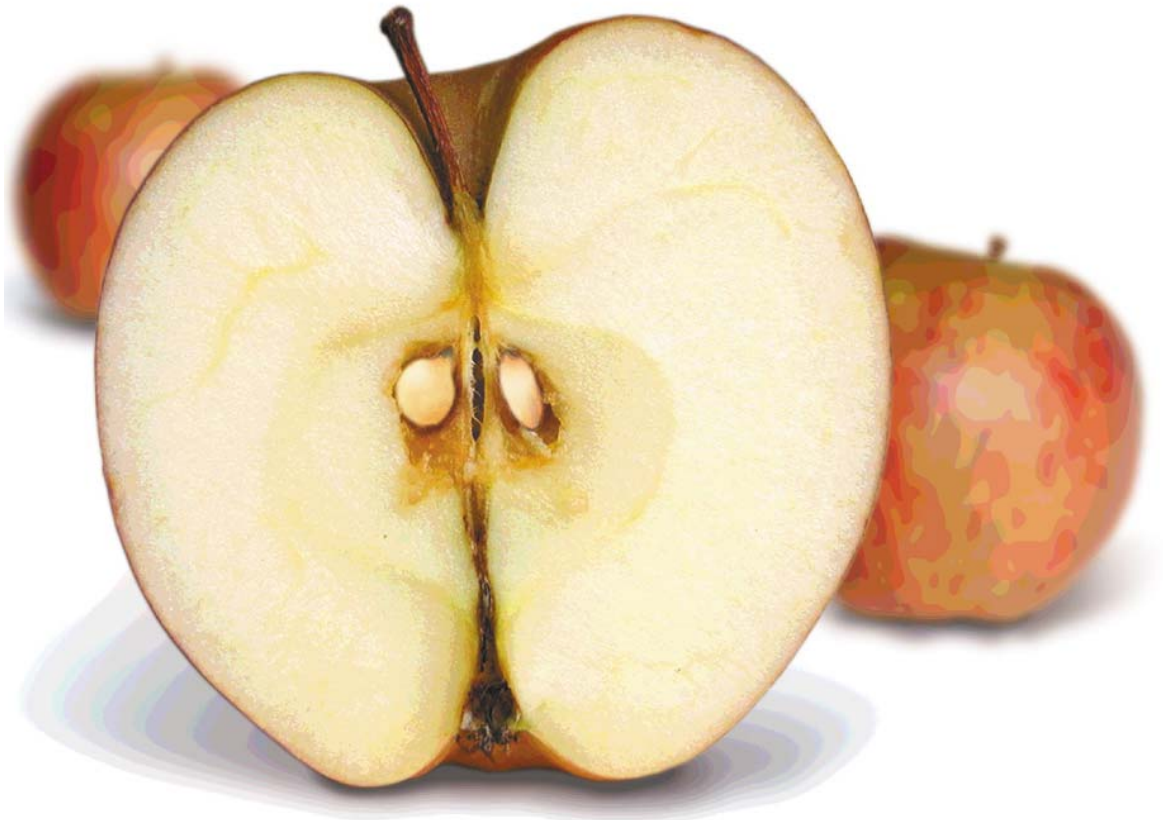
Automatic Workload Repository



DOAG Jahreskonferenz 2004
10/11 November 2004, Mannheim

Christian Antognini
christian.antognini@trivadis.com

Trivadis AG
www.trivadis.com

trivadis
makes IT easier.

Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

Important Questions That Need Answers

- The tuning of a whole system is much more complex than the tuning of a single component (e.g. a report or a user action)
  ➔ Tune single components whenever possible

- For each component the following questions need answers

  ❶ Where is the time spent?
  ❷ How is the time spent?
  } Identification

  ❸ How to reduce the time spent?
  } Tuning

Response Time Tuning

Response Time = Service Time + Wait Time

Cumulated Statistics – The Problem

- Dynamic performance views like V$SYSSTAT shows cumulative statistics since database startup
  - E.g. number of logical I/Os

```
SQL> SELECT sum(value) lio
  2  FROM v$sysstat
  3  WHERE name IN ('db block gets','consistent gets');

       LIO
----------
  57664003
```

- Cumulated statistics are not very useful

  ➔ We need a history
  ➔ We need rates

Cumulated Statistics – The Solution

- What is a rate?
  - Take value v_1 at time t_1
  - Take value v_2 at time t_2
  - Rate = $(v_2 - v_1) / (t_2 - t_1)$
- Statspack provides history and rates
  - Installation required
  - Overhead
  - Limited snapshots
  - Difficult trend analysis
- Thanks to Automatic Workload Repository (AWR) in Oracle 10*g* history and rates are automatically gathered and partially available in memory as well
  - In AWR rates are named metrics

Automatic Workload Repository

- The AWR stores performance statistics used for diagnostic and self-tuning purposes
  - Time model
  - Wait events
  - Metrics
  - Active session history
  - OS statistics
  - SQL statistics (e.g. logical reads per SQL statement)
- Data is both in memory and stored in the database
  - Recent performance statistics are available through V$ views
  - A history over several days is available through DBA_HIST views
  - To reduce the required space, for some statistics, only alerts are stored

Automatic Workload Repository vs. Statspack

- AWR is an enhanced Statspack
- AWR is automatically installed and configured
- AWR has new statistics thanks to 10*g* features
- AWR automatically takes snapshots
  - Default 30 minutes
- AWR automatically purge collected data
  - Default 7 days
- AWR has a GUI (Database/Grid Control)
  - Trivadis has developed a GUI for Statspack: TVD$STAT
- Statspack still exists and has been enhanced as well

Snapshots Interval and Retention Time

- They are shown in DBA_HIST_WR_CONTROL

```
SQL> SELECT snap_interval, retention
  2  FROM dba_hist_wr_control;

SNAP_INTERVAL        RETENTION
-------------------- --------------------
+00000 00:30:00.0    +00007 00:00:00.0
```

- They can be modified through DBMS_WORKLOAD_REPOSITORY (in minutes)

```
SQL> BEGIN
  2    dbms_workload_repository.modify_snapshot_settings(
  3      interval  => 60,
  4      retention => 10*24*60
  5    );
  6  END;
  7  /
```

Automatic Workload Repository Reports

- They can be generated in text or HTML
  - Database/Grid Control
  - Package DBMS_WORKLOAD_REPOSITORY
  - Scripts awrrpt.sql (current DBID and instance) or awrrpti.sql (DBID and instance are requested)
- Very similar to Statspack reports
- The script awrinfo.sql generates general information about AWR such as the size and data distribution

Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

Time Model Statistics

- Two new views give high-level information about the used CPU and elapsed time for different database operations
  - V$SYS_TIME_MODEL
  - V$SESS_TIME_MODEL
  - The amount of time is given in microseconds
- The statistic “DB time” represents the total time spent in database calls; this doesn’t include the time spent by background processes
- Since waits are part of the elapsed time, “DB time” can be much larger that the elapsed time since instance startup
- Some of the information provided by these views is already available in V$SYSTAT and V$SESSTAT

Time Model Statistics – Components

- background elapsed time
  - background cpu time
- DB time
  - DB CPU
  - connection management call elapsed time
  - failed parse elapsed time
    - failed parse (out of shared memory) elapsed time
  - inbound PL/SQL rpc elapsed time
  - Java execution elapsed time
  - parse time elapsed
    - hard parse elapsed time
      - hard parse (sharing criteria) elapsed time
        - hard parse (bind mismatch) elapsed time
  - PL/SQL compilation elapsed time
  - PL/SQL execution elapsed time
  - sequence load elapsed time
  - sql execute elapsed time

Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

What Is the Wait Interface?

- Since version 7.0 the Oracle code has been instructed to report the time waited completing internal operations
- Oracle calls these internal operations events or waits
- The gathered information is available through V$ views and trace facilities (e.g. event 10046)
- The wait interface is the most important source of information used to analyze and/or discover performance problems

What Is the Wait Interface?

- In the following example a TVD$XTAT output shows that a statement "waited" about 17 seconds (difference between CPU and elapsed time)

Call	Misses	Count	CPU [s]	Elapsed [s]	PIO [b]	LIO[b]	Consistent [b]	Current [b]	Rows
Parse	1	1	0.010	0.014	0	0	0	0	0
Execute	0	1	0.120	17.699	1	63	7	56	17
Fetch	0	0	0.000	0.000	0	0	0	0	0
Total	1	2	0.130	17.713	1	63	7	56	17
Average (per row)	0	0	0.008	1.042	0	3	0	3	1

- Where does this difference come from?

What Is the Wait Interface?

- If the trace file has been generated with extended SQL trace information, e.g. through DBMS_MONITOR, the answer can be found in the TVD$XTAT output as well

Component	Total Duration [s]	%	Number of Events	Duration per Event [s]
enq: TM - contention	17.357	82.382	6	2.893
SQL*Net message from client	2.921	13.862	1	2.921
recursive statements	0.490	2.326	n/a	n/a
db file sequential read	0.171	0.813	1	0.171
CPU	0.130	0.617	n/a	n/a
SQL*Net message to client	0.000	0.000	1	0.000
Total	21.069	100.000		

→ The statement waited about 17 seconds acquiring a row lock

- This information comes from the wait interface

What Is New in the Wait Interface?

- More events are available:
  - 10.1.0.3 → 811
  - 9.2.0.5 → 402
  - 8.1.7.4 → 217
  - 7.3.4.5 → 105
- Waits have been grouped in 12 classes
  (e.g. Application, Cluster, Concurrency, Network, User I/O)
- More V$ views contains wait interface information
  (e.g. V$SESSION contains the information that was previously available in V$SESSION_WAIT)
- New V$ views

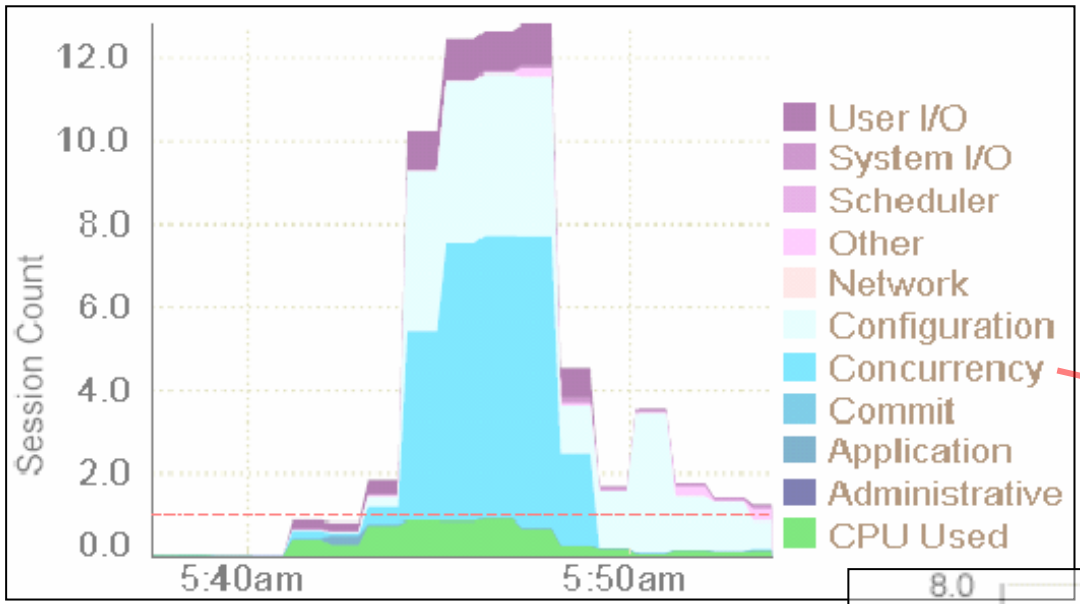
Wait Classes

- In V$EVENT_NAME new columns specify the wait class
- V$SESSION_WAIT_CLASS displays for each session the wait time (in centiseconds) grouped by class

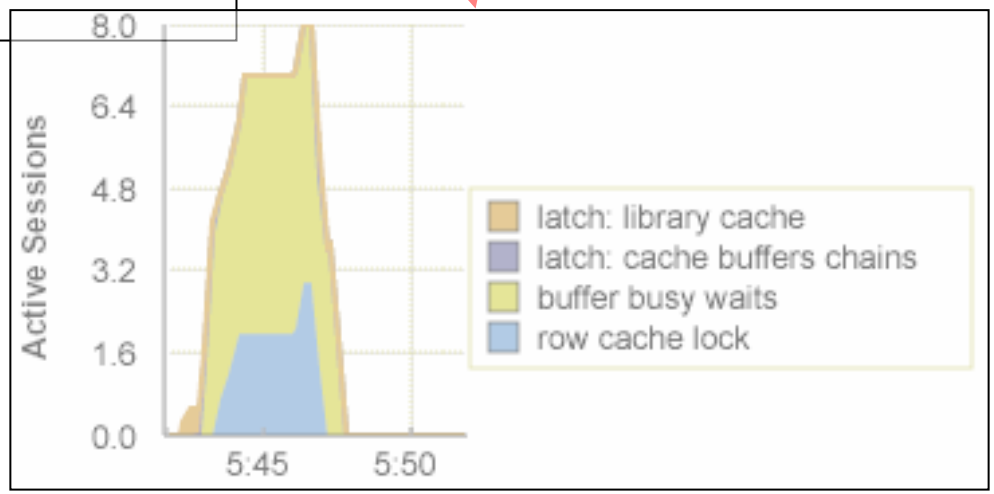
```
SQL> SELECT wait_class, total_waits, time_waited
  2  FROM v$session_wait_class
  3  WHERE sid = 17;

WAIT_CLASS                TOTAL_WAITS TIME_WAITED
------------------------- ----------- -----------
Idle                              466       31640
User I/O                         1613         317
System I/O                         46           3
Application                         2           1
Network                           135           0
```

- V$SYSTEM_WAIT_CLASS and V$SERVICE_WAIT_CLASS display wait class information for the system/services since startup

Wait Classes in Database/Grid Control



Drill down



Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

Which Metrics Exists?

- Metrics are available for
  - System
  - Sessions
  - Services
  - Events
  - Files
- They are organized in groups
- They are computed over an interval in the following units
  - Absolute values
  - Percentages
  - Per second and per transaction
- For each metric a warning and critical threshold can be configured

Which Metrics Exists?

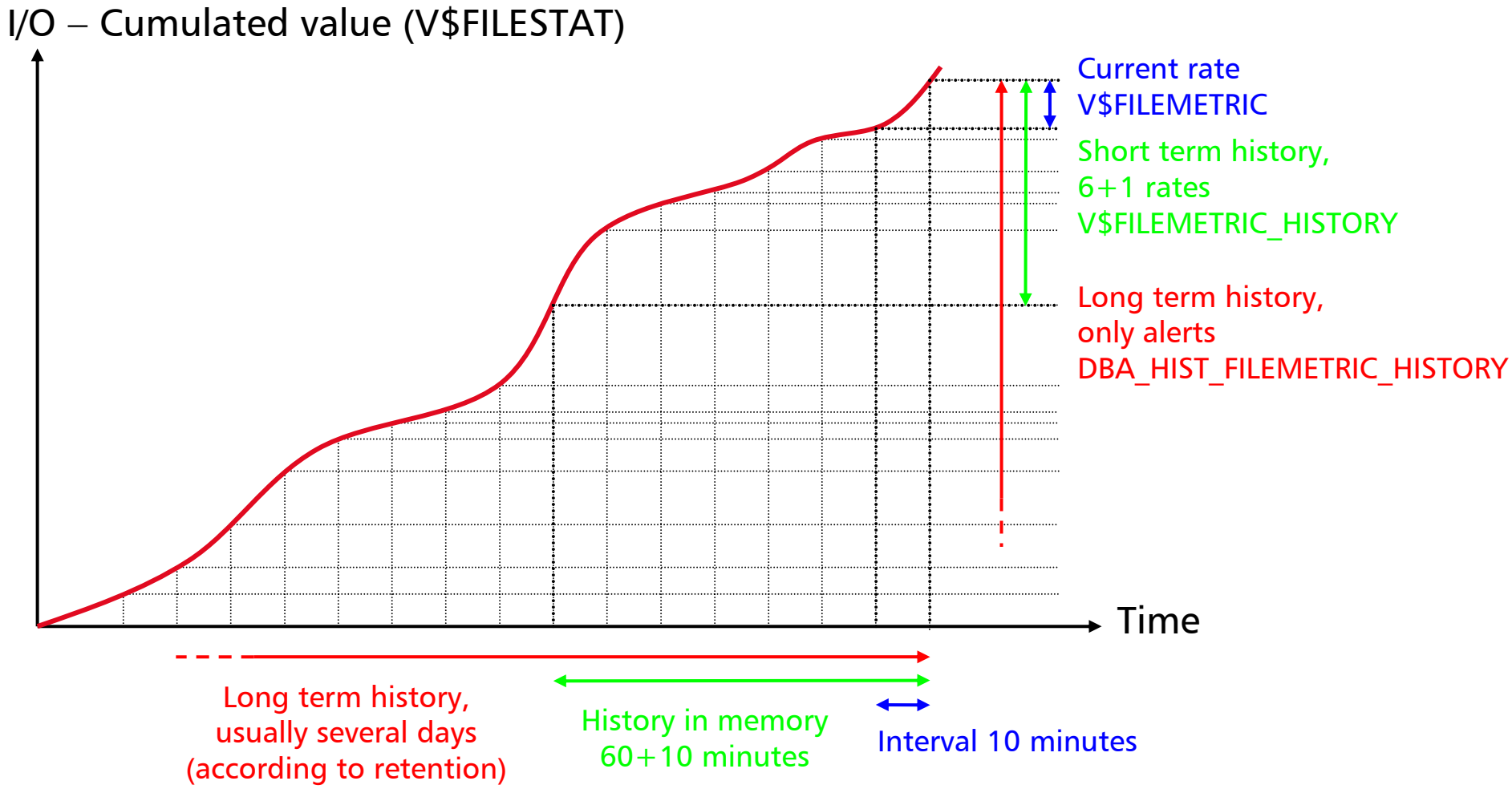
```
SQL> SELECT g.group_id id, g.name, g.interval_size, g.max_interval,
  2         count(*) cnt
  3  FROM v$metricgroup g, v$metricname n
  4  WHERE g.group_id = n.group_id
  5  GROUP BY g.group_id, g.name, g.interval_size, g.max_interval;

ID NAME                            INTERVAL_SIZE MAX_INTERVAL CNT
-- ------------------------------ ------------- ------------ ---
 0 Event Metrics                           6000            1   3
 1 Event Class Metrics                     6000           60   4
 2 System Metrics Long Duration            6000           60 119
 3 System Metrics Short Duration           1500           12  40
 4 Session Metrics Long Duration           6000           60   1
 5 Session Metrics Short Duration          1500            1   8
 6 Service Metrics                         6000           60   2
 7 File Metrics Long Duration             60000            6   6
 9 Tablespace Metrics Long Duration        6000            0   1
```

Number of metrics for each “statistic”,
e.g. each event has 3 metrics

Which Metrics Exists?

```
SQL> SELECT group_id id, metric_name, metric_unit
  2  FROM v$metricname
  3  ORDER BY group_id;

 ID METRIC_NAME                          METRIC_UNIT
--- ------------------------------------ -------------------------
  0 Total Wait Counts                    Waits
  0 Total Time Waited                    CentiSeconds
  0 Number of Sessions Waiting (Event)   Sessions
  1 Total Wait Counts                    Waits
  1 Total Time Waited                    CentiSeconds
  1 Database Time Spent Waiting (%)      % (TimeWaited / DBTime)
  1 Average Users Waiting Counts         Users
  2 Host CPU Utilization (%)             % Busy/(Idle+Busy)
  2 Database Time Per Sec                CentiSeconds Per Second
  2 Txns Per Logon                       Txns Per Logon
  2 Executions Per Sec                   Executes Per Second
...
  7 Average File Read Time (Files-Long) CentiSeconds Per Read
  9 Tablespace Space Usage               Tablespace Blocks

184 rows selected.
```

Current and History Rates – File Metrics Example



Views

- Information
  - V$METRICNAME
  - V$METRICGROUP
  - DBA_HIST_METRIC_NAME
- Global (contains all metrics)
  - V$METRIC
  - V$METRIC_HISTORY
- Service
  - V$SERVICEMETRIC
  - V$SERVICEMETRIC_HISTORY
- Session
  - V$SESSMETRIC
  - DBA_HIST_SESSMETRIC_HISTORY(a)
- System
  - V$SYSMETRIC
  - V$SYSMETRIC_HISTORY
  - V$SYSMETRIC_SUMMARY
  - DBA_HIST_SYSMETRIC_HISTORY(a)
  - DBA_HIST_SYSMETRIC_SUMMARY
- Events
  - V$EVENTMETRIC
  - V$WAITCLASSMETRIC
  - V$WAITCLASSMETRIC_HISTORY
- Files
  - V$FILEMETRIC
  - V$FILEMETRIC_HISTORY
  - DBA_HIST_FILEMETRIC_HISTORY(a)

Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

Active Session History

- Active Session History (ASH) is a history of the information similar to that available in V$SESSION and V$SESSION_WAIT
  - V$SESSION_WAIT shows on which event a session is waiting
  - V$SESSION_WAIT_HISTORY shows the last 10 waits
- A session is active if it uses CPU or waits for an event that doesn't belong to the wait class "Idle"
- Active sessions are sampled every second and stored in a circular buffer in the SGA
  - V$ACTIVE_SESSION_HISTORY
  - The retention is extremely variable
- History available through DBA_HIST_ACTIVE_SESS_HISTORY
  - Only a subset of V$ACTIVE_SESSION_HISTORY is stored on disk

Session Wait History – Example

- The following statement shows for what events waited the foreground sessions during the last hour

```
SQL> SELECT event, count(*), sum(time_waited) time_waited
  2  FROM v$active_session_history
  3  WHERE sample_time >= sysdate - to_dsinterval('0 1:0:0')
  4  AND session_type = 'FOREGROUND'
  5  GROUP BY event
  6  ORDER BY 3 DESC;

EVENT                                COUNT(*) TIME_WAITED
------------------------------ ---------- -----------
enq: TX - row lock contention           525   517191348
enq: TM - contention                    149   126135243
db file sequential read                  61     3240729
latch: library cache                      6     2262512
latch: enqueue hash chains               11     1622636
log file sync                            34     1582910
latch free                                1      549162
buffer busy waits                         1      279213
…
```

Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

OS Statistics

- The cumulated values are available in V$OSSTAT
  - Totals over all processors and average values are available
  - The history is stored in DBA_HIST_OSSTAT

```
SQL> SELECT stat_name, value
  2  FROM v$osstat
  3  WHERE stat_name NOT LIKE 'AVG%';

STAT_NAME                            VALUE
------------------------------ ----------
NUM_CPUS                                1
IDLE_TICKS                       35920275
BUSY_TICKS                        2884344
USER_TICKS                         950628
SYS_TICKS                         1907011
NICE_TICKS                          26705
RSRC_MGR_CPU_WAIT_TIME                  0
IN_BYTES                      4.7691E+10
OUT_BYTES                     2.3344E+11
```

Performance Impact

- Of course the gathering and processing of a large amount of performance statistics has a performance impact on the database
- Usually this overhead can be neglected
  - Oracle says that during a TCP benchmark the overhead was 3%
- To disable AWR the INIT.ORA parameter STATISTICS_LEVEL can be set to BASIC

Licensing Information

- To use the following features/components the Diagnostic Pack must be licensed
  - Package DBMS_WORKLOAD_REPOSITORY
  - View V$ACTIVE_SESSION_HISTORY
  - All views with the prefix DBA_HIST
  - Report awrrpt.sql and awrrpti.sql

- This means that per default when a database is created it contains features/components that cannot be used!
- For more information refer to the document *Oracle Database Licensing Information 10g Release 1*
  - The document also describes on which links of the Database/Grid Control you can click…

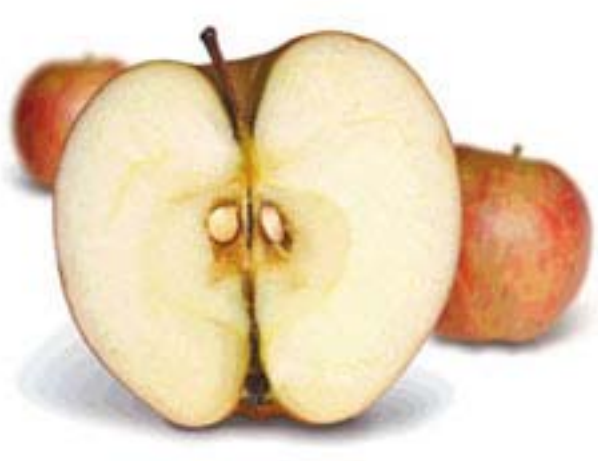
Automatic Workload Repository

> Agenda

Be in
the know.

- > Introduction
- > Time Model
- > Wait Interface
- > Metrics
- > Active Session History
- > Miscellaneous
- > Conclusion

Automatic Workload Repository: Core Statements…



> from Trivadis

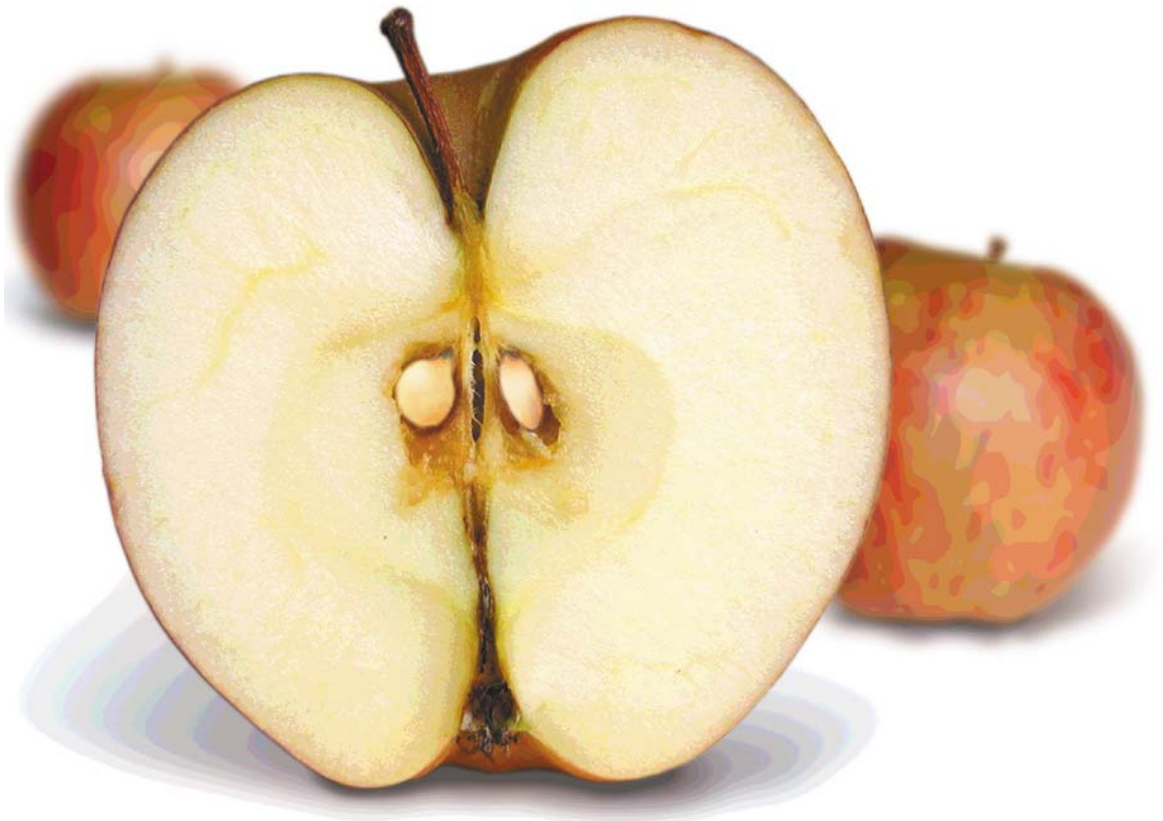
At the core it's all about data.

\+ A lot of very interesting performance statistics with history and rates

! Such a large amount of data increase complexity
  - » Which view contains the information I'm looking for?

– Questionable licensing strategies…

Thank you for your attention



?

Questions?

trivadis
makes IT easier.