



AWR & ADDM are like YIN & YANG

- 7 Business Cases Explained

Yuri van Buren

Senior Oracle DBA Specialist
End-2-End Performance Management Engineer

© CGI Group Inc. 2013

CGI

Experience the commitment®

Yuri van Buren

18 Years with CGI

DBA = Database Administrator

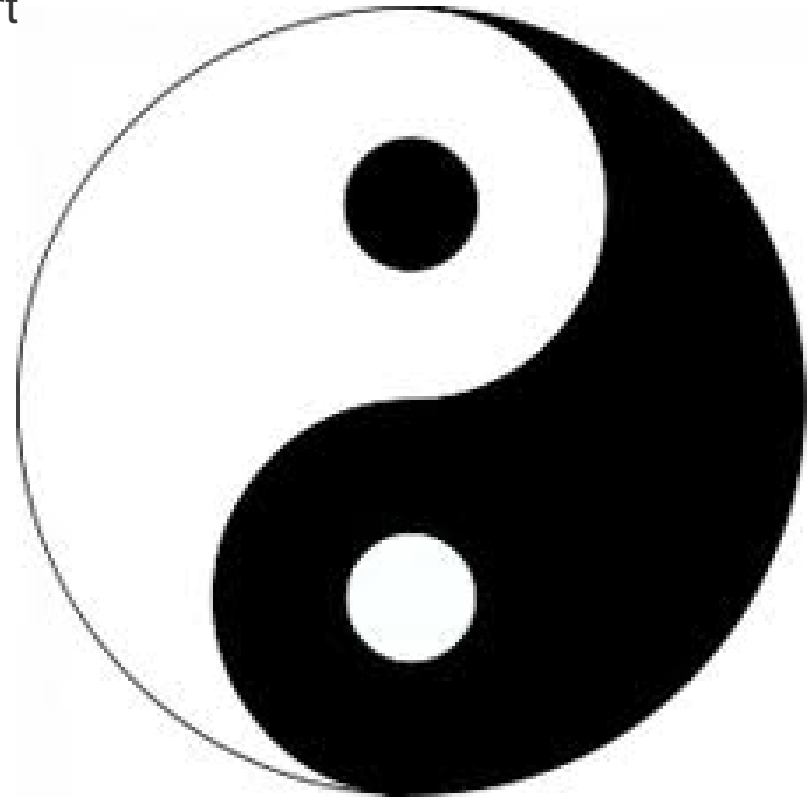
- Operational DBA
- DBA Consultant
- Database Specialist
- Project DBA
- APPS (or EBS) DBA
- DBA Architect
- OCP10g & 11g; OCE 11g Performance Tuning
- E2E Performance Management Engineer
- Principal Consultant

Fourth time presenting at the DOAG



Agenda

- Basics of generating the Automatic Workload Repository (AWR) and Automatic Database Diagnostics Monitor (ADDM) reports
- My method how to read an AWR report
- 7 Business Cases Explained
Combining the power of AWR & ADDM



Basics of generating AWR & ADDM reports

Scripts to create the reports are available on the Oracle RDBMS Server from 10g onwards. They are located in the \$ORACLE_HOME/rdbms/admin directory.

Script to generate the AWR report – awrrpt.sql (Main report in text or HTML)

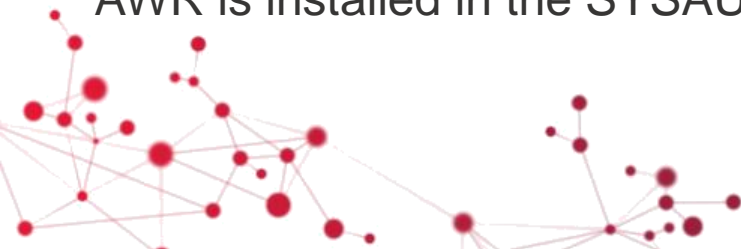
Other great AWR scripts

- -awrsqrpt.sql (Workload report for a particular sql_id)
- -awrddrpt.sql (Workload Repository Compare Periods report)
- -awrextr.sql (To extract AWR data with datapump)
- -awrload.sql (To load data from an awrextr dump).

Script for ADDM – addmrpt.sql (Only text output possible).

AWR is necessary for the advisors to give recommendations.

AWR is installed in the SYSAUX tablespace and cannot be moved!



Basics Part 2: Setting interval and retention times

AWR is enabled by default every hour and kept for 7 days (10g) or 8 days (11g) only! Query the current values with:

```
select snap_interval, retention from dba_hist_wr_control;
```

SNAP_INTERVAL	RETENTION
+00000 01:00:00.0	+00007 00:00:00.0

Setting interval and retention times example (20 minutes interval and 14 days retention time):

```
begin
```

```
    dbms_workload_repository.modify_snapshot_settings (  
        interval => 20, retention => 14*24*60  
    );
```

```
end;
```

```
/
```

*Note that setting interval to **zero** minutes, disables AWR data collection!!*

Make a manual snapshot with: execute dbms_workload_repository.create_snapshot;

Performance Mind Set

The top 3 performance symptoms in Oracle Databases are:

- Too many Logical IO's
- Too many or slow Physical IO's
- Too much contention (e.g. Latch and/or Lock contention).



How to read an AWR report (... My Way...)

1) Check that you look at the correct database instance on the correct machine and check that the time interval is the one you want to analyze. Calculate the **average active sessions** (DB Time/Elapsed Time).

When AAS >> Nr of CPU's on the server you are in trouble.



2) Go to the top Timed Events section and see which Events are on top.

Is it **CPU**, **physical IO** (db file sequential read + db file scattered read, direct path read) or physical IO caused by the Logwriter visible as log file sync or is it **contention** e.g. Latch issues (buffer busy waits, library cache, shared pool etc). or Locking (Enqueue) issues.

Next step depends on the main time contributor found.



Be aware of the following performance facts

Fact : DB Time increases as system load increases.

Fact : On a CPU bound host all wait events are artificially inflated.

Fact : "db file ..." read times includes run-queue time.

Therefore you need to tune for CPU, before Waits when CPU is constrained.
High CPU usage is most often caused by too many Logical IOs per second.

So reduce the number of Logical IOs before focussing on Waits on a CPU bound host.



Approach with High CPU in DB Time

From the ***CPU used by this session*** metric and the **Logical Reads per second** metric you can calculate and “spec” your instance its “Effective CPU Speed”, expressed in μs per LIO.

The reciprocal value of this gives you the maximum amount of LIOs/sec that your Oracle instance can run on one CPU.

E.g. *Logical reads per second*: 14180 and *CPU used by this session* per second 20.7

□ $0.207/14180 = 14,5\mu\text{s}$ CPU per LIO.

And reciprocal it is $14180/0.207 = 68502$ LIOs/sec per CPU.

Go to the top LIO per segment section. Check how many objects cause in total 80% of the load. Sometimes only 1 to 5 segments are causing almost all of the LIOs. Go to the “SQL ordered by (buffer) Gets section”. A bufferget is a synonym for a Logical IO. See which queries take the most time.



Approach with High Physical Reads in DB Time

Calculate the average physical IO times for the scattered and sequential read.
For the whole instance the I can calculate the average PIO call time:

Average PIO call (ms) = (time waited for scattered and sequential reads) / physical reads count

This gives an indication if their are slow disks (E.g. PIO Call times well above 6 or 10ms depending on your disk setup) or just many PIOs going on (or both).

Go to the top PIO per segment section. Check how many objects cause in total 80% of the load. Sometimes only 1 to 5 segments are causing almost all of the PIOs. Go to the “SQL ordered by Reads section”. A “Read” is a synonym for a Physical IO. See which queries take the most time.



Approach with High Contention

Look in the respective sections of “Latch Activity” or “Enqueue Activity” to determine the type of problem. Try to understand why Oracle needs so many latches and/or enqueues at the same time.

If it is the dictionary of shared pool latch you might have a `shared_pool_size` that is too small.

If it is the “cache buffer chain latch”, you might see a symptom of a system that is doing way too much LIOs/sec. It might be that you are scanning a lot of unselective indexes for instance. Fixing the excessive LIO problem often fixes the latch issue.



7 Business Cases Explained

- 1) High CPU time
- 2) High SQL*Net more data from client
- 3) High direct path reads
- 4) High CPU time ... Is it SQL or PL/SQL?
- 5) High "enq: TX - row lock contention"
- 6) High Buffer Busy Waits
- 7) High Log File Sync and Log File Parallel Write times



I will try to show that the assumptions from reading an AWR report can be made very firm (up to 100%) by combining the AWR data with data from ADDM.



Business Case 1

High CPU Usage



Case1_awrrpt_1_19229_19231.txt



Case1_addm_19229_19231.txt



Business Case 2

High SQL*Net more data from client



Case2_awrrpt_1_10545_10549.txt



Case2_addmrpt_1_10545_10549.txt



Business Case 3

High direct path reads



Case3_awrrpt_1_25195_25196.txt



Case3_addmrpt_1_25195_25196.txt



Business Case 4

High CPU time ... Is it SQL or PL/SQL?



Case4_awrpt_1_16342_16348.txt



Case4_addmrpt_1_16342_16348.txt



Business Case 5

High "enq: TX - row lock contention"



Case5_awrrpt_1_1155_1157.txt



Case5_addmrpt_1_1155_1157.txt



Business Case 6

High Buffer Busy Waits



Case6_awrrpt_1_1244_1245.txt



Case6_addmrpt_1_1244_1245.txt



Business Case 7

High Log File Sync and Log File Parallel Write



Case7_awrrpt_1_1238_1239.txt



Case7_addmrpt_1_1238_1239.txt



Conclusion

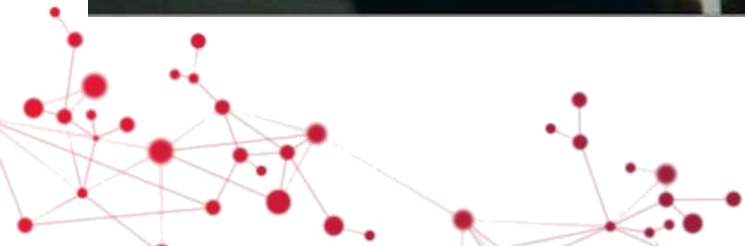
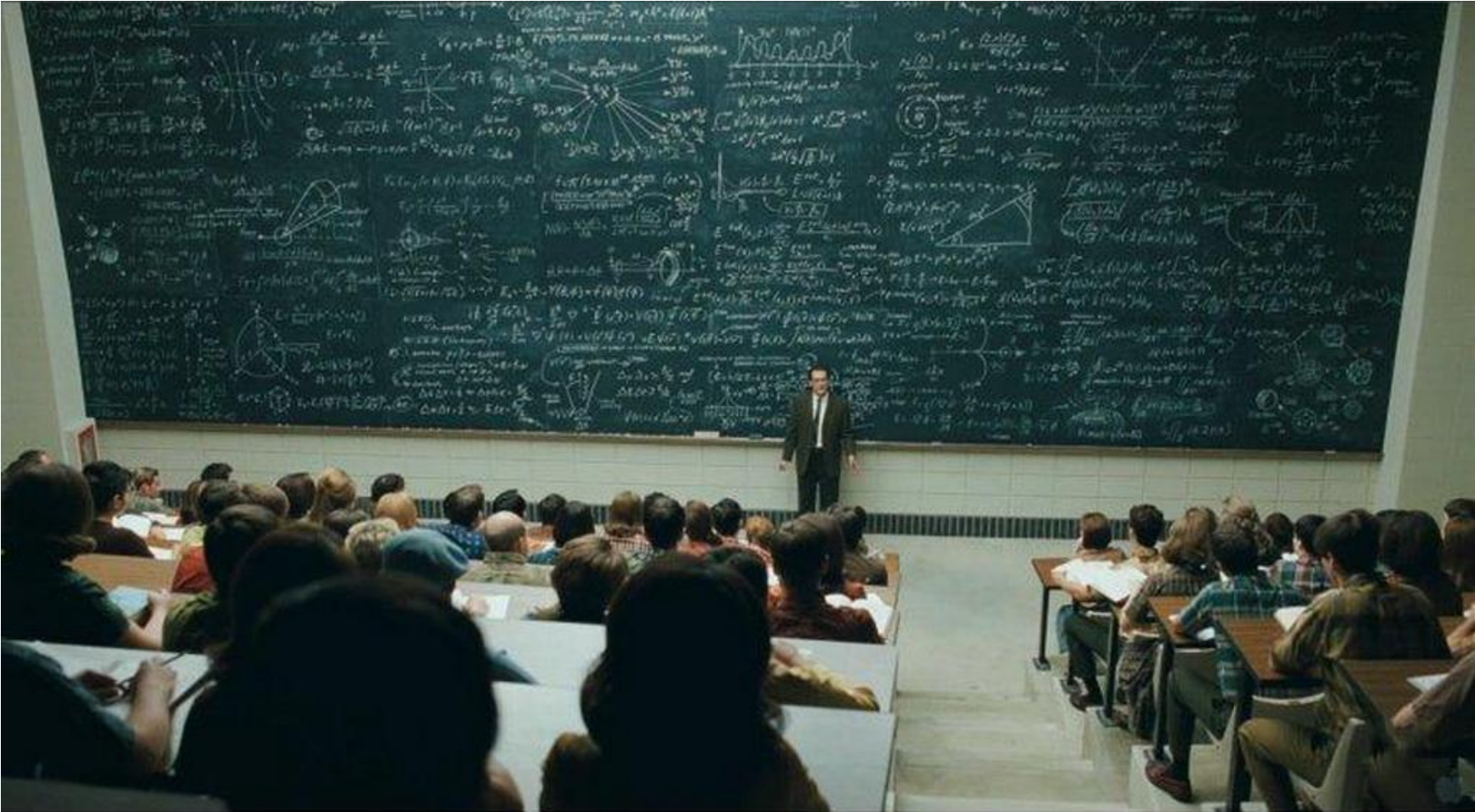
The assumptions from reading an AWR report can be made very firm (up to 100%) by combining the AWR data with data from ADDM.

Recommendation

From now on I hope you will always generate both AWR and ADDM reports from the same time and use the power of these two tools together to provide you with more proof (up to 100%) of performance issues that occur.



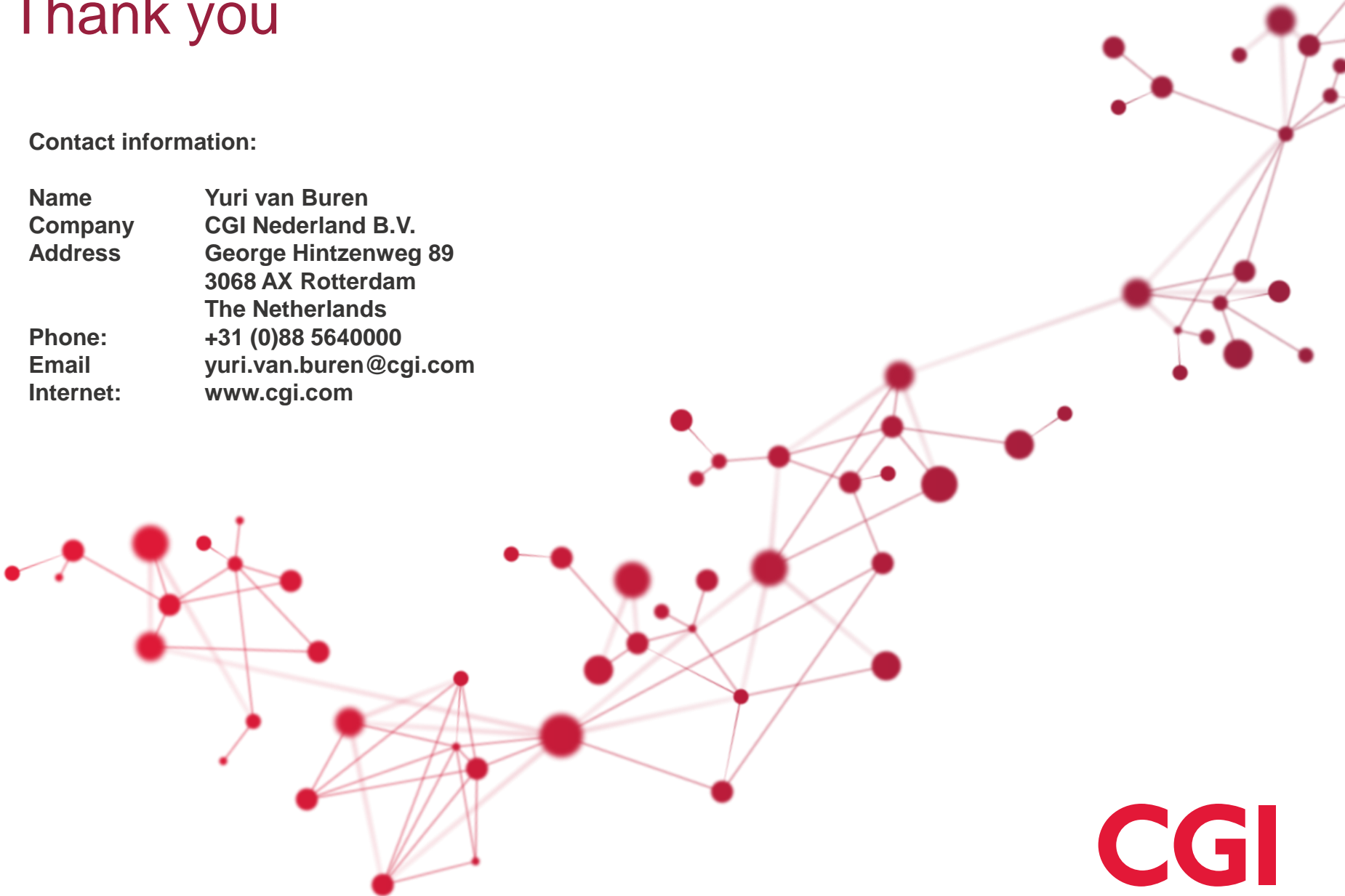
Questions?



Thank you

Contact information:

Name	Yuri van Buren
Company	CGI Nederland B.V.
Address	George Hintzenweg 89 3068 AX Rotterdam The Netherlands
Phone:	+31 (0)88 5640000
Email	yuri.van.buren@cgi.com
Internet:	www.cgi.com



CGI

Experience the commitment®