

Exploring and exploiting ASH data with maximum efficiency

Nicolas Jardot
dbi services

Keywords

Oracle Database, Performance, Tuning, ASH,

Introduction

When users are complaining about the database performance, we need some data and metrics to analyze the cause and troubleshoot the issue. Oracle provides several layers of data we can look at to have an idea of what is happening on the system:

- Oracle statistics: exposed through performance views and aggregate in AWR/Statspack snapshots
- Session tracing: in depth activity of one session
- Active Session History (ASH): information about all sessions active on the database

This session focuses in ASH, how it works and how we can exploit the data.

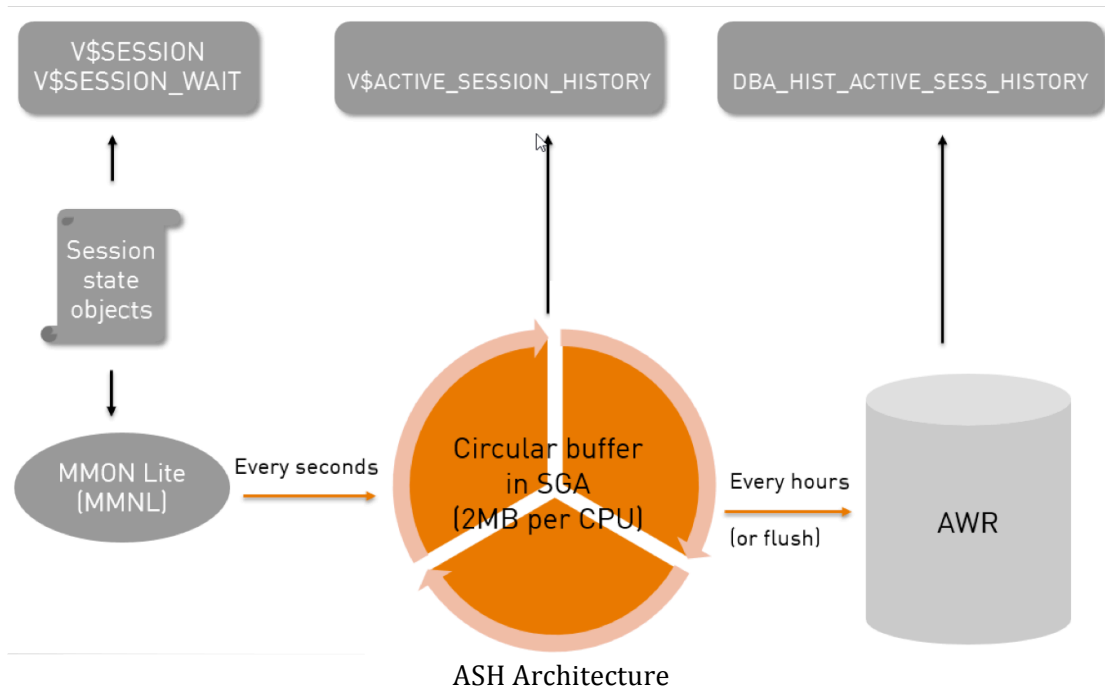
ASH Mechanism

First, what is ASH? ASH stands for Active Session History and it has been introduced in Oracle 10g.

It works as a sampler. By default the database takes a snapshot every seconds of all active session including both foreground and background activity and records the details in a memory buffer exposed through the view V\$ACTIVE_SESSION_HISTORY.

The goal is to keep at least the last hour activity in memory then 1/10 of the snapshots are saved in AWR view DBA_HIST_ACTIVE_SESS_HISTORY.

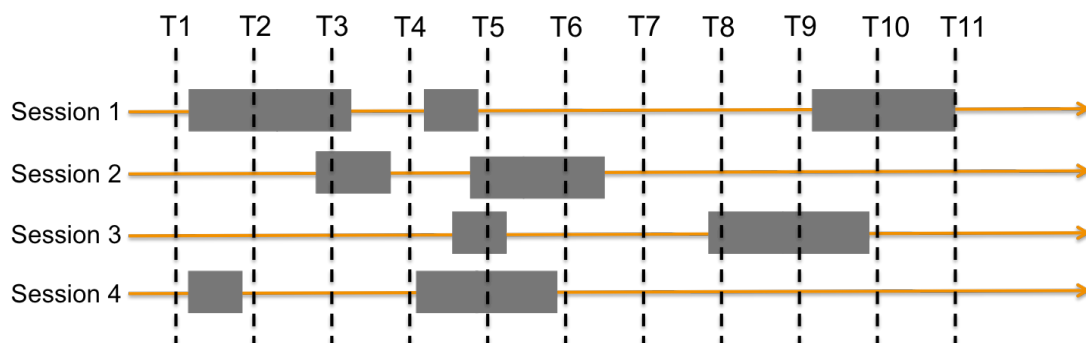
Basically, the in-memory views contains fields extracted from V\$SESSION view like the SID and the current EVENT of the session. Having snapshot every seconds provide an accurate view of the database activity.



ASH Architecture

The process MMON Lite is responsible for extracting in a non-blocking way information from V\$SESSION and V\$SESSION_WAIT every seconds.

Then every hours or if buffer is full, 1/10 of the samples are flushed into AWR views to keep history.



Sampling mechanism

ASH captures data only for active session at the time of the snapshot. The activity for session 4 in the above example is never captured whereas all activity from session 3 should have been captured.

ASH is part of Diagnostics Pack and should be an Enterprise Edition only feature but there are some tools to simulate the same behavior in Standard Edition like L-ASH.

Read ASH Data

Name	Null?	Type
-----	-----	-----
SAMPLE_ID		NUMBER
SAMPLE_TIME		TIMESTAMP (3)
IS_AWR_SAMPLE		VARCHAR2 (1)
SESSION_ID		NUMBER
SESSION_SERIAL#		NUMBER
SESSION_TYPE		VARCHAR2 (10)
...		
SQL_ID		VARCHAR2 (13)
IS_SQLID_CURRENT		VARCHAR2 (1)
...		
EVENT		VARCHAR2 (64)
WAIT_CLASS		VARCHAR2 (64)

ASH table definition extract

The view V\$ACTIVE_SESSION_HISTORY can be viewed as a fact table. To read data from this view, we have to choose a dimension to filter the data.

In order to read the data, there is a very important point, we should always delimit a time frame.

As ASH is sampling, we know that activity during more than 1 second will be always captured but that active session for less than 1 second may or may not be captured. Moreover, a session waiting for more than 2 seconds will be captured several times. Due to that architecture, function like sum or average should not be used on that view.

Based on the fact that there is a snapshot every seconds, the only thing that is needed to know the time is to count the number of samples, it will corresponds to the number of seconds where the activity has been captured.

For example, if you need to get the list of the top SQL onto which the database spent most of its time: dimension is sql_id, time frame is the last hour:

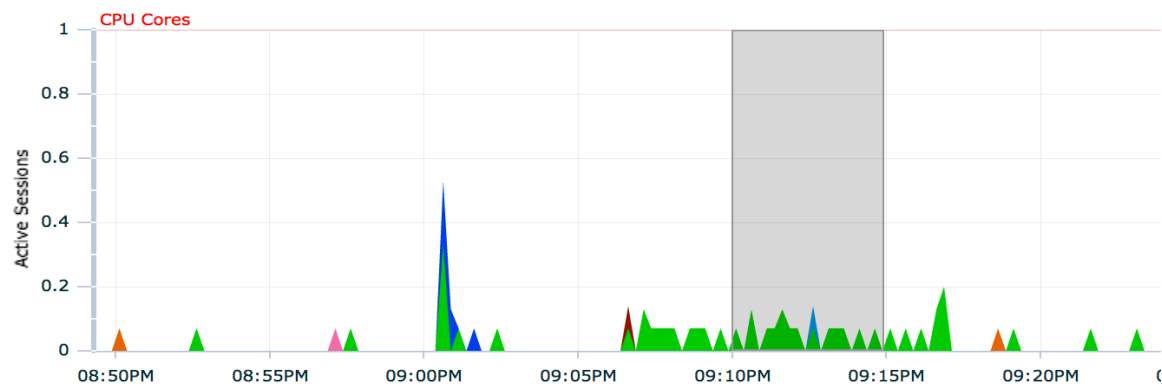
```
SELECT sql_id, count(*),
round(count(*)/sum(count(*) over (), 2) pctload
FROM v$active_session_history
WHERE sample_time > sysdate - 1/24/60
AND session_type <> 'BACKGROUND'
GROUP BY sql_id ORDER BY count(*) desc;
```

Exploit ASH Data

What can we do with ASH data? As seen just before, we can run custom scripts to get the information we are interested in, especially for getting tops (top SQL, top Event, top session...).

Having sample every seconds provides the number of active session in the database per seconds, more often called AAS for Average Active Sessions just by counting the number of session captured in each snapshot.

Based on that, we retrieve ASH data in a very well know place for people using Oracle Grid/Cloud Control: the graphic and the two tables in the Top Activity page.



Top activity screen extract

One of the most efficient ways of exploiting ASH data is to draw graphics to get the number of sessions and more important which type of activity they are performing (CPU, I/O...)

Top activity screen is limited to a 5 minutes interval for details but Oracle provides ASH Analytics that allows a deeper analysis. The interval can be customized and the dimension of analysis can also be changed.

In addition to Oracle tools, there is an interesting tool that allows exploiting ASH data called Orachrome Lighty. The benefit of a tool like Lighty is that it's fully compatible with both Enterprise and Standard Edition. With such tool you can have the same tuning analysis for both type of databases. In Standard Edition the tool relies on L-ASH and Statspack to provide same information as ASH and AWR.

L-ASH is a set of jobs that simulates ASH by extracting the content of V\$SESSION every seconds into a table and purging the table on a regular basis.

Contact Address:

Nicolas Jardot
dbi services sa
Avenue de la gare 42
2800 Delémont
Switzerland

Phone: + 41 79 835 02 95
Email: nicolas.jardot@dbi-services.com
Website: <http://www.dbi-services.com>