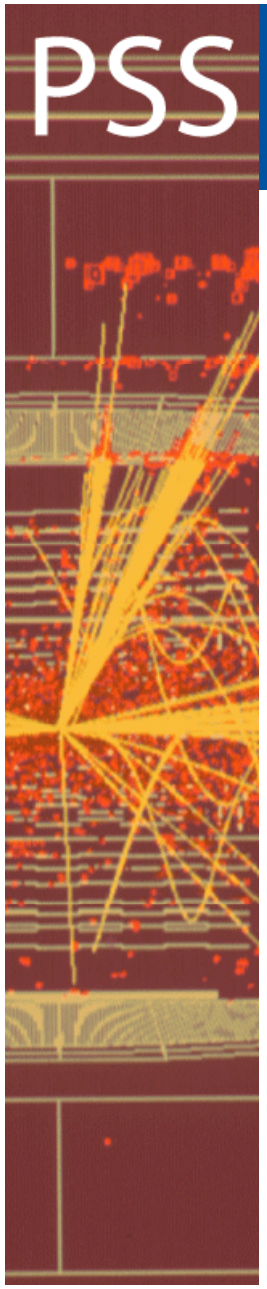
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Session profiles



- Session profiles used to limit resource usage
  - Number of concurrent sessions
  - Idle time
  - No CPU limit at the moment
- Limits cannot be too restrictive
  - Scenario: a few applications can consume the cluster resources if they suddenly change behaviour

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

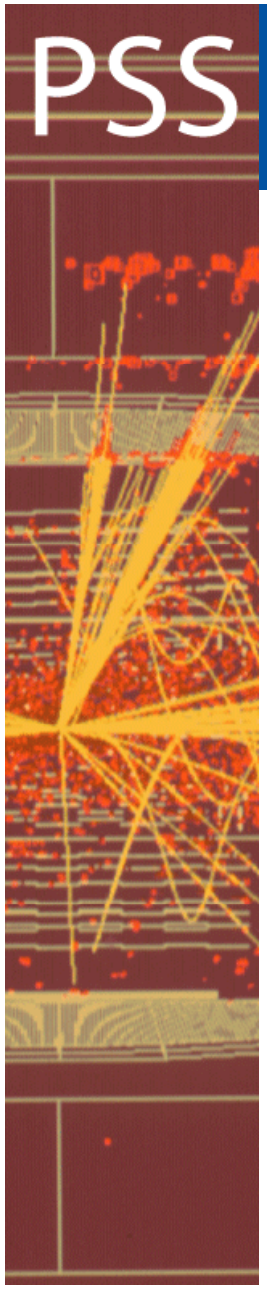


Case: single DB user absorbing DB resources

- A couple of application opening hundreds of sessions can reach max processes in Oracle
- An application changes ‘behavior’ and becomes CPU intensive.
  - A dozen of such sessions are enough to degrade the entire cluster performance

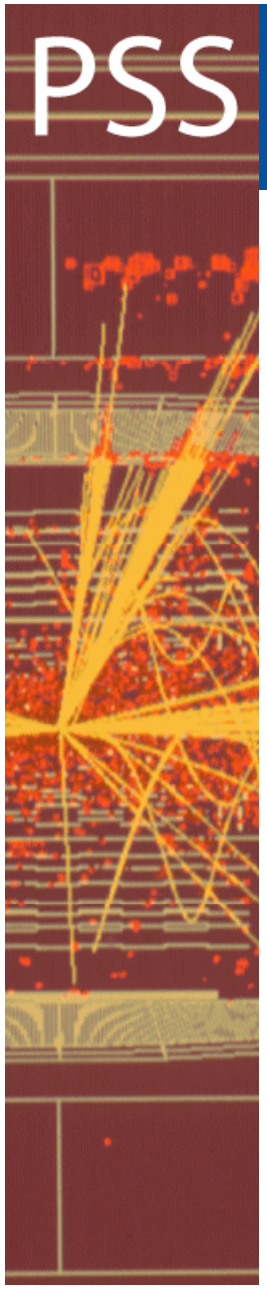
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Reaction in case of high load by 1 app



- Identify broken SQL
  - If possible, tune the SQL and deploy the change
- Relocate the app. Two strategies
  - to run on more nodes if this can fix the problem
  - to run on fewer nodes to isolate the issue
- In emergency, reduce the number of concurrent sessions for that app
  - apply a custom profile
- Last resource lock the user to preserve cluster functionality
- In all cases contact the application owner
  - Revert last app changes if applicable

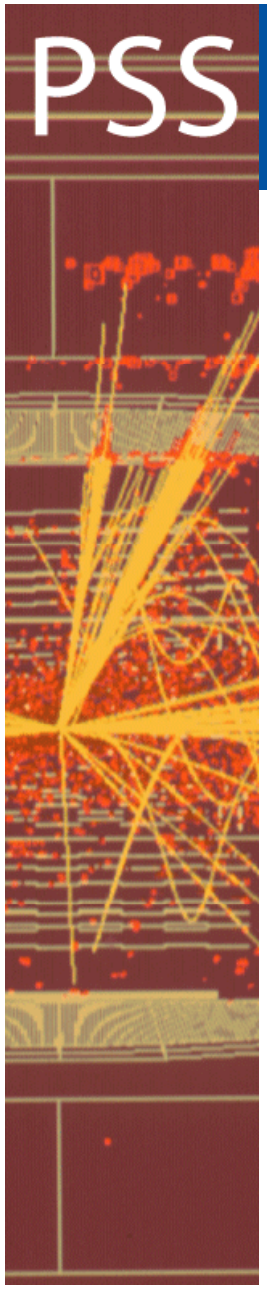
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Serialization due connections

- In Oracle a high connection rate is a killer for performance and scalability
  - Sessions consume systems resources when activated
  - Open sessions utilize memory
  - Diagnostic: can be difficult, these type of sessions don’t show up as slow SQL

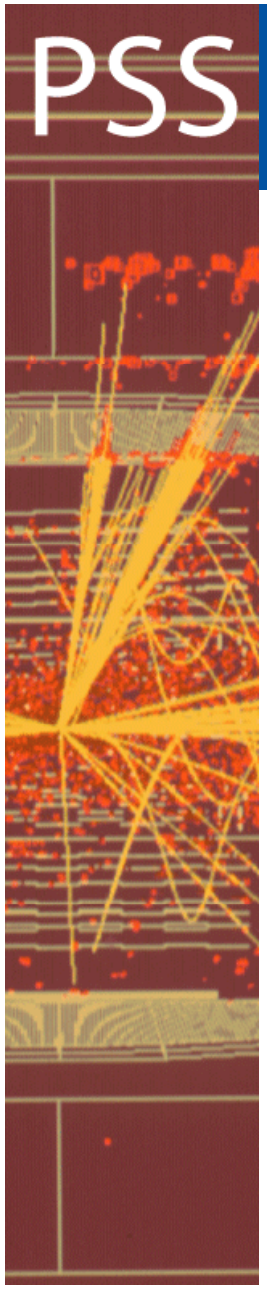
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



Serialization due connections

- Experience from production:
  - An application was running 100s of connections from lxbatch jobs
  - Solution for this case: the use of SQlite replicas
  - In general: use multi tier architectures and connection pooling
    - In development: CORAL proxy server

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



Space utilization

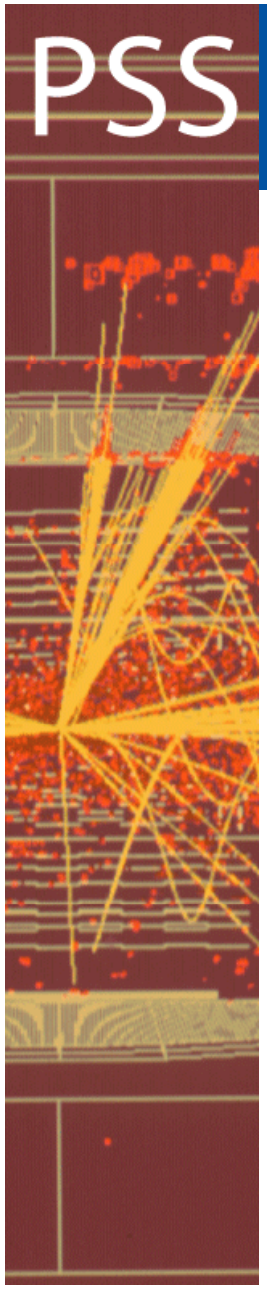
- Large tables can cause performance problems
  - Some times tables are artificially large
  - Check the space filling of blocks
  - Use dba_tables and look at AVG_ROW_LEN and NUM_ROWS columns
  - Moreover this is a wastage or resource
- Production experience
  - PVSS application writing only a few rows per block due to a bug

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

Sizing and Capacity Planning:

Notes on a recent exercise

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



AWR for capacity planning

- An important part of capacity planning is collecting data
- Use current production metric to extrapolate usage is a technique
- AWR contains system and DB metric collected every hour
  - In particular dba_hist_sysmetric_summary
  - Example: how to collect metrics for IO usage by all cluster instances

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

AWR for capacity planning

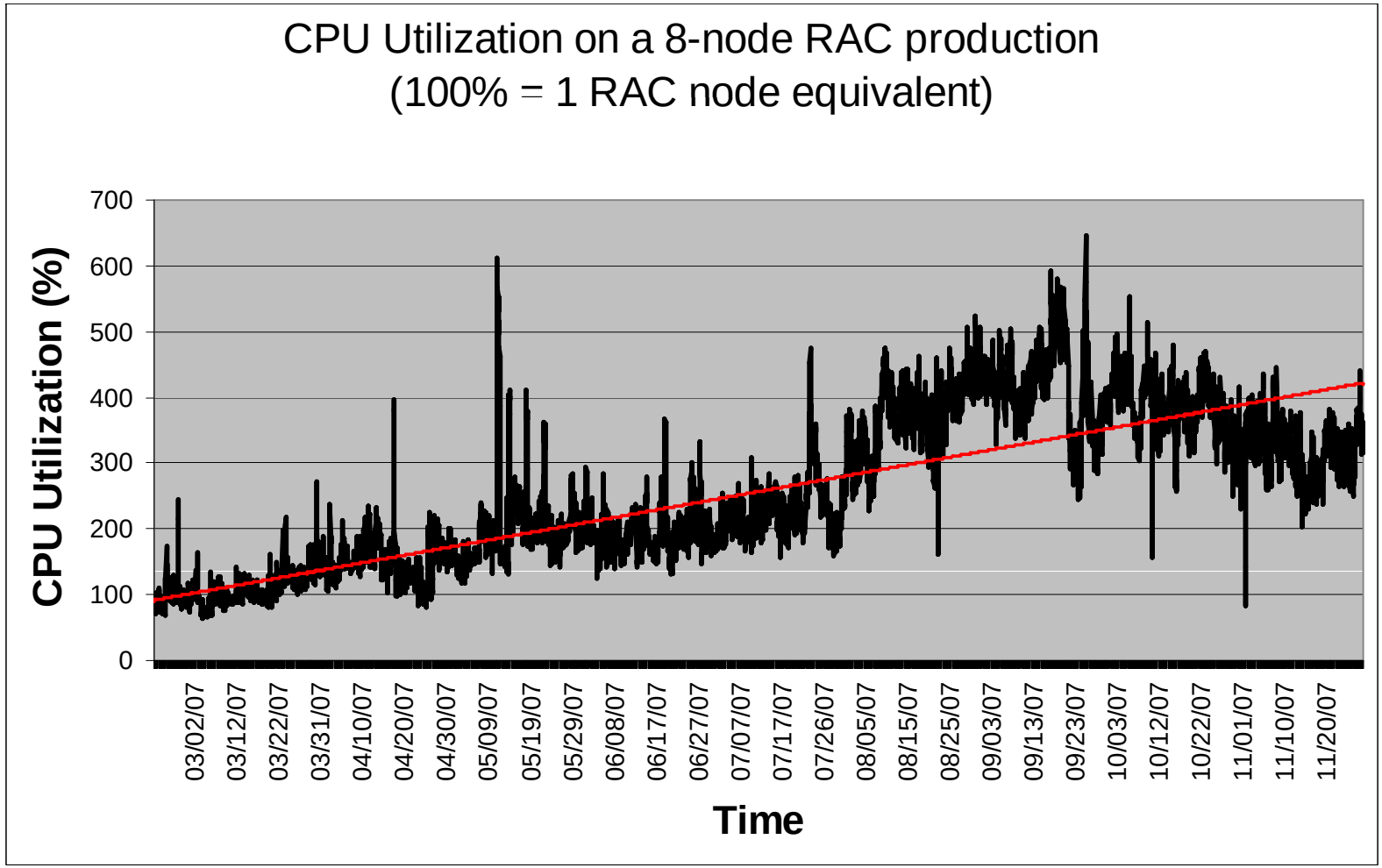
```
select min(begin_time), max(end_time),
   sum(case metric_name when 'Physical Read Total Bytes
   Per Sec' then average end) Physical_Read_Total_Bps,
   sum(case metric_name when 'Physical Write Total Bytes
   Per Sec' then average end) Physical_Write_Total_Bps,
   sum(case metric_name when 'Physical Read Total IO
   Requests Per Sec' then average end)
   Physical_Read_IOPS,
   sum(case metric_name when 'Physical Write Total IO
   Requests Per Sec' then average end)
   Physical_write_IOPS
   --ETC,
          snap_id
   from dba_hist_sysmetric_summary
   group by snap_id
   order by snap_id;
```

In addition set the retention period for AWR:

- Example:
  DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS (retention => 60*24*31)

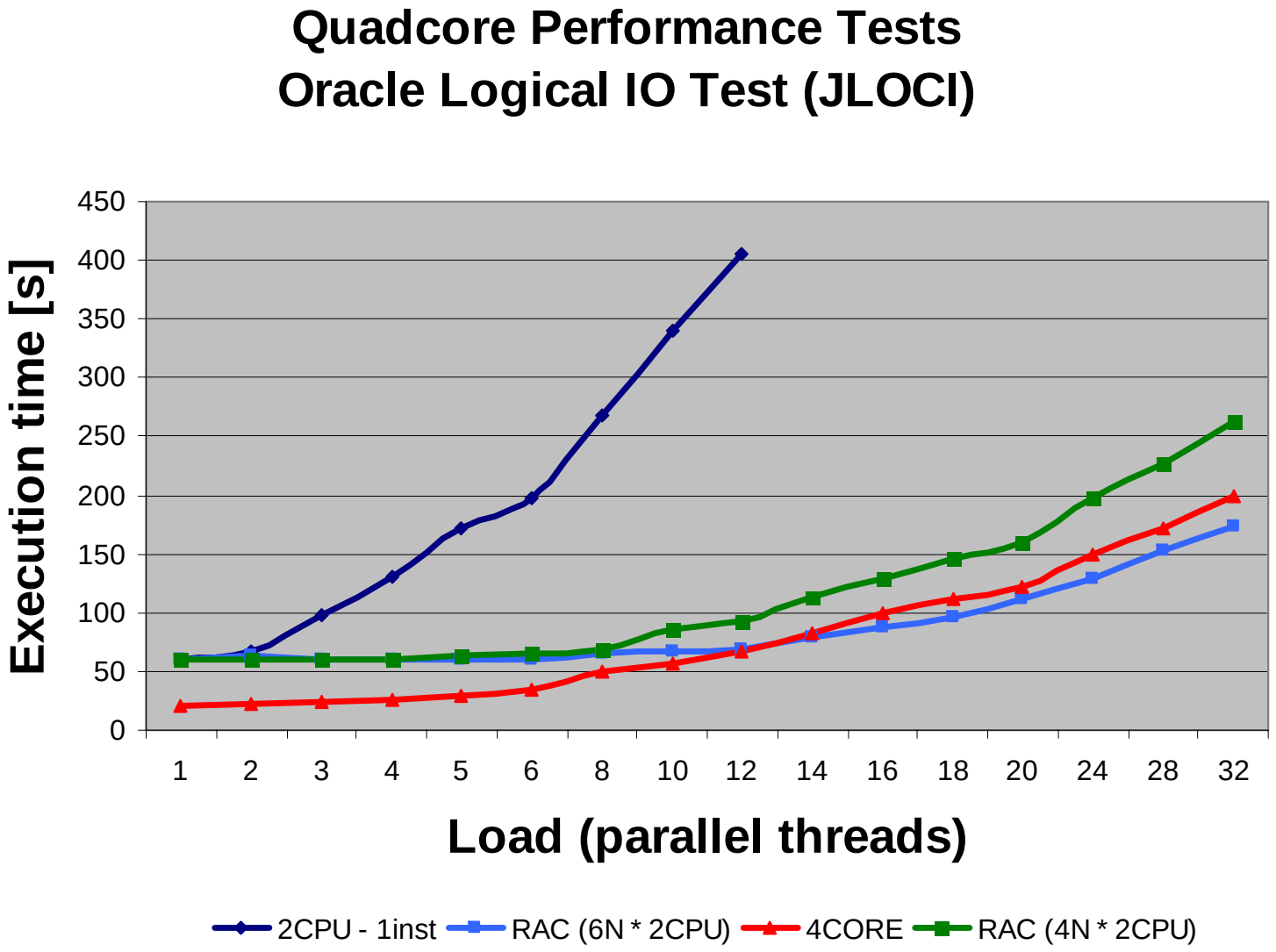
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

AWR for capacity planning, an example



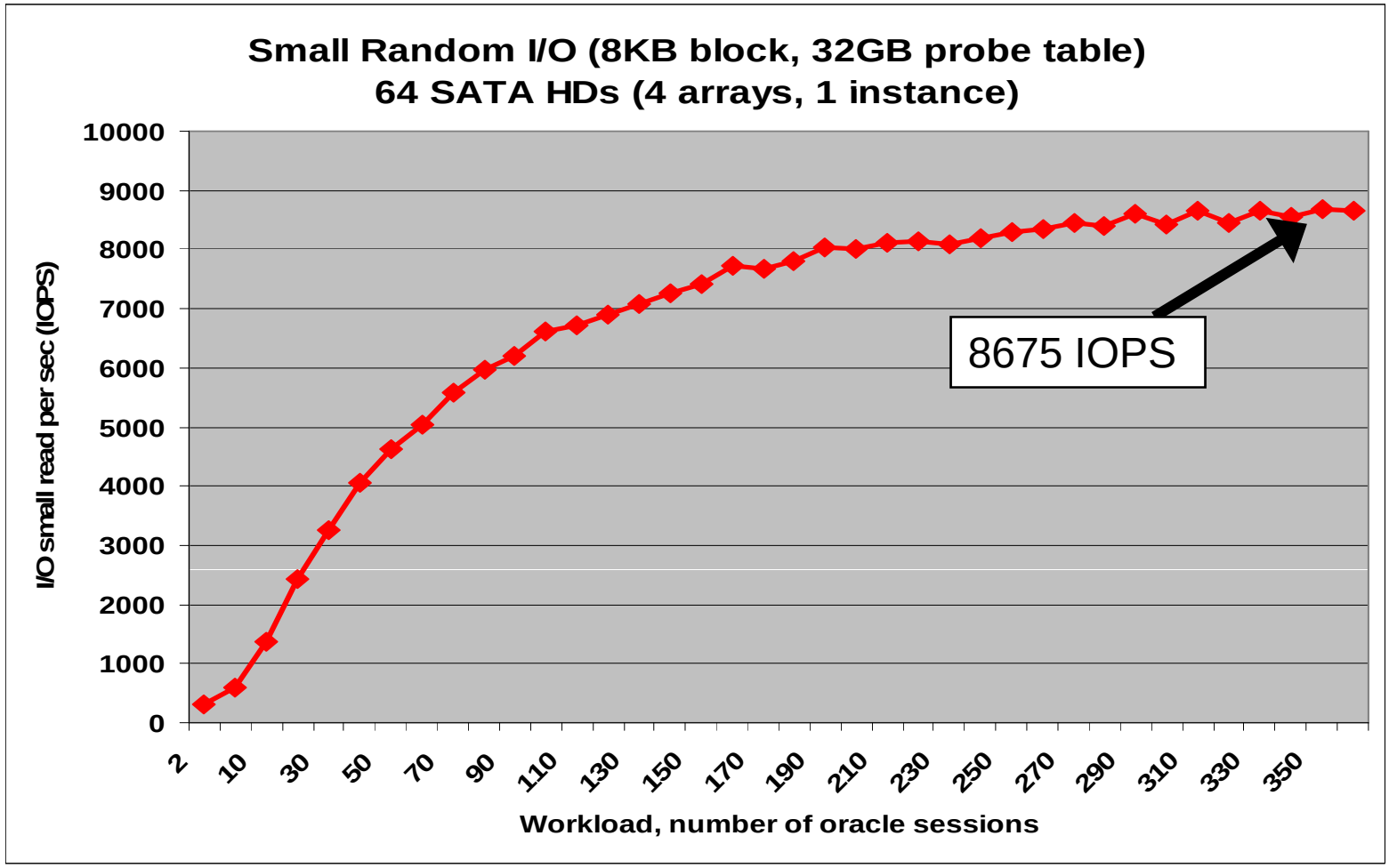
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Evaluating CPU performance

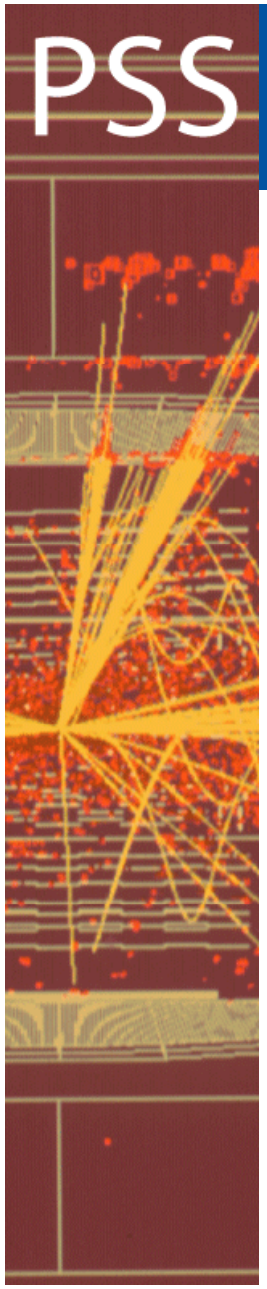


CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Evaluating I/O performance



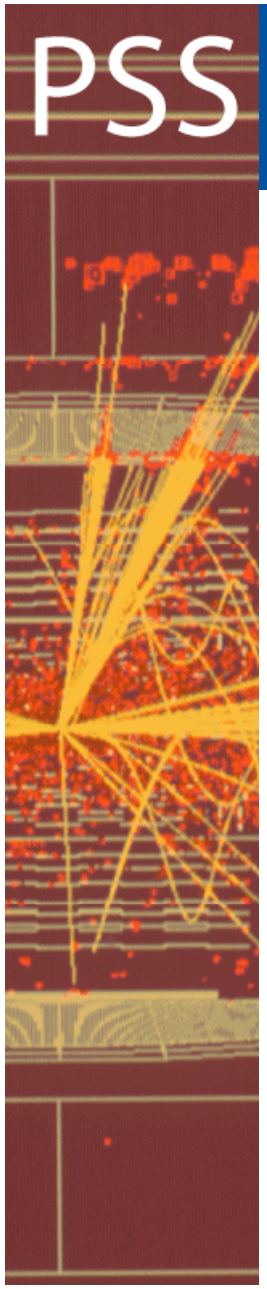
CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it



The sizing spreadsheet

- From current experience with Physics DBs the most critical (bottleneck) metrics are
  - CPU
  - IOPS for random I/O
- In addition disk capacity is critical
  - Consider mirroring requirements
  - Consider on-disk backup / physical standby space requirements
- Put this together with the expect workload
  - From tests and extrapolation from production
  - Involve the application owners

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



Conclusions

- A review of topics concerning Oracle performance for DB administrators
  - Using the experience of CERN's production DB Service for Physics
  - Oracle architecture
  - Monitoring and troubleshooting techniques
  - Sizing and capacity planning

- Links:
  - **http://www.cern.ch/phydb**
  - **http://twiki.cern.ch/twiki/bin/view/PSSGroup/PhysicsDatabasesSection**

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