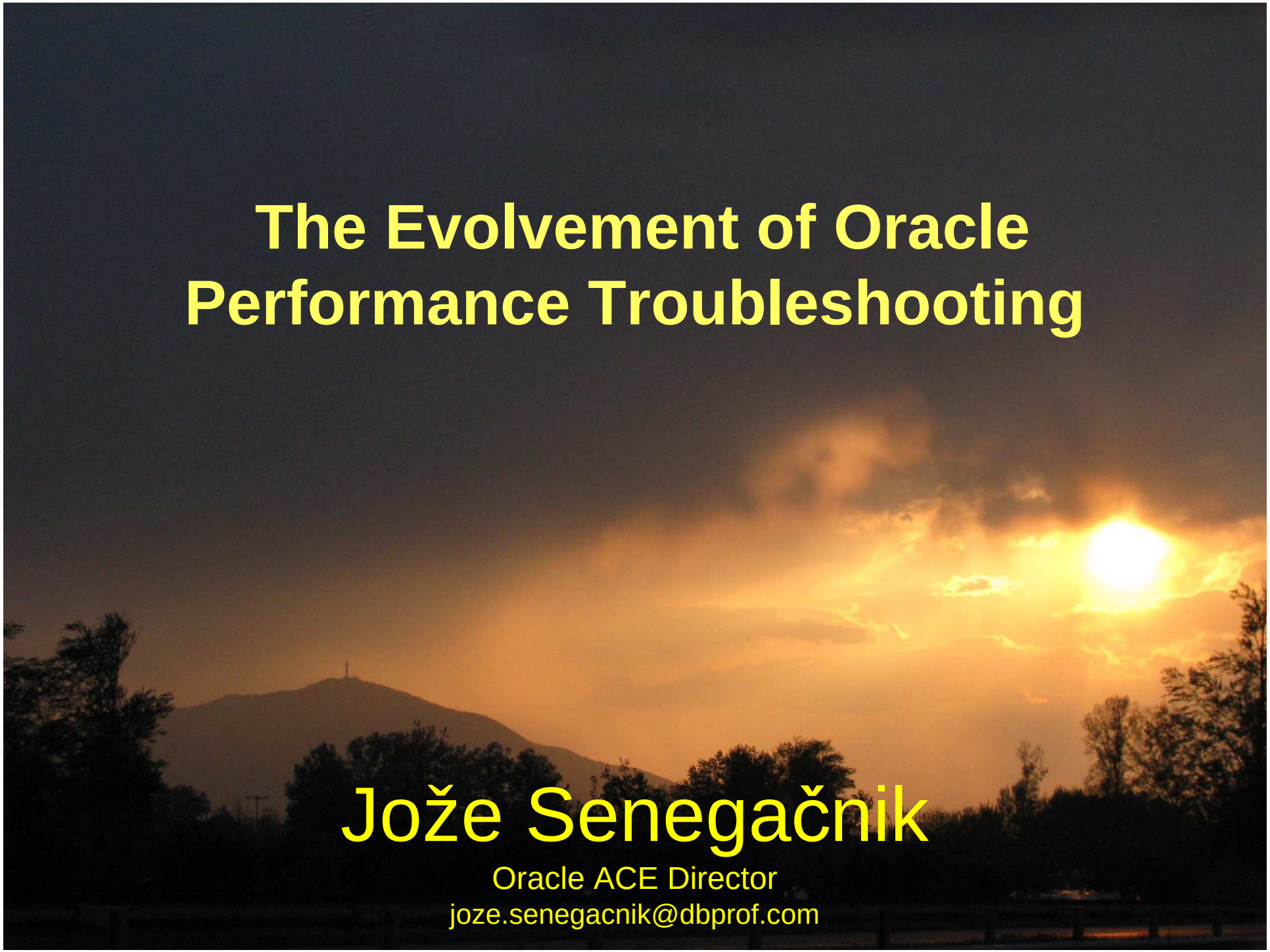
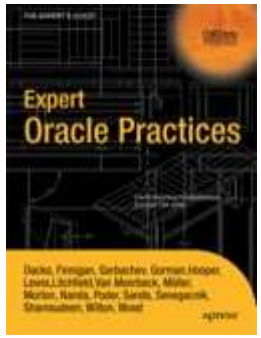

The Evolvement of Oracle Performance Troubleshooting
Jože Senegačnik
Oracle ACE Director
joze.senegacnik@dbprof.com

About the Speaker

Jože Senegačnik

- First experience with Oracle Version 4 in 1988
- 24 years of experience with Oracle RDBMS.
- Proud member of the OakTable Network www.oaktable.net
- Oracle ACE Director
- Co-author of the OakTable book “Expert Oracle Practices” by Apress (Jan 2010)
- VP of Slovenian OUG (SIOUG) board
- CISA – Certified IS auditor
- Blog about Oracle: http://joze-senegacnik.blogspot.com

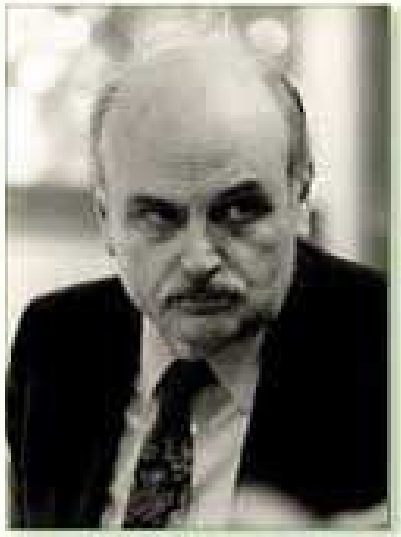


- PPL(A) – private pilot license PPL(A) / instrument rated IR/SE
- Blog about flying: http://jsenegacnik.blogspot.com
- Blog about Building Ovens, Baking and Cooking: http://senegacnik.blogspot.com

The Beginning

Beginning of The Relational Theory

- 1969
  - E.F. Codd publishes the internal version of his famous paper internally to IBM.
- June 1970
  - Edgar “Ted” F. Codd publicly publishes the paper: A Relational Model of Data for Large Shared Data Banks
    - Information should be stored in tables
    - IBM refuses to implement his model to preserve revenues of IMS/DB
    - Customers pressured IBM to build it (System-R project) and a relational language SEQUEL (Structured English Query Language - later SQL). Oracle used pre-launch conference papers to write their own SQL & launched it first.



Oracle Version 1

- There was no Version 1!

Oracle V2.3: The first Oracle

- Digital PDP-11 Macro-11
- Written in Assembler Language for PDP-11.
- The first commercial version of the database is sold to Wright-Patterson Air Force Base in 1979.
- It would be the first commercial version of any relational database sold.
- Features:
  - Two task architecture as 16 bit system allowed only 32K process size
  - Rule Based Optimizer
    - Table order
    - Equality or partially qualified keys for indexes
  - Tools
    - UFI (User Friendly Interface)
    - Rpt/rpf report writer
    - iag/iap for fast screen application generation
    - 3GL interfaces, Fortran, COBOL, Pascal

Oracle V2.3: The first Oracle

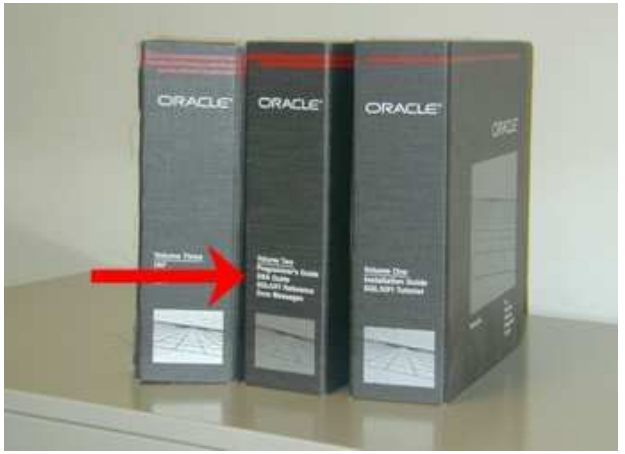
- Oracle V2.3 Tuning
  - Add debug code to identify problem statement
  - Cut and Paste into UFI
  - Change table order or modify indexing
    (Compulsive Index Tuning Disorder) 
  - Modify the design 

Oracle V3: Portability

- Written in VAX C
  - Complete rewrite
  - Used VAX/VMS shared memory capabilities for secure single task architecture
  - Supported transactions
- Version 3 is the FIRST 32-bit RDBMS.

Oracle V4: Portability

- V4
  - Cube
    - MS-DOS in 640K
  - Read consistency
  - BI files, copies of ‘before image’ of block
  - views, subqueries
  - iag/iap became Fastforms
  - Tuning very similar to V2.3



Oracle V5: Distributed

- SQL*Net
  - Client/server
  - Distributed (v5.1)
- SQL*Plus replaced UFI
- SQL*Forms replaced Fastforms
- Stable!
  - And needed to be as Ingres, Sybase, Informix were all competing

Oracle V5: Distributed

- Tuning
  - Database monitor for locks, file IO
  - Trace
    - Details of what SQL was running and access paths
  - Use ODS to identify problem session and then trace to identify problem SQL then tune as in Oracle V2.3

Oracle V6: Scalability

- New architecture
  - DBWR for database file writes
  - LGWR for log writes
  - Rollback segments for undo and CR
  - Row level locking
    - (option – default was block locking)
  - First hot backup
  - Only PL/SQL anonymous blocks
  - Tablespaces
  - OPS
- V6 took a long time to stabilize
  (6.0.28. was my first production version)

"There were user conferences where I thought I needed to wear a bulletproof vest. People were really upset with us."

Randy Baker, Head of Oracle Support

Oracle V6: Scalability

- New ‘Dynamic Performance Tables’ v$
  - v$sysstat
  - v$lock
  - v$waitstat
  - timed_statistics parameter, off by default
- SQL*DBA Monitor replaced ODS
- SQLtrace and tkprof
- Explain plan

Oracle V6: Scalability

- Bstat/estat used for tuning benchmarks
  - Created (and dropped) copies of V$ tables
- Performance Tuning Guide
  - Introduced a ‘Tuning Method’
    - Tune Application and SQL
    - Memory
    - IO
    - Contention
  - Ratios
    - Cache hit ratio !!!!!
    - Golden era for “Expert” ratio tuning

Oracle V6: Scalability

- Stats package first written
  - Capture ‘snapshots’ of V$tables in tables in database
  - Report across any two snapshots
  - Report organized so that most important information at top.
  - Details there for drilldowns
  - Distributed internally within Oracle and to some customers.

Oracle7: Programmable

- Procedures, Packages, Triggers (stored PL/SQL)
- Constraints
- Cost Based Optimizer (hardly usable)
- Shared pool
  - addresses parsing/memory issues
- Parallel Query (7.1)

Oracle7: Programmable

- V$SQL to see what is in shared pool
  - which SQL was doing most buffer gets, disk reads
- Wait Events introduced (event 10046)
  - The story behind
  - Tkprof processes wait events in 7.1 and stops processing wait events in 7.3 
    (somebody commented out part of the code)
- Enterprise Manager

Breakthrough in Tuning - YAPP Method

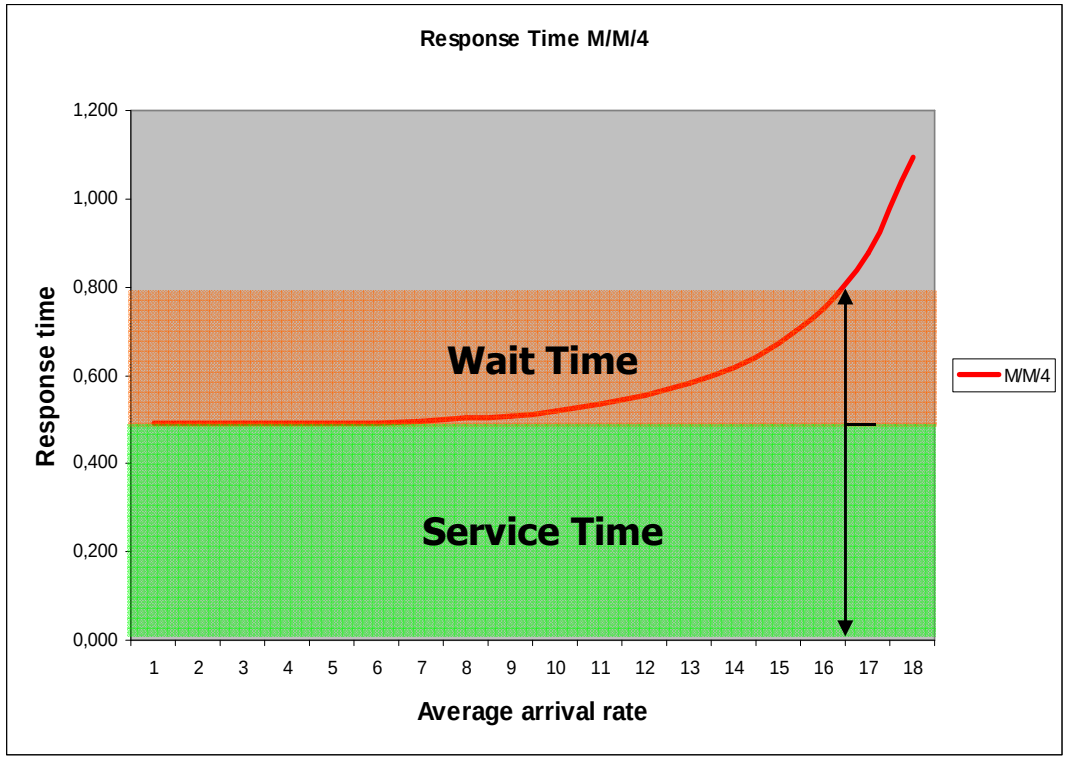
- Tuning with ratios – limited success
- Anjo Kolk, Shari Yamaguchi, Jim Viscusi in 1999 publish **Yet Another Performance Profiling Method** (Or YAPP-Method)
  - Available on MOS [DocID:148518.1] – Oracle Technical Paper
- The motto is: "*Common sense before hard work*" (analyze the response time and then remove the bottlenecks).
- Start of religious war
  - whether to tune with wait events or ratios
- YAPP method becomes the base for AWR in 10gR1.

ORACLE
Support Services

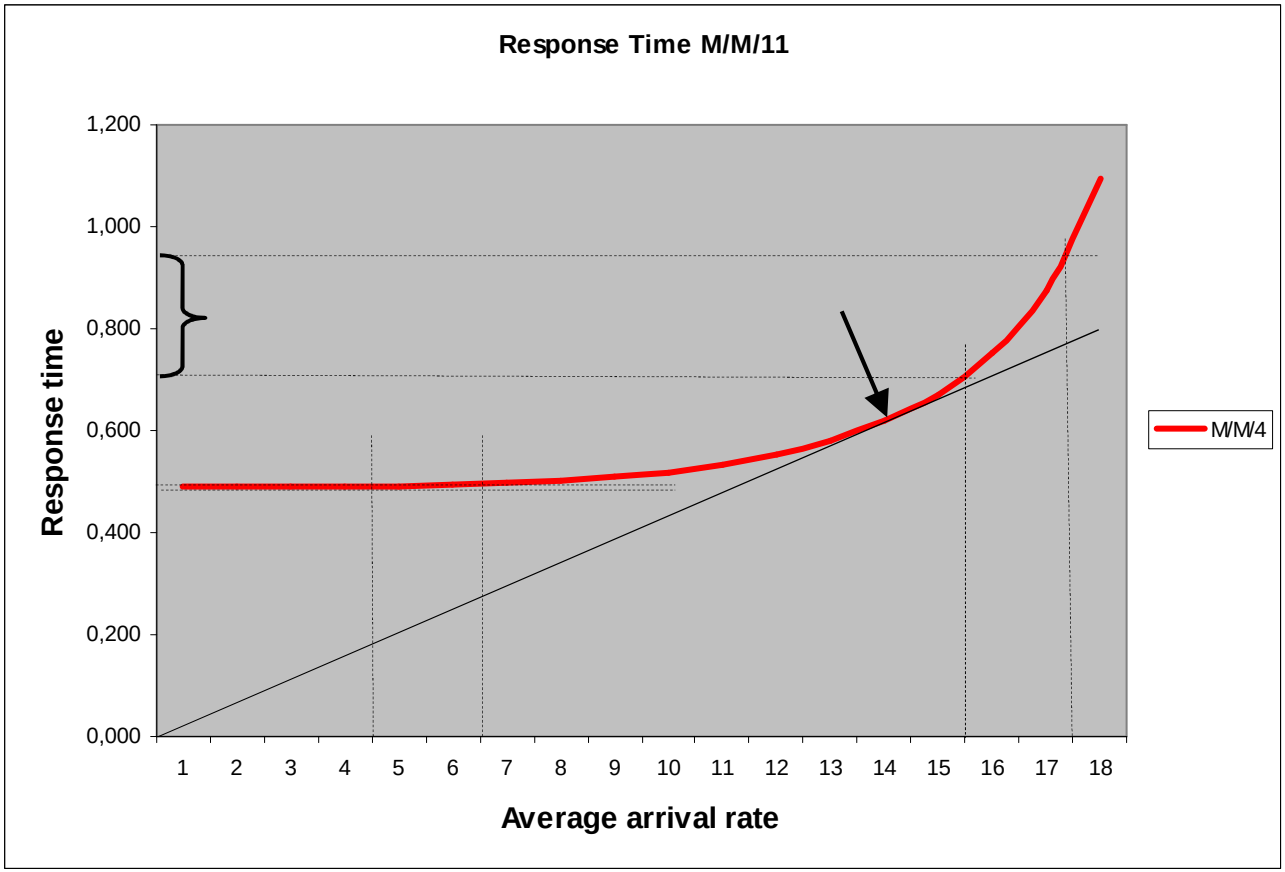
Anjo Kolk

Definition of the Response Time

Response Time = Service Time + Wait Time



Response Time Fluctuation

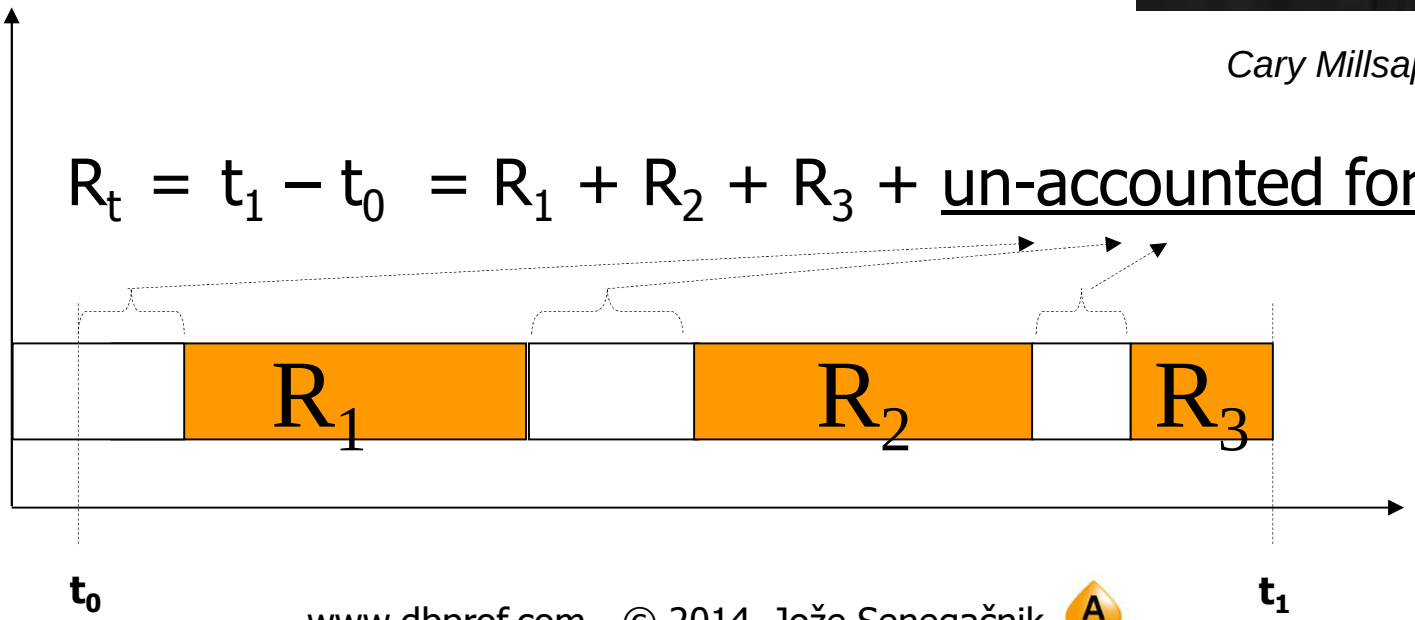


Un-Accounted For Time

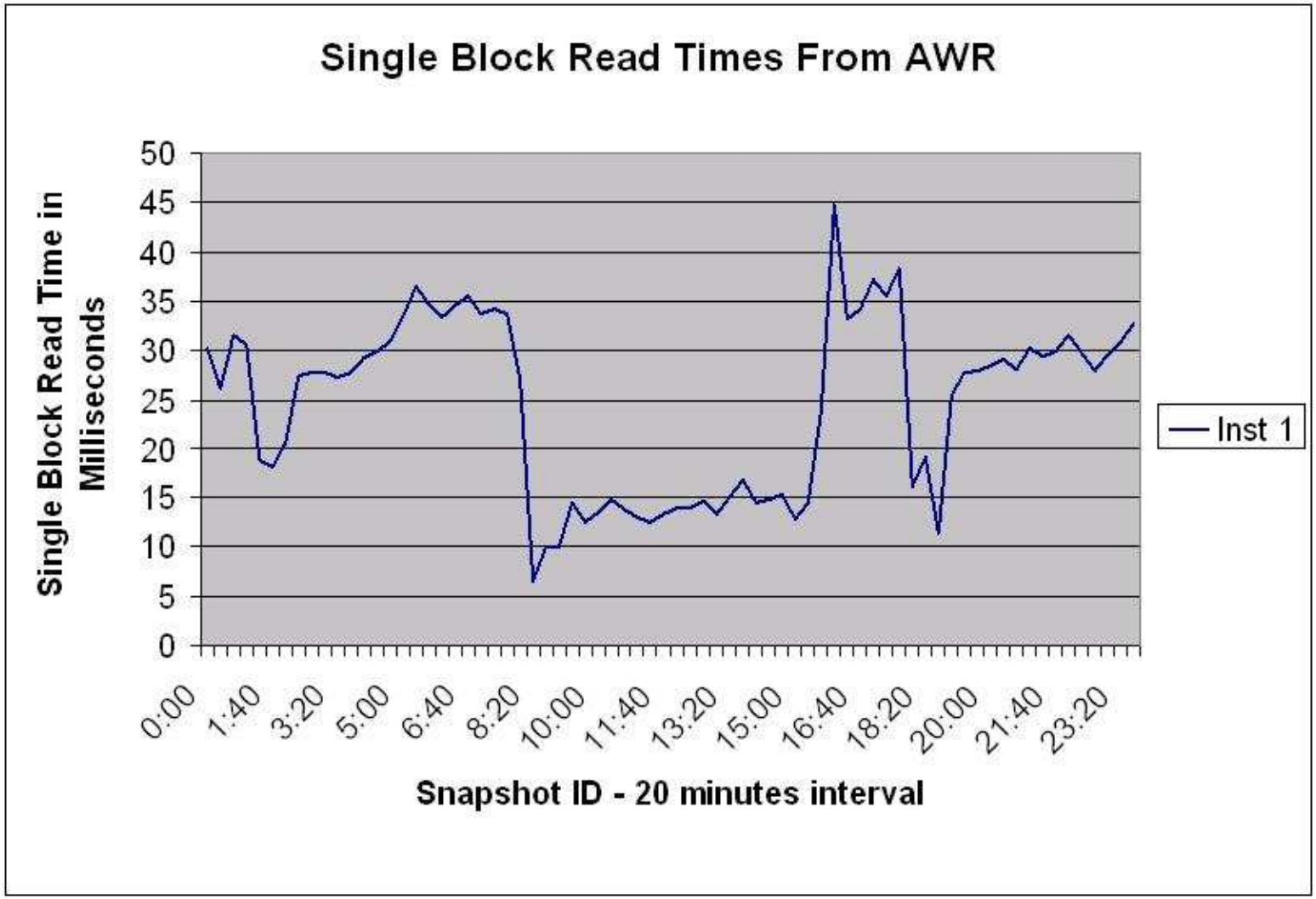
- Carry Millsap from Hotsos creates their trace analyzer and introduces "Un-accounted for time"
- Oracle reports response times R_1, R_2, R_3
- Missing time spent in OS queue or because of swapping/paging
- Could be a measurement error as well.

Cary Millsap

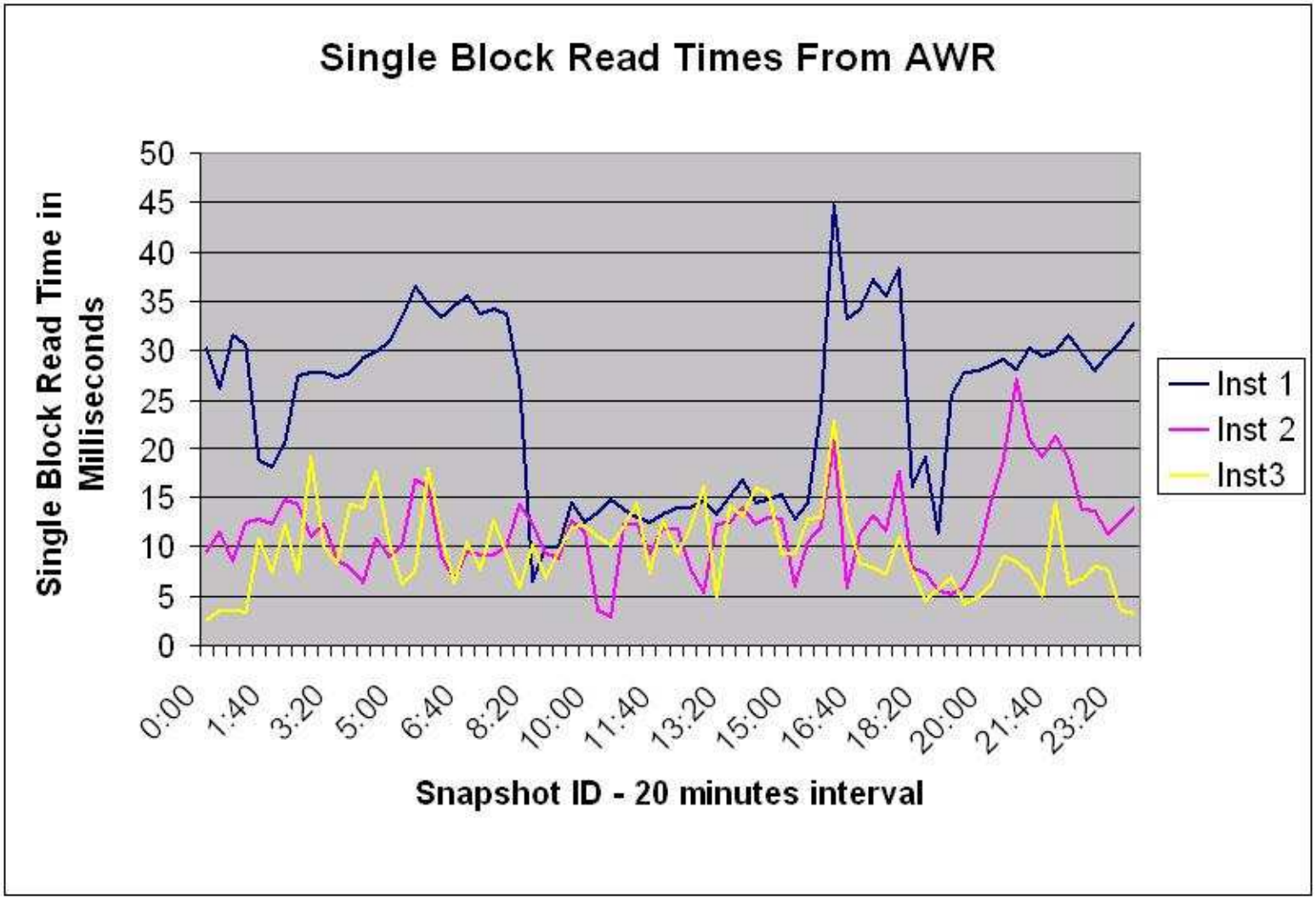
$$R_t = t_1 - t_0 = R_1 + R_2 + R_3 + \underline{\text{un-accounted for time}}$$



Measurement Error – db file sequential read



Measurement Error – db file sequential read



Oracle 8 & 8i: Objects & Java

- Partitioning
- First ‘cache fusion’ features
- 8i STATSPACK as part of product
  - High level summary
    - Wait times prominent
  - Drilldowns
  - Time based method of analysis

Oracle 9i: Unbreakable

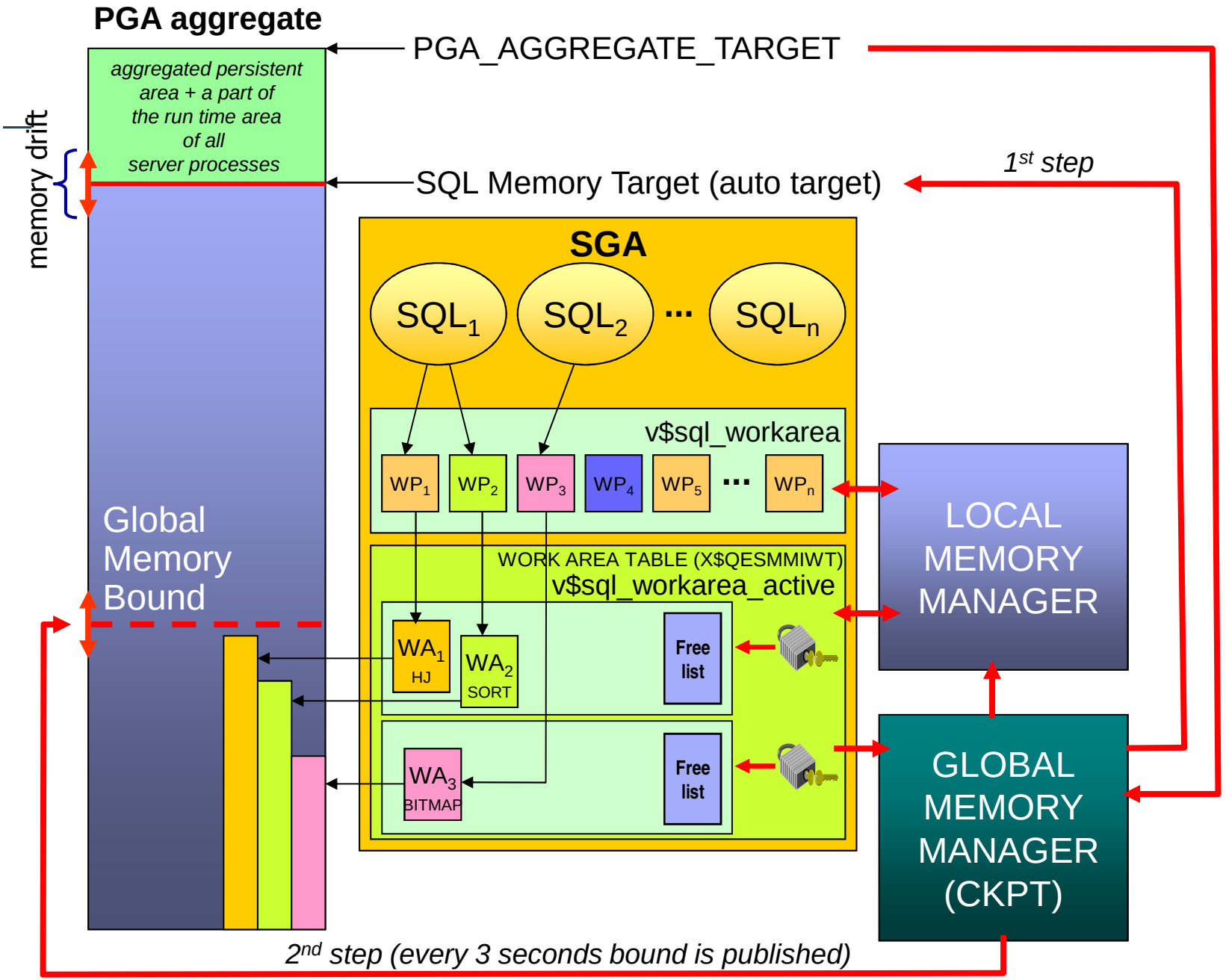
- RAC instead of OPS
- Statistics changes (Timing is everything)
  - Times in microsecond (CPUs are getting faster)
  - V$SQL_PLAN
  - Times in V$SQL
    - no longer need proxies of buffer gets, disk reads
  - Segment statistics
  - Object ids in waiting sessions
  - Row source statistics (in runtime execution plan)

Oracle 9i: Unbreakable

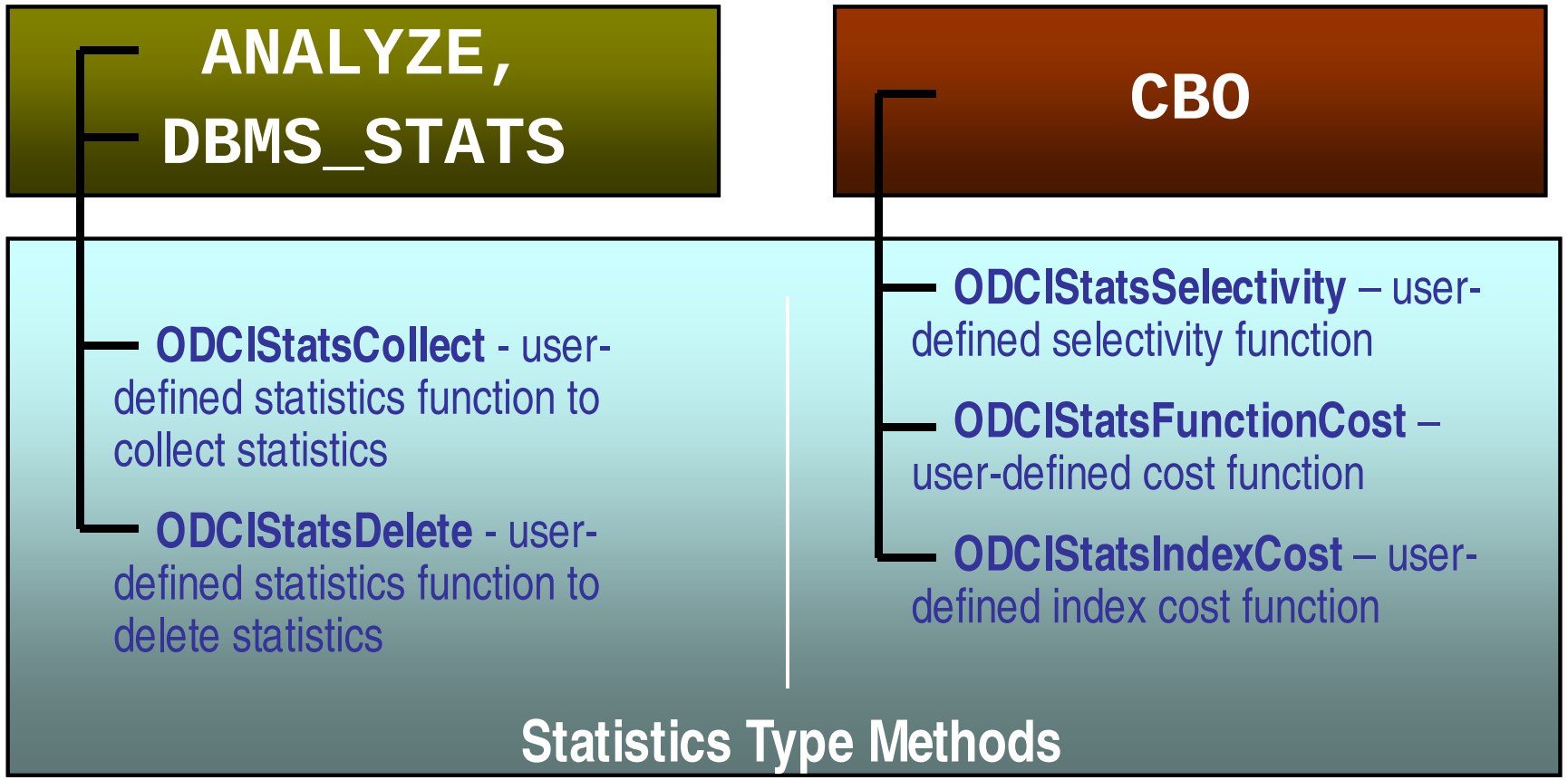
- Statistics changes
  - Statistics_level parameter
    - Timed_statistics defaults to true
  - Advisories
    - buffer cache,
    - shared pool
    - PGA
- Tkprof processes wait events once again.
- In the mean time SQL trace file analyzers were written – Cary Millsap (Hotsos)

Oracle 9i: Unbreakable

- Statspack changes
  - CPU time into ‘Top Wait Events’
  - SQL rep
    - Report usage of SQL between two snaps including all plans that had been used
- More timing data => better diagnosis
- Automatic PGA memory management – finally PGA is allocated from OS and returned to OS when not needed any more.


PGA aggregate
PGA_AGGREGATE_TARGET
aggregated persistent area + a part of the run time area of all server processes
memory drift
SQL Memory Target (auto target)
1st step
SGA
SQL1
SQL2
...
SQLn
v$sql_workarea
WP1
WP2
WP3
WP4
WP5
...
WPn
WORK AREA TABLE (X$QESMMIWT)
v$sql_workarea_active
WA1
HJ
WA2
SORT
WA3
BITMAP
Free list
Free list
Global Memory Bound
LOCAL MEMORY MANAGER
GLOBAL MEMORY MANAGER (CKPT)
2nd step (every 3 seconds bound is published)

User-Defined Statistics Type (For Objects and PL/SQL)



Latest Versions
10g and 11g

Oracle 10gR1: Managability

- Autotuning of SGA
  - buffer cache
  - shared pool
  - large pool
  - java pool
- AWR
  - Statspack++
  - Installs as part of database
  - Built into database so more efficient

Oracle 10gR1: Managability

- ADDM
  - Time as common currency
  - Time based tuning recommendations
- Tuning Advisor
  - Makes tuning recommendations for tuning an individual statement
  - Useful for tuning simple statements – prepares SQL Profiles, suggestions...
  - Requires tuning pack

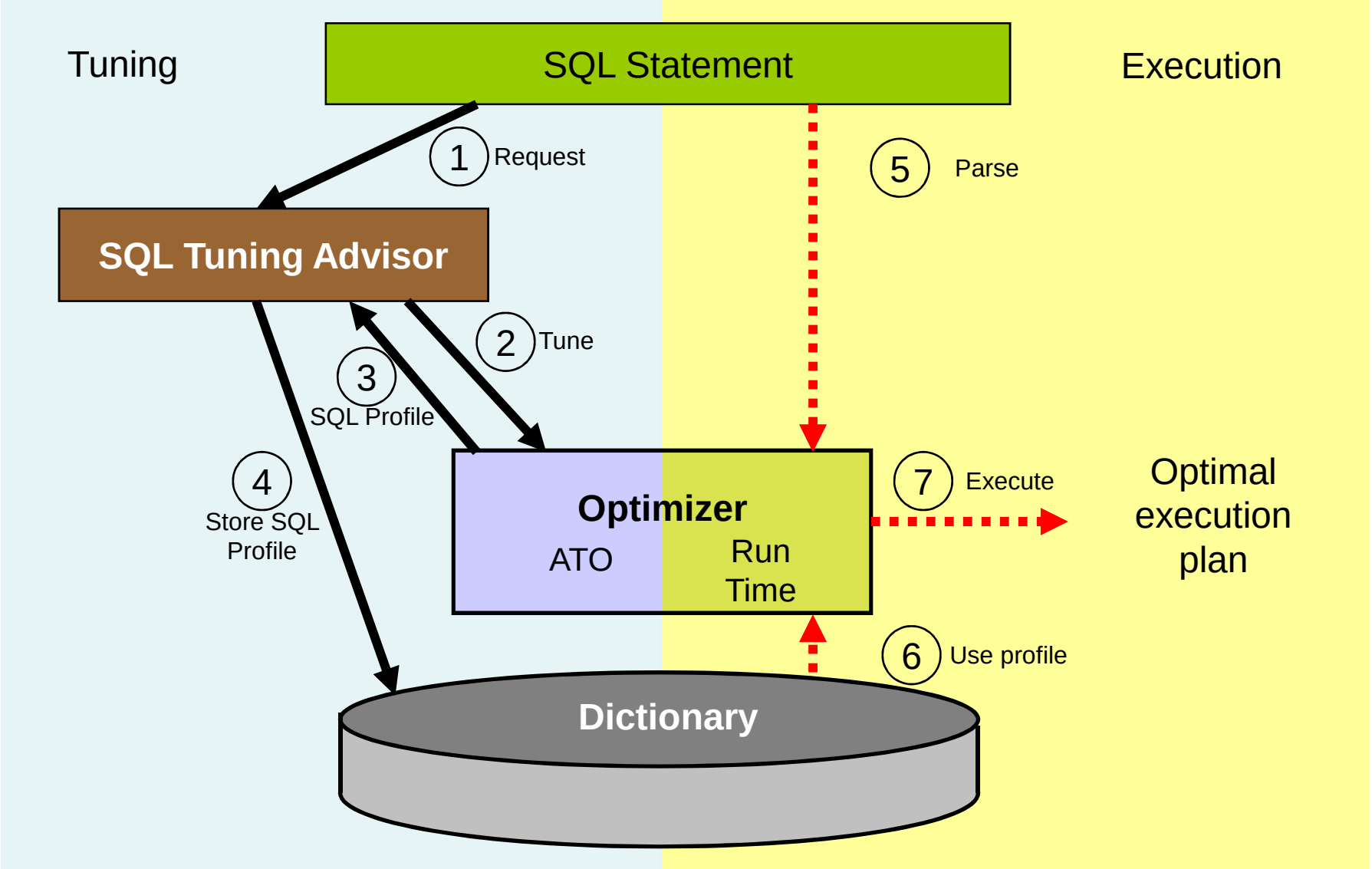
Oracle 10gR1: Managability

- New Data sources
  - Time model
    - DB time!
    - Allows breakdown of time spent in database.
    - Includes Java time, PLSQL time, connection time, parse time breakdown
  - ASH
    - Sampled activity data with many dimensions
    - Sampled DB time
    - Low impact on performance

Oracle 10gR2: Managability

- What's new in Oracle 10gR2?
  - Periodic updates of statistics in V$SQL
  - Blockers identified in ASH
  - ASH report
  - AWR diff diff report (compare two periods)
  - SQL Report
  - More verbose trace files
    - P1,p2,p3 enumerated
    - Object id
    - Timestamps

Automatic Tuning Process



OPT_ESTIMATE Hint ised in SQL Profiles

- Adjusting the number of rows returned from a table
  - e.g. 10 times as many rows as expected are returned from table T

```
OPT_ESTIMATE(@SEL$1, TABLE, T@SEL$1, SCALE_ROWS=10)
```

- Adjusting the number of rows returned through an index scan
  - E.g. 10 times fewer rows as expected are returned from table CUSTOMER through index CUSTOMERS_PK

```
OPT_ESTIMATE(@SEL$1, INDEX_SCAN, CUSTOMER@SEL$1, CUSTOMERS_PK, SCALE_ROWS=.1)
```

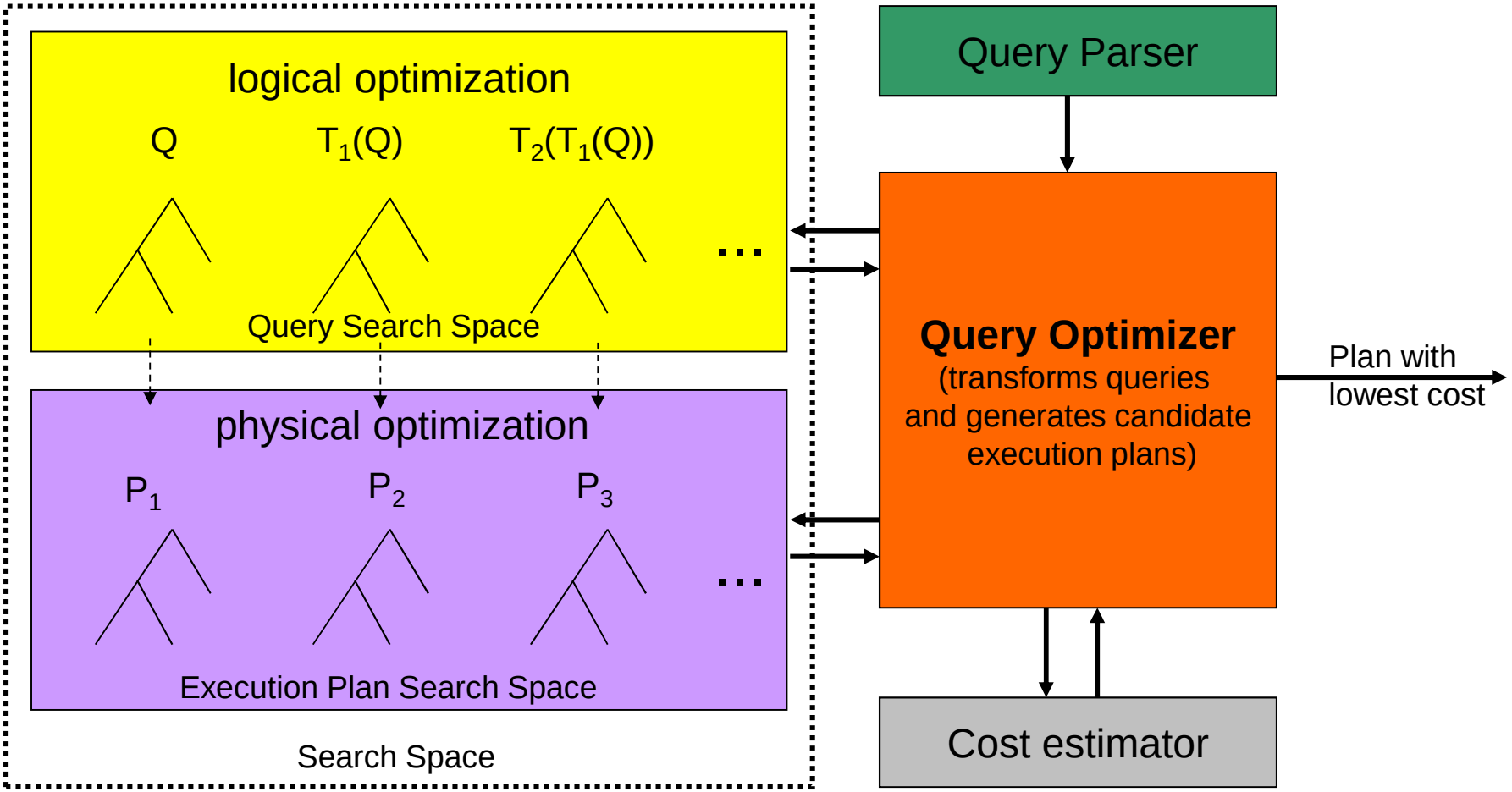
- Adjusting the number of rows returned from a join
  - E.g. 3.6 times as many rows as expected are returned when T1 and T2 are joined

```
OPT_ESTIMATE(@SEL$1, JOIN, (T1@SEL$1, T2@SEL$1), SCALE_ROWS=3.6)
```

Cost Based Query Optimization in 10g/11g

- Before 10g only heuristic transformations!
- Query optimization is performed in two phases
  1. **Logical optimization** (query transformation)
     - All known sources of information are used by the CBO to generate additional clauses during the query transformation phase.
  2. **Physical optimization** – finds information about:
     - Possible access method to every table (full scan, index lookup,...)
     - Possible join method for every join (HJ, SM, NL)
     - Join order for the query tables (join(join(A,B), C) or (join(join(C,B), A) ...

Query Optimization




ORACLE Enterprise Manager
Grid Control 11g
Home
Targets
Deployments
Alerts
Compliance
Jobs
Hosts | Databases | Middleware | Web Applications | Services | Systems | Groups | Virtual Servers | All Targets
Database Instance:
Top Activity
Drag the shaded box to change the time period for the detail section below.
View Data
CPU Cores
Active Sessions
Jun 21, 10:00PM
Jun 22, 01:00AM
Jun 22, 04:00AM
Jun 22, 07:00AM
Jun 22, 10:00AM
Jun 22, 01:00PM
Jun 22, 04:00PM
Jun 22, 07:00PM
Jun 22, 10:00PM
Other
Queueing
Network
Administrative
Configuration
Commit
Application
Concurrency
System I/O
User I/O
Scheduler
CPU + CPU Wait
Detail for Selected 30 Minute Interval
Start Time Jun 22, 2012 10:41:37 AM
Top SQL
Actions Schedule SQL Tuning Advisor Go
Select All | Select None
Select
Activity (%)
SQL Hash Value
SQL Type
4.84 1qy6p9q85q41j INSERT
4.19 2992p7nm8stu2 SELECT
3.55 c67draknz9hx8 SELECT
3.55 guqu0vguqqdna SELECT
3.23 6qmtnamdwqu36 SELECT
2.9 axm5005kdufpd SELECT
2.58 qc6k70xc7kfwj SELECT
2.58 2bpp4r8ajsuz3 SELECT
2.58 qmqvnhj71y1au SELECT
1.94 ccwwunp6k3tcn SELECT
Actions Schedule SQL Tuning Advisor Go
Total Sample Count: 310
Top Sessions
View Top Sessions
Activity (%)
Session ID
User Name
Program
7.56 4943 aam.exe
5.52 4526 aam.exe
5.52 407 aam.exe
5.52 3838 aam.exe
5.23 407 aam.exe
2.03 3413 IBI.exe
1.45 5285 oracle@ (J000)
1.45 3058 aam.exe
1.45 5719 frmweb.exe
1.16 762 TSERV50.exe
Total Sample Count: 344

Automatic Memory Management **Enabled** Disable

Total Memory Size 10240 MB Advice

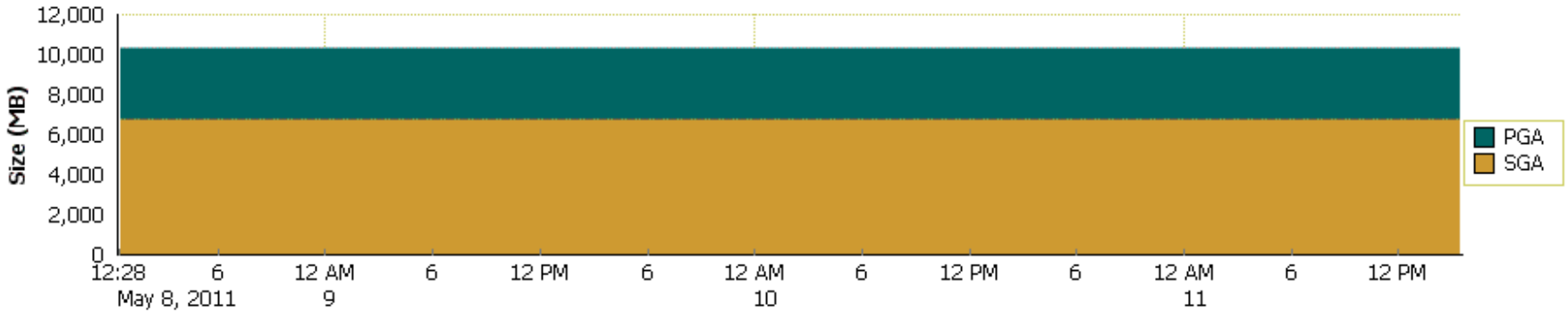
Maximum Memory Size 10240 MB

The database must be restarted before any changes to this value take effect.

> Every automatics brings automatic instability in your system!!!
>
> *Tanel Poder*

Allocation History

This chart shows the history of the components of the Memory.

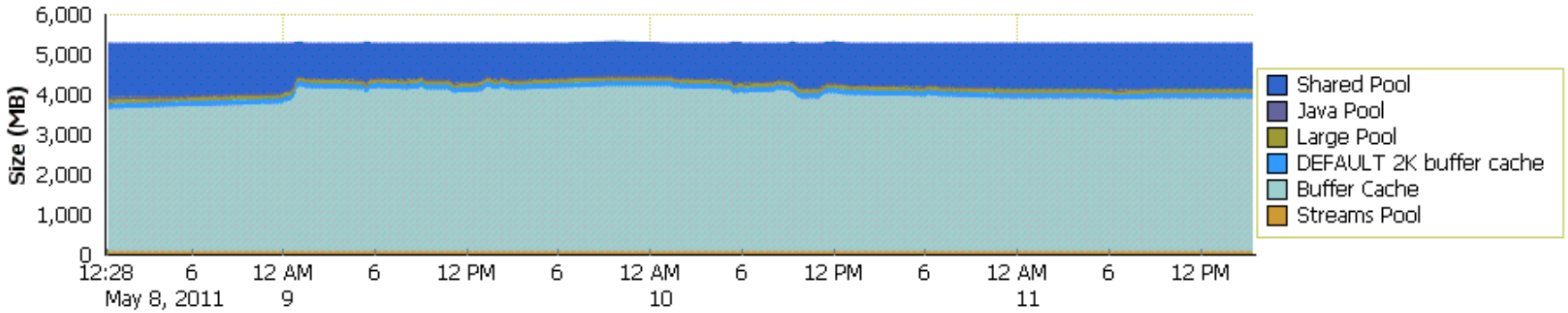


SGA PGA

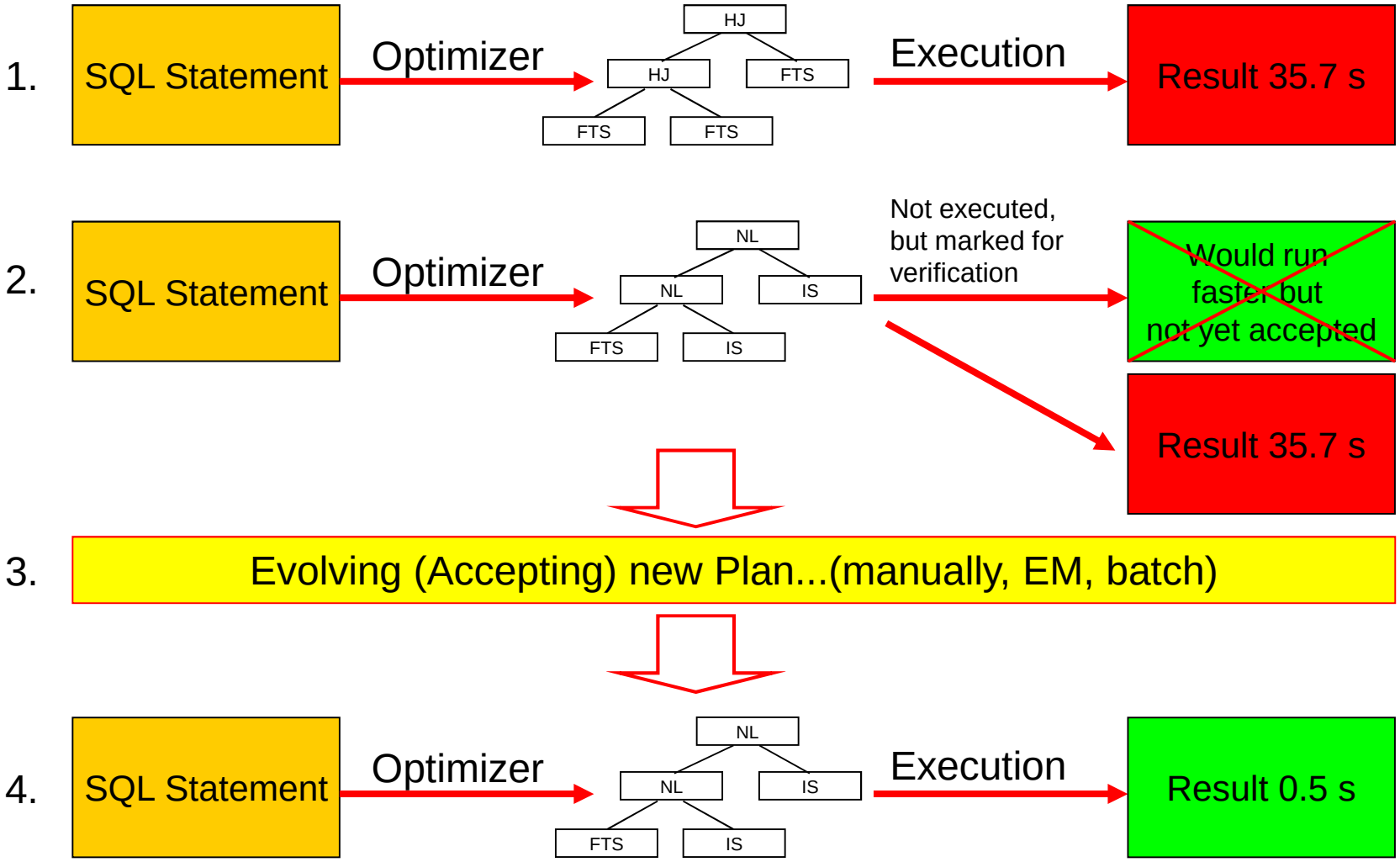
The System Global Area (SGA) is a group of shared memory structures that contains data and control information for one Oracle database. The SGA is allocated in memory when an Oracle database instance is started.

Allocation History

This chart shows the history of the components of the SGA.



Scenario With SQL Plan Management



Adaptive Cursor Sharing (ACS)

- Adaptive cursor sharing (ACS) allows the optimizer to generate a set of plans that are optimal for different sets of bind values and the cursor is still shared.

SQL Plan Monitor (1)

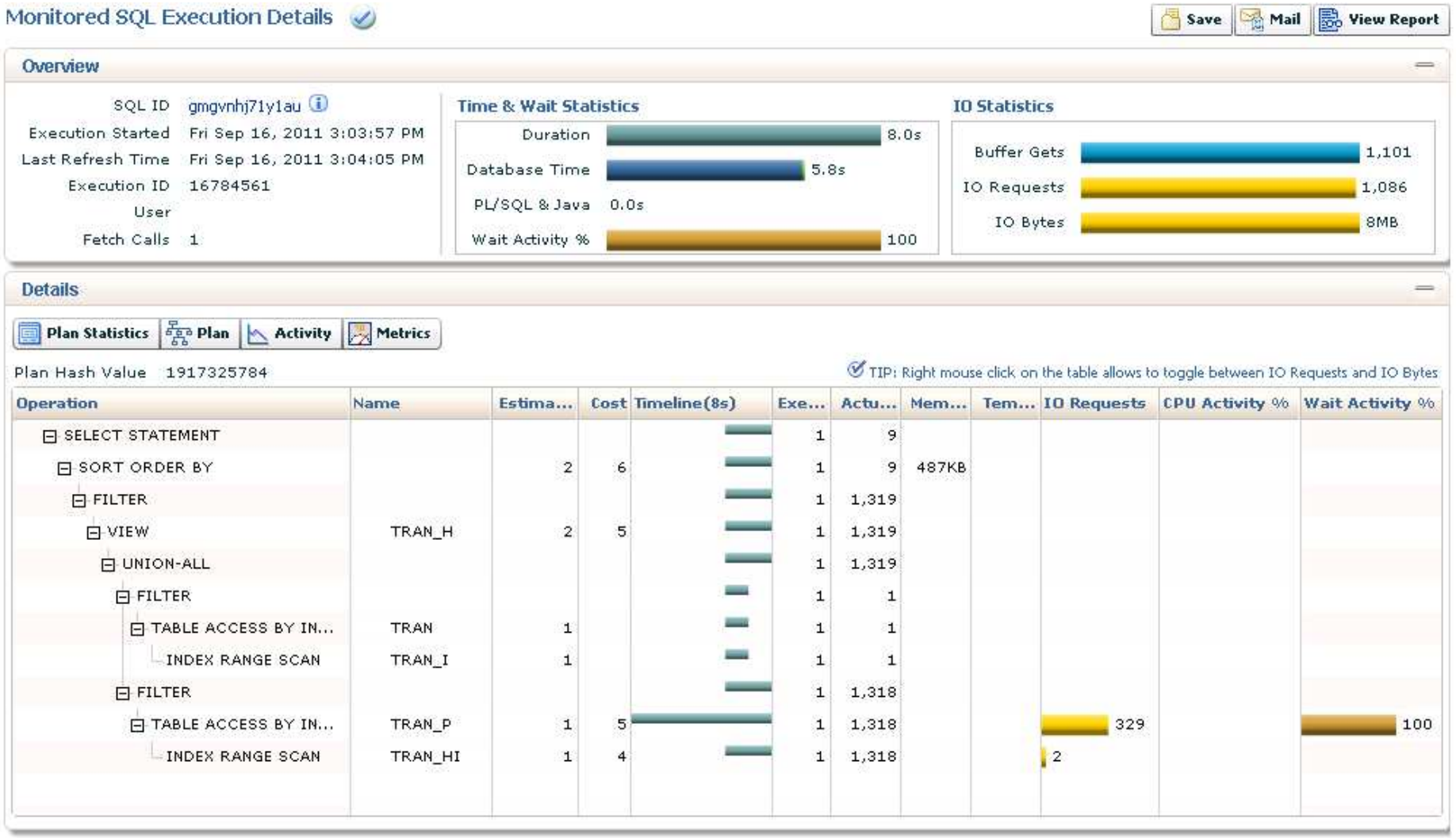
- New **11gR1** feature – requires Tuning pack licensing
- New views **V$SQL_MONITOR, V$SQL_PLAN_MONITOR**
- Automatically kicks in for statements running longer than 5 seconds.
- Captures statistics about SQL execution every second.
- For parallel execution every process involved gets separate entries in V$SQL_MONITOR and V$SQL_PLAN_MONITOR
- Enabled by default for long running statements if parameter CONTROL_MANAGEMENT_PACK_ACCESS if it is set to "DIAGNOSTIC+TUNING" and STATISTICS_LEVEL=ALL|TYPICAL

SQL Monitoring Output

- dbms_sqltune.report_sql_monitor

Operation	Name	Rows (Estim)	Rows (Actual)	Read Reqs	Read Bytes	Mem (Max)	Activity (%)
SELECT STATEMENT			1135				
SORT ORDER BY		24	1135			178K	
VIEW	EB_STMTEND	24	1135				
SORT UNIQUE		24	1135			195K	
UNION-ALL			1135				
NESTED LOOPS			2				
NESTED LOOPS		1	2				
NESTED LOOPS		1	2				
TABLE ACCESS BY INDEX ROWID	EB_ACCOUNT_BAL	1	1				
INDEX UNIQUE SCAN	EB_ACCT_BAL_UIDX	1	1				
TABLE ACCESS BY INDEX ROWID	RB_STMT_MAST_SK	1	2				
INDEX RANGE SCAN	RXM_INTERNAL_KEY_PK	1	2				
INDEX UNIQUE SCAN	RSM_INTERNAL_KEY_PK	1	2				
TABLE ACCESS BY INDEX ROWID	RB_STMT	1	2				
NESTED LOOPS			1133				
NESTED LOOPS		23	1133				
NESTED LOOPS		23	1133				
TABLE ACCESS BY INDEX ROWID	EB_ACCOUNT_BAL	1	1				
INDEX UNIQUE SCAN	EB_ACCT_BAL_UIDX	1	1				
TABLE ACCESS BY INDEX ROWID	RB_STMT_MAST_HIST_SK	23	1133	20	160KB		100.00
INDEX RANGE SCAN	RB_STMT_MAST_HIST_SK_I1	2	1134				
INDEX UNIQUE SCAN	RSM_INTERNAL_KEY_PK	1	1133				
TABLE ACCESS BY INDEX ROWID	RB_STMT	1	1133				

SQL Monitor Output In EM



Automatic Cardinality Feedback (11g)

- ORACLE stores the actual number of rows fetched and uses them to feedback the cardinality on the next execution by using OPT_ESTIMATE hints even before the query transformation.
- **_optimizer_use_feedback** hidden parameter governs the behavior
- It seems that Oracle uses automatic cardinality feedback only when there is a big difference between the estimated cardinality and the actual cardinality.

References:

- Graham Wood: A Brief History of DB Time

Thank you for your interest!

Q&A