Ash and AWR Performance Data

Kellyn Pot'Vin
Sr. Technical Consultant
Enkitec

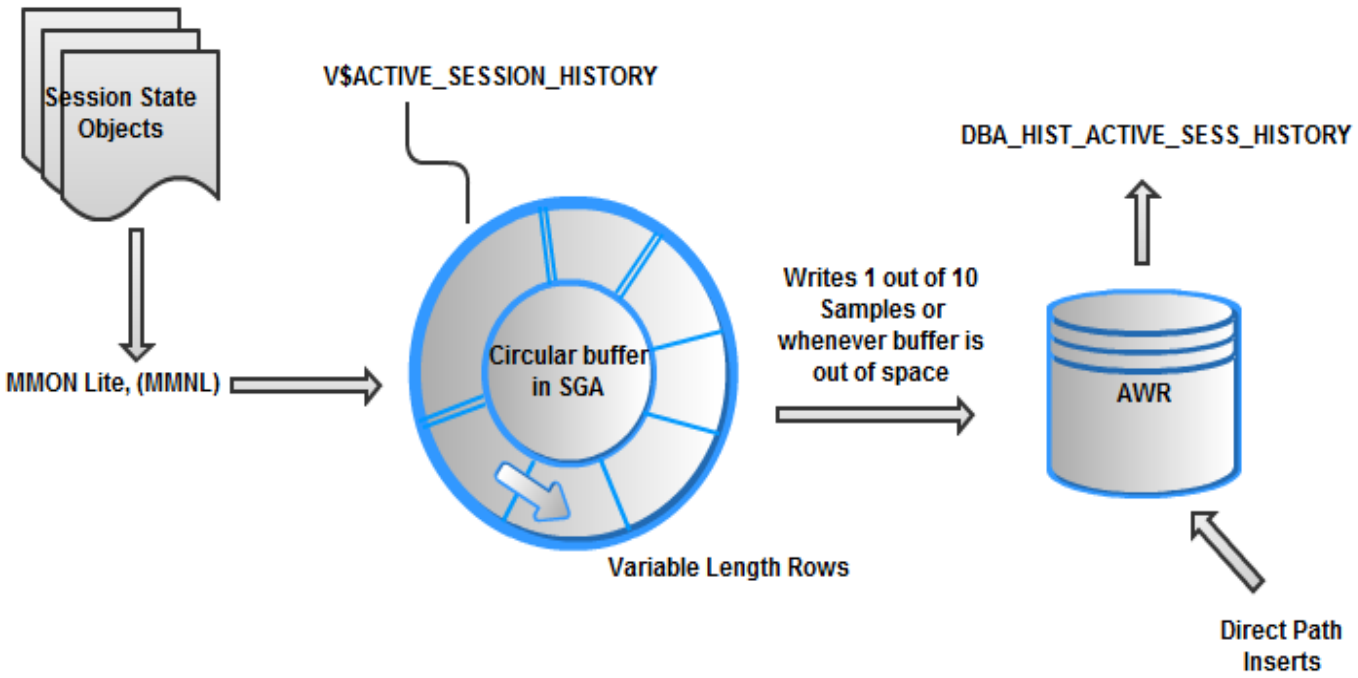
A Little About Me:

- Sr. Technical Consultant for Enkitec
- Multi-Platform Tuning Specialist
- (Lately) EM12c Specialist
- Oracle ACE
- Training Days Director for RMOUG
- Blog at DBAKevlar.com

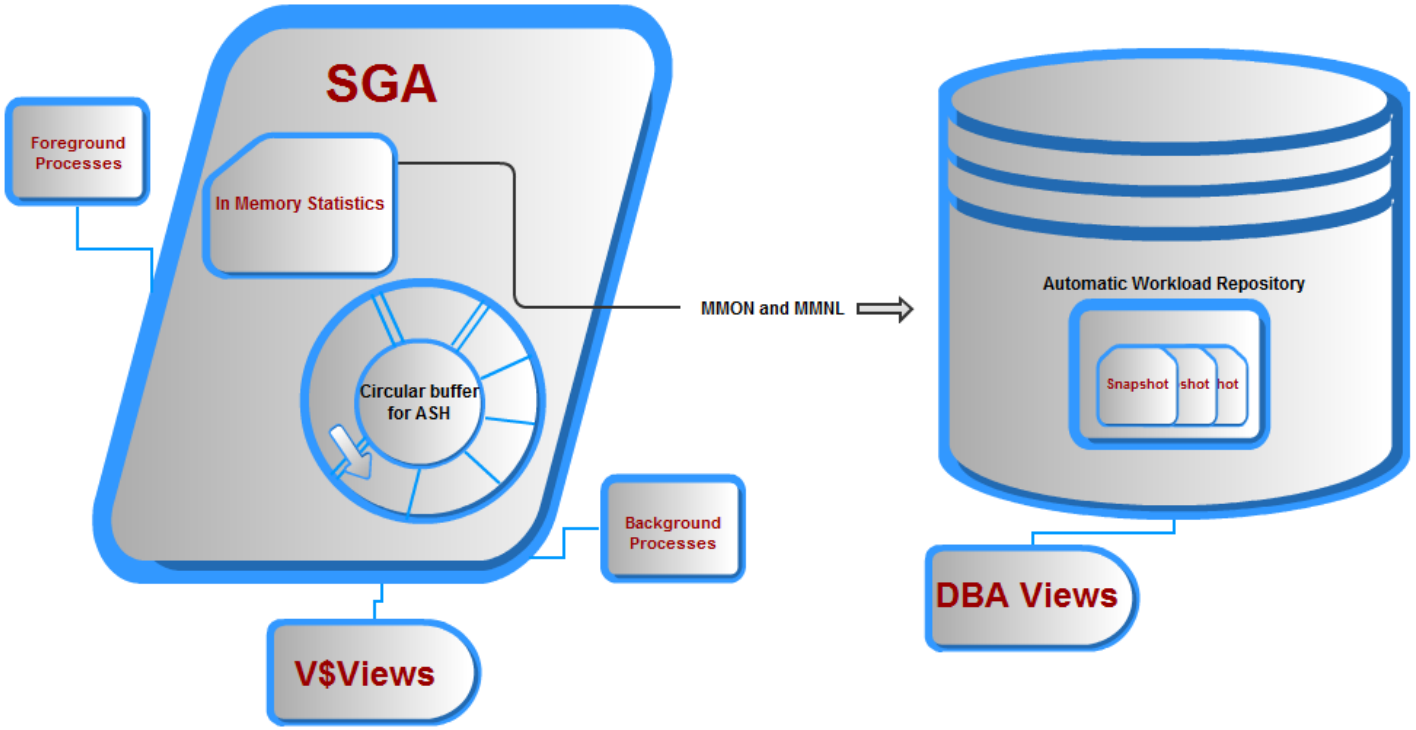
Brief History

- ASH= Active Session History
- AWR= Automatic Workload Repository
- Introduced in Oracle 10g
- Evolution to statspack, requests for performance reporting improvements.
- “Always on” approach to performance metrics with requirement of non-locking collection process.
- Requires Management Tuning Pack License from Oracle.

ASH Architecture



AWR Architecture



AWR Repository

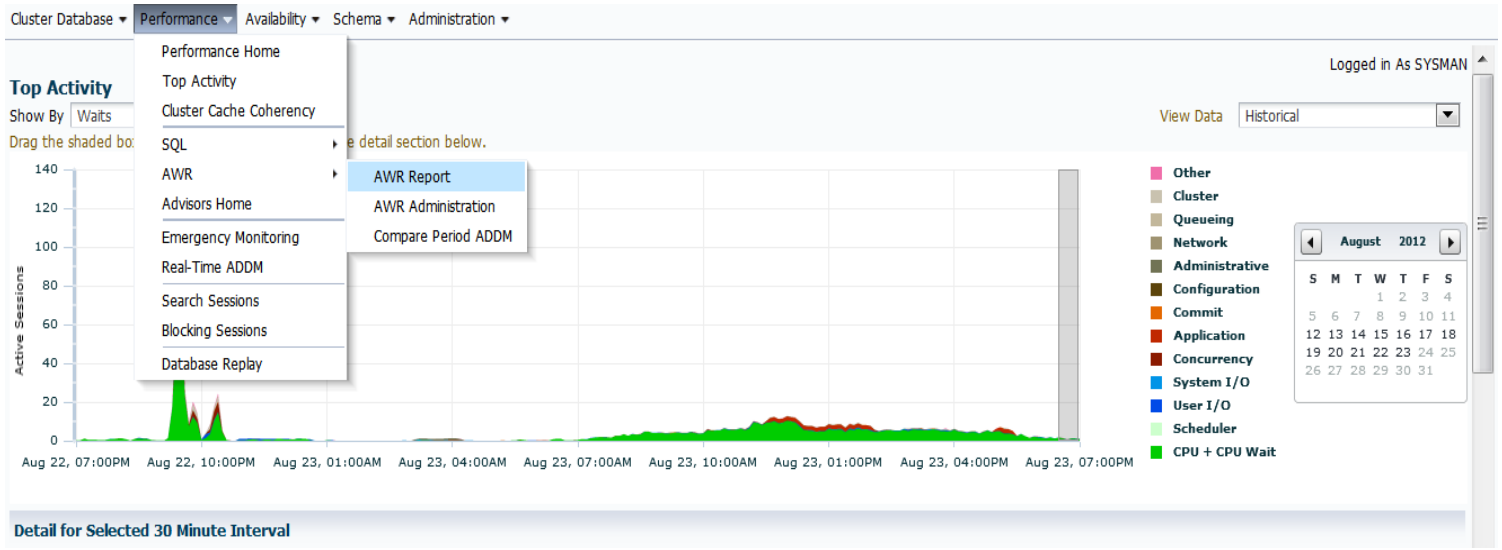
- Used not only by the AWR reports
  - Automatic Database Diagnostic Monitor, (ADDM Reporting)
  - SQL Tuning Advisor
  - Segment Advisor
- By default, snapshots every hour. Retention is for 7 days. Both are modifiable.
- Snapshots can be taken at any time:

```
EXEC
DBMS_WORKLOAD_REPOSITORY.create_snapshot;
```

ASH Data

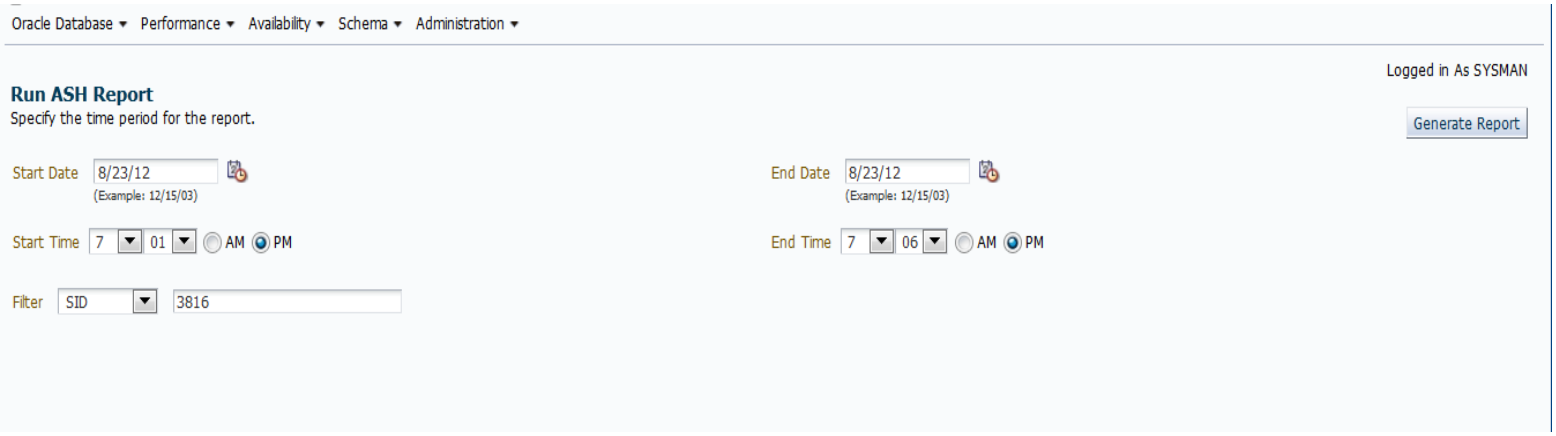
- Samples each active database session every second.
- Data is held in buffer in memory.
- Built into the Oracle kernel and accessed through the v$active_session_history view.
- In an AWR snapshot, 1 row in 10 from ASH buffer is placed into the AWR repository.
- Managed by the MMNL, (Memory Monitor Lite)
- Should not be used to track occurrence.

Running AWR from Enterprise Manager



Running ASH Report from EM

- ASH is always by time, not snapshot.
- Set start date and time.
- End date and time
- Generate report



HTML Format ASH

ASH Report For

(1 Report Target Specified)

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
,	2601412324		2	10.2.0.5.0	YES	om

CPUs	SGA Size	Buffer Cache	Shared Pool	ASH Buffer Size
24	65,536M (100%)	55,280M (84.4%)	10,279M (15.7%)	40.5M (0.1%)

	Sample Time	Data Source
Analysis Begin Time:	28-Aug-12 14:49:15	V$ACTIVE_SESSION_HISTORY
Analysis End Time:	28-Aug-12 14:54:15	V$ACTIVE_SESSION_HISTORY
Elapsed Time:	5.0 (mins)	
Sample Count:	299	
Average Active Sessions:	1.00	
Avg. Active Session per CPU:	0.04	
Report Target:	SQL_ID like 'bv1nfusy2nxkd'	18.9% of total database activity

ASH Report

- Top Events
- Load Profile
- Top SQL
- Top PL/SQL
- Top Sessions
- Top Objects/Files/Latches
- Activity Over Time

Back to Top

Top Events

- Top User Events
- Top Background Events
- Top Event P1/P2/P3 Values

Back to Top

Running Reports, Command Line

$ORACLE_HOME/rdbms/admin/**awrrpt.sql**;
$ORACLE_HOME/rdbms/admin/**ashrpt.sql**;
$ORACLE_HOME/rdbms/admin/**awrsqlrpt.sql;**

Less Known AWR Reports:

awrinfo.sql General AWR Info

awrddrpt.sql Comparison report between snapshots

awrblmig.sql Migrates pre-11g baseline data into 11g Baseline tables.

awrgrpt.sql RAC Aware AWR Report.

AWR Info Report

- Snapshot Interval Information
- Basic Info on Instances and Nodes
- No User or Application Schema info.
- Space Usage by SYSAUX
- WRH$ and Non- AWR Objects, ordered by size
- Snapshot info and if any errors.
- Advisor Tasks

AWR Info Report

```
~~~~~~~~~~~~~~~
AWR INFO Report
~~~~~~~~~~~~~~~

Report generated at 11:48:06 on Aug 21, 2012 ( Tuesday ) in Timezone -05:00

Warning: Non Default AWR Setting!
--------------------------------------------------------------------------------
Snapshot interval is 10 minutes and Retention is 10 days


       DB_ID DB_NAME   HOST_PLATFORM                           INST STARTUP_TIME      LAST_ASH_SID PAR
----------- --------- --------------------------------------- ----- ----------------- ------------ ---
  2601412324 RACX      DB2.racx.com - Linux x86 64-bit             2 21:55:27 (08/08)      26308459 YES
* 2601412324 RACX      DB1.racx.com - Linux x86 64-bit             1 21:55:27 (08/08)      26294975 YES

#######################################################
(I) AWR Snapshots Information
#######################################################

*****************************************************
(1a) SYSAUX usage - Schema breakdown (dba_segments)
*****************************************************
|
| Total SYSAUX size                      2,111.3 MB ( 6% of 32,768.0 MB MAX with AUTOEXTEND ON )
|
| Schema  SYS          occupies          1,940.7 MB (  91.9% )
| Schema  SYSMAN       occupies             48.8 MB (   2.3% )
| Schema  XDB          occupies             48.3 MB (   2.3% )
| Schema  MDSYS        occupies             32.9 MB (   1.6% )
| Schema  OLAPSYS      occupies             15.6 MB (   0.7% )
| Schema  WMSYS        occupies              7.1 MB (   0.3% )
| Schema  SYSTEM       occupies              6.9 MB (   0.3% )
| Schema  CTXSYS       occupies              4.7 MB (   0.2% )
| Schema  EXFSYS       occupies              3.6 MB (   0.2% )
| Schema  DBSNMP       occupies              1.8 MB (   0.1% )
| Schema  ORDSYS       occupies              0.5 MB (   0.0% )
| Schema  DMSYS        occupies              0.3 MB (   0.0% )
| Schema  TSMSYS       occupies              0.3 MB (   0.0% )
|
*****************************************************
```

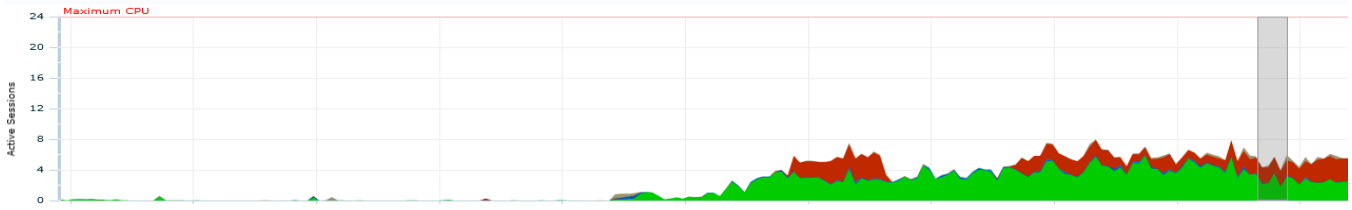
ASH Info Report, (cont.)

```
***********************************
(2) Advisor Task - Oldest 5
***********************************

OWNER/ADVISOR  TASK_ID/NAME                    CREATED          EXE_DURATN EXE_CREATN HOW_C STATUS
-------------- ------------------------------- ---------------- ---------- ---------- ----- ------------
SYS/ADDM       18813/ADDM:2601412324_2_6586    23:00:25 (07/21)          0          0 AUTO  COMPLETED
SYS/ADDM       18822/ADDM:2601412324_1_6586    23:00:25 (07/21)          0          0 AUTO  COMPLETED
SYS/ADDM       18814/ADDM:2601412324_2_6587    00:00:38 (07/22)          0          0 AUTO  COMPLETED
SYS/ADDM       18823/ADDM:2601412324_1_6587    00:00:38 (07/22)          0          0 AUTO  COMPLETED
SYS/ADDM       18815/ADDM:2601412324_2_6588    01:00:04 (07/22)          0          0 AUTO  COMPLETED
SYS/ADDM       18824/ADDM:2601412324_1_6588    01:00:04 (07/22)          0          0 AUTO  COMPLETED

***********************************
(3) Advisor Tasks With Errors - Last 50
***********************************

no rows selected



#######################################################
(III) ASH Usage Info
#######################################################

***********************************
(1a) ASH histogram (past 3 days)
***********************************

NUM_ACTIVE_SESSIONS   NUM_SAMPLES
------------------- ------------
0000 - 0004               12,749
0005 - 0009                2,883
0010 - 0014                   85
0015 - 0019                    3
0020 - 0024                    3

***********************************
(1b) ASH histogram (past 1 day)
***********************************

NUM_ACTIVE_SESSIONS   NUM_SAMPLES
------------------- ------------
0000 - 0004                7,434
0005 - 0009                2,005
```

AWR and ASH in Real Life

Scenario

- RAC, 2-nodes, Ver. 10.2.0.5
- Application Waits Seen in EM Performance Page.
- Out of the ordinary CPU Resource usage
- 50 minutes of time for evaluation. AWR set to 10 min. interval on snapshots.



Run AWR for Timeline Shown in Enterprise Manager

```
              Snap Id      Snap Time      Sessions Curs/Sess
            --------- ------------------- -------- ---------
Begin Snap:     11421 27-Aug-12 12:20:02        79       7.1
  End Snap:     11426 27-Aug-12 13:10:06        88       6.9
   Elapsed:               50.06 (mins)
   DB Time:              293.78 (mins)
```

Who needs a top five when the top 2 are so impacting?

```
Top 5 Timed Events                                         Avg %Total
~~~~~~~~~~~~~~~~~~                                        wait   Call
Event                                   Waits    Time (s)   (ms)   Time Wait Class
------------------------------ ------------ ----------- ------ ------ ----------
CPU time                                          12,197          69.2
enq: TX - row lock contention        10,392       5,073    488   28.8 Applicatio
db file sequential read              47,466         147      3    0.8   User I/O
```

Top SQL by Elapsed Time

```
     5,069          1          951        5.3    28.8 4z1vnc0995bm6
Module: APEX:APPLICATION 10
UPDATE                          SET                     = NULL WHERE    = :B2 AND
                  = :B1

     3,101      3,102        4,820        0.6    17.6 ggqz1puqrjgbt
Module: httpd.worker@              (TNS V1-V3)




     2,931      3,000            0        N/A    16.6 b6vaxgxt4wh8v
Module: APEX:APPLICATION 10
select
```

AWR Segment Info

```
Segments by Logical Reads            DB/Inst:          1  Snaps: 11421-11426
-> Total Logical Reads:   1,524,727,563
-> Captured Segments account for   95.1% of Total

           Tablespace                      Subobject  Obj.       Logical
Owner         Name    Object Name            Name     Type         Reads  %Total
---------- ---------- -------------------- ---------- ----- ------------ -------
W               T     F                               TABLE  639,319,136   41.93
W               T     I           S                   TABLE  297,136,720   19.49
W               T     C_          _PK                 INDEX   96,400,496    6.32
```

```
Segments by Physical Reads           DB/Inst:          1  Snaps: 11421-11426
-> Total Physical Reads:         90,289
-> Captured Segments account for    16.9% of Total

           Tablespace                      Subobject  Obj.      Physical
Owner         Name    Object Name            Name     Type         Reads  %Total
---------- ---------- -------------------- ---------- ----- ------------ -------
W               LABEL SYS                             LOB          7,367    8.16
W               DATA  SYS                             LOB          4,568    5.06
W               DATA  SYS                             LOB            714     .79
```

```
Segments by Row Lock Waits           DB/Inst:          1  Snaps: 11421-11426
-> % of Capture shows % of row lock waits for each top segment compared
-> with total row lock waits for all segments captured by the Snapshot

                                                                    Row
           Tablespace                      Subobject  Obj.         Lock    % of
Owner         Name    Object Name            Name     Type        Waits Capture
---------- ---------- -------------------- ---------- ----- ------------ -------
A                 ?                     PK            INDEX          186   37.96
A                 ?                     N_            INDEX           92   18.78
A                 ?                     NS            INDEX           45    9.18
```

Next Steps Using ASH

- Limiting from a 50 minute/1 hour view to more definitive view of the database a given timeline.
- Top SQL
- Top Sessions
- Top Waits
- Blocking Sessions
- Top Objects
- Waits by time during sample intervals.

Top Modules and Clients

Service	Module	% Activity	Action	% Action
	APEX:APPLICATION 10	90.54	PAGE 60	46.35
			PAGE 27	10.58
			PAGE 0	8.33
	UNNAMED	6.55	UNNAMED	6.28
SYS$BACKGROUND	UNNAMED	1.48	UNNAMED	1.48
	APEX:APPLICATION 7	1.20	PAGE 0	0.78

Client ID		% Activity	Avg Active Sessions
User	Program	Service	
CMC :567659981851685		30.96	1.75
APEX_PUBLIC_USE	httpd.worker@	(TNS V	
HM :2287297672616212		15.37	0.87
APEX_PUBLIC_USE	httpd.worker@	(TNS V	
K 2198237089083354		1.77	0.10
APEX_PUBLIC_USE	httpd.worker@	(TNS V	
D 5048327914813888		1.38	0.08
APEX_PUBLIC_USE	httpd.worker@	(TNS V	

Top SQL Statements

```
                                          Distinct            Avg Active
SQL Command Type                            SQLIDs % Activity   Sessions
---------------------------------------- ---------- ---------- ----------
SELECT                                          918      60.55       3.42
UPDATE                                           24      29.90       1.69
PL/SQL EXECUTE                                   74       5.20       0.29
INSERT                                           23       1.47       0.08
          ---------------------------------------------------------------

Top SQL Statements               DB/Inst:             (Aug 27 12:20 to 12:55)

       SQL ID    Planhash % Activity Event                            % Event
------------- ---------- ---------- ------------------------------ ----------
4z1vnc0995bm6  2494116169      28.45 enq: TX - row lock contention       28.45
UPDATE                        SET                     = NULL WHERE    = :B2 AND
                  = :B1

b6vaxgxt4wh8v  1583487037      17.69 CPU + Wait for CPU                  17.69
select                                          "
```

Blocking Sessions

```
   Blocking Sid % Activity Event Caused                          % Event
--------------- ---------- ------------------------------ ----------
User                  Program                            # Samples Active     XIDs
-------------------- ------------------------------ ------------------ --------
     3884,22252       13.11 enq: TX - row lock contention       13.11
APEX_PUBLIC_USER      httpd.worker@a...m (TNS V1-V3) 2,096/2,100 [100%]        1
```

The Blocking Session, along with Top Object should be noted.

```
      Object ID % Activity Event                                % Event
--------------- ---------- ------------------------------ ----------
Object Name (Type)                                       Tablespace
-------------------------------------------------------- --------------------------
          57604       28.47 enq: TX - row lock contention       28.45
APEX_040000.WWV_FLOW_WORKSHEET_RPTS (TABLE)              APEX_APP
```

Next Step

- Kill Blocking Session?
- Investigate Further?

- Investigate SQL_ID's with AWR SQL

Two SQL_ID's are in question:

4z1vnc0995bm6

b6vaxgxt4wh8v

4z1vnc0995bm6- Update

```
SQL Summary                              DB/Inst:            Snaps: 11421-11425

                Elapsed
   SQL Id      Time (ms)
------------- ----------
4z1vnc0995bm6  3,890,675
Module: APEX:APPLICATION 10
 PAGE 47
UPDATE                          SET                    = NULL WHERE    = :B2 AND
                  = :B1

          ---------------------------------------------------------------

SQL ID: 4z1vnc0995bm6                    DB/Inst:            Snaps: 11421-11425
-> 1st Capture and Last Capture Snap IDs
   refer to Snapshot IDs witin the snapshot range
-> UPDATE     _    _         _                    _   _                    ..

    Plan Hash           Total Elapsed                 1st Capture   Last Capture
#   Value                    Time(ms)    Executions       Snap ID        Snap ID
--- --------------- ---------------- ------------- ------------- --------------
1   2494116169              3,890,675           733         11422          11425
```

Update Cont.

```
Stat Name                                  Statement   Per Execution % Snap
---------------------------------------- ---------- -------------- -------
Elapsed Time (ms)                           3,890,675        5,307.9    28.8
CPU Time (ms)                                     466            0.6     0.0
Executions                                        733            N/A     N/A
Buffer Gets                                     5,959            8.1     0.0
Disk Reads                                          0            0.0     0.0
Parse Calls                                       735            1.0     0.0
Rows                                              733            1.0     N/A
User I/O Wait Time (ms)                             0            N/A     N/A
Cluster Wait Time (ms)                            360            N/A     N/A
Application Wait Time (ms)                  3,889,825            N/A     N/A
Concurrency Wait Time (ms)                          0            N/A     N/A
Invalidations                                       0            N/A     N/A
Version Count                                      20            N/A     N/A
Sharable Mem(KB)                                  331            N/A     N/A
          --------------------------------------------------------------

Execution Plan
----------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                        | Rows  | Bytes | Cost (%CPU)| Time     |
----------------------------------------------------------------------------------------------------------
|   0 | UPDATE STATEMENT             |                             |       |       |     2 (100)|          |
|   1 |  UPDATE                      |                        TS   |       |       |            |          |
|   2 |   TABLE ACCESS BY INDEX ROWID|                        TS   |     1 |   132 |     2   (0)| 00:00:01 |
|   3 |    INDEX UNIQUE SCAN         |                        TS_PK |     1 |       |     1   (0)| 00:00:01 |
----------------------------------------------------------------------------------------------------------
```

B6vaxgxt4wh8v- Select

```
SQL Summary                              DB/Inst:             Snaps: 11421-11428

                Elapsed
   SQL Id      Time (ms)
------------- ----------
b6vaxgxt4wh8v  4,103,452
Module: APEX:APPLICATION 10
 PAGE 60
select




          ------------------------------------------------------------

SQL ID: b6vaxgxt4wh8v                    DB/Inst:             Snaps: 11421-11428
-> 1st Capture and Last Capture Snap IDs
   refer to Snapshot IDs witin the snapshot range
-> select                                            '...

    Plan Hash           Total Elapsed                 1st Capture   Last Capture
#   Value                    Time(ms)    Executions       Snap ID        Snap ID
--- ---------------- ---------------- ------------- ------------- --------------
1   1583487037             4,103,452             0         11422          11428
          ------------------------------------------------------------
```

Select Cont.

```
Execution Plan
----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                  | Name                      | Rows  | Bytes |TempSpc| Cost (%CPU)|
----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                           |                           |       |       |       | 26946 (100)|
|   1 |  SORT AGGREGATE                            |                           |     1 |    14 |       |            |
|   2 |   TABLE ACCESS BY INDEX ROWID              |             L             |     1 |    14 |       |     6   (0)|
|   3 |    INDEX RANGE SCAN                        |             L_IX2         |     2 |       |       |     3   (0)|
|   4 |  SORT AGGREGATE                            |                           |     1 |    14 |       |            |
|   5 |   TABLE ACCESS BY INDEX ROWID              |             L             |     1 |    14 |       |     6   (0)|
|   6 |    INDEX RANGE SCAN                        |             L_IX2         |     2 |       |       |     3   (0)|
|   7 |  SORT AGGREGATE                            |                           |     1 |    14 |       |            |
|   8 |   TABLE ACCESS BY INDEX ROWID              |             L             |     1 |    14 |       |     6   (0)|
|   9 |    INDEX RANGE SCAN                        |             L_IX2         |     2 |       |       |     3   (0)|
|  10 |  COUNT STOPKEY                             |                           |       |       |       |            |
|  11 |   NESTED LOOPS                             |                           |     1 |    43 |       |   200   (3)|
|  12 |    TABLE ACCESS FULL                       |      CE_(                 |     1 |    13 |       |   198   (3)|
|  13 |    TABLE ACCESS BY INDEX ROWID             |                           |     1 |    30 |       |     2   (0)|
|  14 |     INDEX UNIQUE SCAN                      |      _PK                  |     1 |       |       |     1   (0)|
|  15 |  WINDOW SORT                               |                           |  110K|    38M|       | 26946   (3)|
|  16 |   COUNT STOPKEY                            |                           |       |       |       |            |
|  17 |    VIEW                                    |                           |  110K|    38M|       | 26946   (3)|
|  18 |     CONCATENATION                          |                           |       |       |       |            |
|  19 |      FILTER                                |                           |       |       |       |            |
|  20 |       HASH JOIN RIGHT OUTER                |                           |  110K|    19M|  6616K| 26852   (3)|
|  21 |        TABLE ACCESS FULL                   |         S                 |  282K|  3307K|       |  1410   (3)|
|  22 |        HASH JOIN                           |                           |  110K|    17M|       | 24157   (3)|
|  23 |         HASH JOIN                          |                           | 3882 |   401K|       |  3255   (1)|
|  24 |          TABLE ACCESS BY INDEX ROWID       |         ERS               | 1891 | 81313 |       |  1449   (1)|
|  25 |           NESTED LOOPS                     |                           | 3782 |   350K|       |  2908   (1)|
|  26 |            VIEW                            |                           |    2 |   104 |       |    10  (20)|
|  27 |             SORT UNIQUE                    |                           |    2 |    63 |       |    10  (60)|
|  28 |              UNION-ALL                     |                           |       |       |       |            |
|  29 |               TABLE ACCESS BY INDEX ROWID  |               S           |    1 |    26 |       |     4   (0)|
|  30 |                INDEX RANGE SCAN            |               S_CON       |    5 |       |       |     2   (0)|
|  31 |               FILTER                      |                           |       |       |       |            |
|  32 |                NESTED LOOPS                |                           |    1 |    37 |       |     4   (0)|
|  33 |                 TABLE ACCESS FULL          |                           |    1 |    19 |       |     3   (0)|
|  34 |                 TABLE ACCESS BY INDEX ROWID|                       S   |    1 |    18 |       |     1   (0)|
|  35 |                  INDEX UNIQUE SCAN         |                           |    1 |       |       |     0   (0)|
|  36 |            INDEX RANGE SCAN                |                           | 1891 |       |       |     8   (0)|
|  37 |          INDEX FAST FULL SCAN              |                       TNO |  500K|  5379K|       |   340   (3)|
|  38 |         TABLE ACCESS FULL                  |                       TER | 7997K|   495M|       | 20808   (3)|
|  39 |      FILTER                                |                           |       |       |       |            |
|  40 |       NESTED LOOPS OUTER                   |                           |    1 |   183 |       |    94   (3)|
```

Long Story Short….

- Subsequent ASH reports showed ***blocked sessions*** became ***blocking sessions***.
- Update statement and select belong to same code. Update is executed, then large select, no commit until **AFTER** select is complete.
- Request to development to commit before select and tuning recommendation from AWR SQL_ID specific report for select statement.

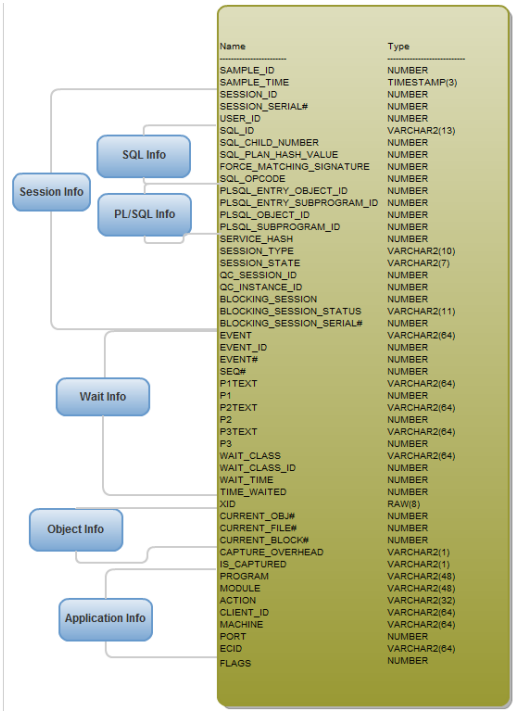
Querying ASH Data Directly

- More defined reporting
- No need to pull full report
- Detail on waits that are of interest
- Join to non-AWR objects

- Simple queries presented…

V$ACTIVE_SESSION_HISTORY

- More Column Data in 11g than shown.
- Flags column is for future development.
- Broken down into usable sections, easier to query.



Knowing What's in the ASH Buffer

- Deters from making assumptions on what data is being queried.
- Know your samples!

```
  1  SELECT MIN(SAMPLE_TIME) AS "Earliest Sample",
  2  MAX(SAMPLE_TIME) as "Most Recent Sample"
  3* FROM  v$active_session_history
SQL> /

Earliest Sample
---------------------------------------------------------------------------
Most Recent Sample
---------------------------------------------------------------------------
29-AUG-12 06.32.50.265 AM
30-AUG-12 12.26.21.124 PM
```

Wait Events Across Nodes

```
  1  select * from (
  2  select sql_id,  inst_id,
  3     sum(decode(vash.session_state,'ON CPU',1,0)) as  "ON CPU",
  4     sum(decode(vash.session_state,'WAITING',1,0)) as  "WAITING ON CPU" ,
  5     event , count(distinct(session_id||session_serial#)) as "SESSION COUNT"
  6  from gv$active_session_history vash
  7  where sample_time > sysdate - 5 /( 60*24)
  8  group by event, inst_id, sql_id
  9  order by 4 desc
 10* ) where rownum < 11
SQL> /

SQL_ID           INST_ID     ON CPU WAITING ON CPU EVENT                                                          SESSION COUNT
------------- ---------- ---------- -------------- ---------------------------------------------------------------- -------------
bmvcknyd8j72d          2          0             13 db file sequential read                                                     4
aga2mvmgv3w0w          2          0             12 direct path read temp                                                       1
7rhhv9rm6wfth          2          0              8 TCP Socket (KGAS)                                                           1
5s1nj0c9hb9gj          1          0              7 direct path read temp                                                       1
4xhr6sm3h5116          1          0              6 TCP Socket (KGAS)                                                           1
7rhhv9rm6wfth          1          0              6 TCP Socket (KGAS)                                                           1
                       1          0              5 log file parallel write                                                     1
bmvcknyd8j72d          1          0              4 db file sequential read                                                     1
                       2          0              4 log file parallel write                                                     1
2kdwvvg6r3wjc          1          0              3 TCP Socket (KGAS)                                                           2

10 rows selected.
```

Query top 10 SQL_ID's in the last 10 minutes?

```
  1  SELECT * FROM
  2  (SELECT NVL(SQL_ID,'NULL') AS SQL_ID,
  3  SUM(1) as "DBTime in Seconds"
  4  FROM V$ACTIVE_SESSION_HISTORY
  5  WHERE sample_time > sysdate -10/(24*60)
  6  GROUP BY SQL_ID
  7  ORDER BY 2 DESC)
  8* WHERE ROWNUM < 11
SQL> /

SQL_ID        DBTime in Seconds
------------- -----------------
7rhhv9rm6wfth               183
99rdyn8d1bjzv                86
NULL                         36
0uvr1k5nug5a8                25
cczz51gzm0h65                23
ggqz1puqrjgbt                21
624hxtvjd2msf                20
cabn5q1a2xps9                 9
8fp3ja8qnqx5r                 9
cmwtpm0smcjy3                 8

10 rows selected.
```

SQL_ID and CPU Usage

```
  1  select * from (
  2  select sql_id,  inst_id,
  3         sum(decode(vash.session_state,'ON CPU',1,0))  as "Number on CPU",
  4         sum(decode(vash.session_state,'WAITING',1,0)) as "Number Waiting on CPU"
  5  from  gv$active_session_history vash
  6  where sample_time > sysdate - 5 /( 60*24)
  7  group by sql_id, inst_id
  8  order by 3 desc
  9* ) where rownum < 11
SQL> /

SQL_ID           INST_ID Number on CPU Number Waiting on CPU
------------- ---------- ------------- ---------------------
206f2ak30uqxv          2           300                     0
fq6qaj001cxxv          1           115                     0
gfu7v7dkkdbv8          1            96                     0
gfu7v7dkkdbv8          2            85                     0
5wxfn4hb2mbyn          1            51                     0
cczz51gzm0h65          1            44                     0
5wxfn4hb2mbyn          2            44                     0
cmqr2a0v9uj01          2            43                     0
99rdyn8d1bjzv          1            42                     0
99rdyn8d1bjzv          2            40                     0

10 rows selected.
```

IO Waits by Object from ASH

```
  1  SELECT TW.*, DO.object_name FROM
  2  (SELECT current_obj#,
  3  ROUND(SUM(CASE WHEN time_waited >= 1000000 THEN 1
  4  ELSE 1000000 / time_waited END)) as "Estimated IO Waits",
  5  SUM(1) as "Estimated DBTime"
  6  FROM V$ACTIVE_SESSION_HISTORY
  7  WHERE sample_time > sysdate - 5/(24*60)
  8  AND TIME_WAITED >0
  9  and wait_class ='User I/O'
 10  GROUP BY current_obj#
 11  ORDER BY 2 DESC) TW, DBA_OBJECTS DO
 12  WHERE DO.object_id=TW.current_obj#
 13* and ROWNUM < 11
SQL> /

CURRENT_OBJ# Estimated IO Waits Estimated DBTime OBJECT_NAME
------------ ------------------ ---------------- ----------------------------------------
-----------------------------------------
       66637               4509                9 SYS_LOB0000066636C00006$$
       66645                145                1 SYS_LOB0000066641C00003$$
       67315                145                1 XAK_RATE_TABLE_1
```

SQL Text with ASH

- SQL for most recent five minutes of sample data from ASH

```
SELECT ash.sql_id
,      ash.sql_child_number
,      s.sql_text
,      TO_CHAR(MIN(ash.sample_time),'hh24:mi:ss') AS min_sample_time
,      TO_CHAR(MAX(ash.sample_time),'hh24:mi:ss') AS max_sample_time
FROM   v$active_session_history ash
,      v$sql s
WHERE  ash.sql_id           = s.sql_id (+)
AND    ash.sql_child_number = s.child_number (+)
and    ash.sample_time > SYSDATE - 5/(24*60)
GROUP  BY
       ash.sql_id
,      ash.sql_child_number
,      s.sql_text
ORDER  BY
       MIN(ash.sample_time)
```

SQL Results

- SQL_ID, SQL Text, Sample Time that Process was captured in.

```
cczz51gzm0h65                1

SELECT COUNT(1) FROM                WHERE                IN (SELECT              FROM
WHERE                                                                         OR :B1
=            AND TRUNC                = TRUNC(SYSDATE)
11:15:15 11:15:15

gfu7v7dkkdbv8                3
select


               ount(*) over ()                        ( select  *  from ( SELECT


11:15:15 11:15:15
```

Tyler Muth ASH Mining Query

```
select snap_id "snap",num_interval "dur_m", end_time "end",inst "inst",
 max(decode(metric_name,'Host CPU Utilization (%)',          average,null)) "os_cpu",
 max(decode(metric_name,'Host CPU Utilization (%)',          maxval,null)) "os_cpu_max",
 max(decode(metric_name,'Database Wait Time Ratio',                round(average,1),null)) "db_wait_ratio",
max(decode(metric_name,'Database CPU Time Ratio',                round(average,1),null)) "db_cpu_ratio",
max(decode(metric_name,'SQL Service Response Time',                average,null)) "sql_res_t_cs",
max(decode(metric_name,'Executions Per Sec',                average,null)) "exec_s",
max(decode(metric_name,'Logical Reads Per Sec',                average,null)) "l_reads_s",
max(decode(metric_name,'User Commits Per Sec',                average,null)) "commits_s",
max(decode(metric_name,'Physical Read Total Bytes Per Sec',        round((maxval)/1024/1024,1),null)) "read_mb_s_max"
  from(
  select  snap_id,num_interval,to_char(end_time,'YY/MM/DD HH24:MI') end_time,instance_number inst,metric_name,round(average,1) average,
  round(maxval,1) maxval
 from dba_hist_sysmetric_summary
where
snap_id between &SNAP_ID_MIN and &SNAP_ID_MAX
 and metric_name in ('Host CPU Utilization (%)','Average Active Sessions','Executions Per Sec','Hard Parse Count Per Sec','Logical Reads Per Sec','Logons Per Sec',
 'Physical Read Total Bytes Per Sec','Physical Read Total IO Requests Per Sec','Physical Write Total Bytes Per Sec','Physical Write Total IO Requests Per Sec',
 'Redo Generated Per Sec','User Commits Per Sec','Current Logons Count','DB Block Gets Per Sec','DB Block Changes Per Sec',
 'Database Wait Time Ratio','Database CPU Time Ratio','SQL Service Response Time','Background Time Per Sec'))
 group by snap_id,num_interval, end_time,inst
 order by snap_id, end_time,inst;
```

ASH Mining Output

```
SQL> /
Enter value for snap_id_min: 10757
Enter value for snap_id_max: 10760
old  16: snap_id between &SNAP_ID_MIN and &SNAP_ID_MAX
new  16: snap_id between 10757 and 10760

      snap      dur_m end                 inst     os_cpu os_cpu_max db_wait_ratio db_cpu_ratio sql_res_t_cs     exec_s  l_reads_s  commits_s read_mb_s_max
---------- ---------- -------------- ---------- ---------- ---------- ------------- ------------ ------------ ---------- ---------- ---------- -------------
     10757         10 12/08/22 21:39          1       46.8       95.9          20.3         79.7            0    34080.5   503927.8      157.7            .7
     10757         10 12/08/22 21:39          2       44.3       98.1          22.7         77.3            0      32911   450327.6      138.8           9.1
     10758         10 12/08/22 21:49          1       21.6       39.5          24.2         75.8            0    25696.7   365775.4      141.7           5.8
     10758         10 12/08/22 21:49          2         20       41.6          28.3         71.7            0    23125.1   323426.9      130.3           8.9
     10759         10 12/08/22 21:59          1        9.5       33.1          19.2         81.1            0    10076.8   146136.9       57.7            .2
     10759         10 12/08/22 21:59          2       11.1       27.5          19.8         80.9            0    12348.2   176585.1       68.1            .5
     10760         10 12/08/22 22:09          1        1.3        1.9          16.3         83.8            0       81.7     5535.7         .5            .5
     10760         10 12/08/22 22:09          2        5.6        7.8          61.3         38.7           .2      982.3    47646.4        1.6          19.5

8 rows selected.
```

Additional Options:

- Physical Read Averages
- Physical Writes, (Max/Averages)
- Redo Info
- Logon Info
- Hard Parsing, etc.

Best Practice When Querying ASH Data

- Keep it Simple and don't reinvent the wheel.
- Samples are an alias for time, not for counts.
- Understand what is valuable and compare to packaged reports.
- Be aware on RAC of node specific data.
- Take care when querying Obj#, File# and Block#, (still issues in different versions…)
- Check the time that is available in buffer, don't assume!

AWR/ASH Links/Blogs

- Karl Arao: http://karlarao.wordpress.com
- Tyler Muth: http://tylermuth.wordpress.com/
- Kyle Hailey, John Beresniewicz, Graham Wood: http://ashmasters.com/
- Mine- "For the Love of ASH and AWR" http://dbakevlar.com/2011/02/for-the-love-of-awr-and-ash/

QUESTIONS

Email: dbakevlar@gmail.com
Company: http://enkitec.com
Website: http://dbakevlar.com
User Group: http://www.rmoug.org