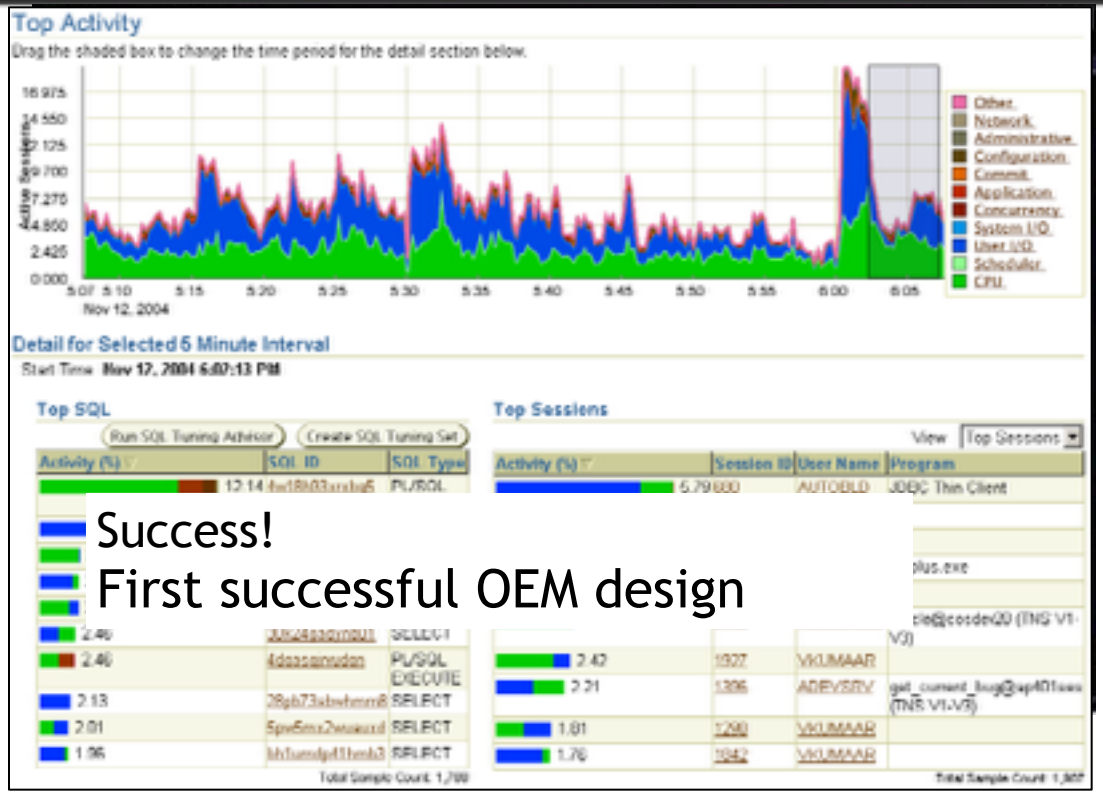
ASH Masters

ASH SQL Query Repository

Kyle Hailey
http://kylehailey.com
Kyle@delphix.com

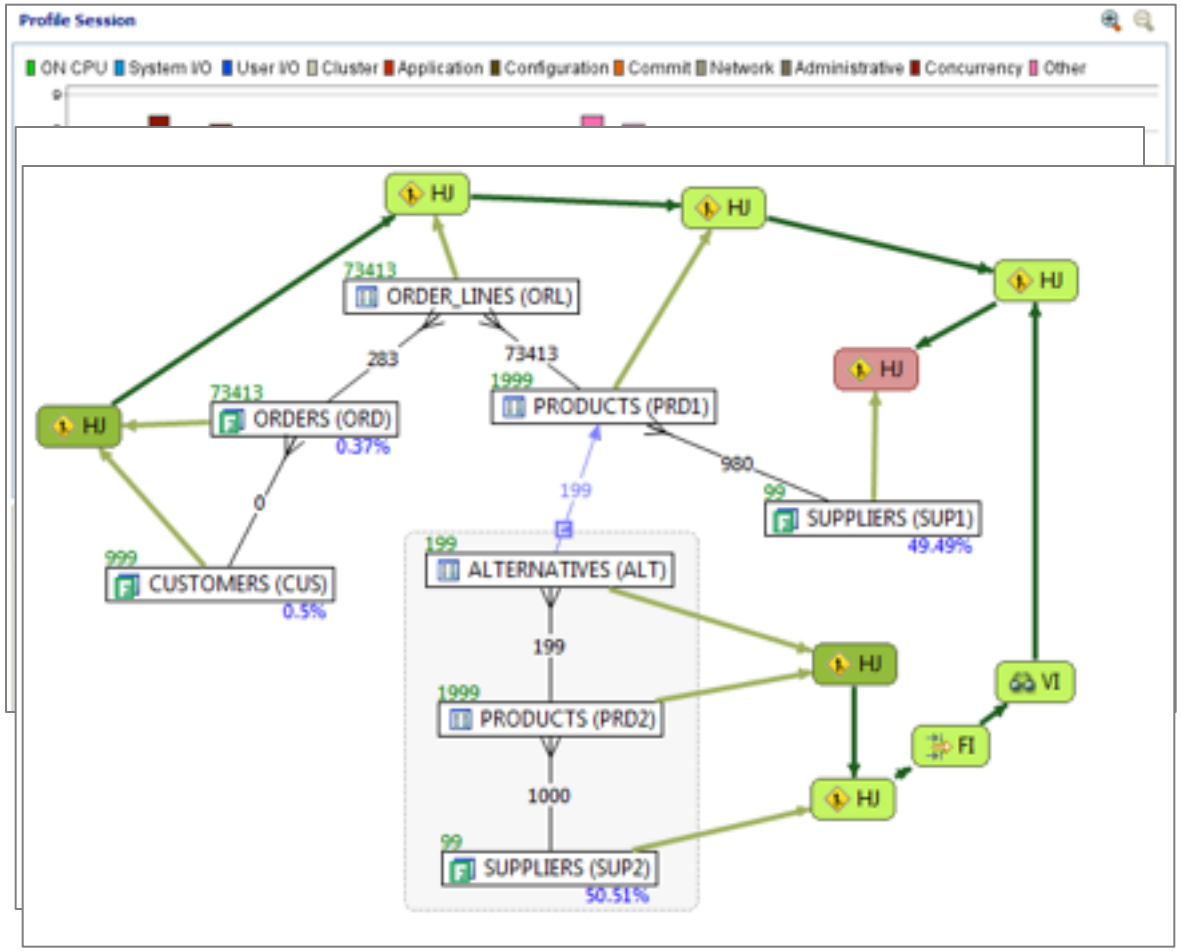
Who is Kyle Hailey

- 1990 Oracle
  - 90 support
  - 92 Ported v6
  - 93 France
  - 95 Benchmarking
  - 98 ST Real World Performance
- 2000 Dot.Com
- 2001 Quest
- 2002 Oracle OEM 10g



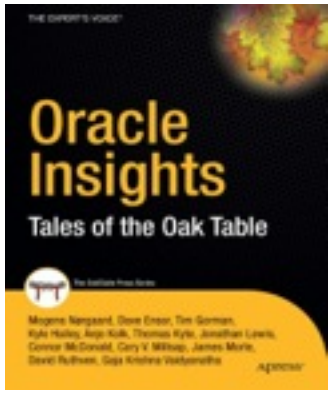
Who is Kyle Hailey

- 1990 Oracle
  - 90 support
  - 92 Ported v6
  - 93 France
  - 95 Benchmarking
  - 98 ST Real World Performance
- 2000 Dot.Com
- 2001 Quest
- 2002 Oracle OEM 10g
- 2005 Embarcadero
  - DB Optimizer



Who is Kyle Hailey

- 1990 Oracle
  - 90 support
  - 92 Ported v6
  - 93 France
  - 95 Benchmarking
  - 98 ST Real World Performance
- 2000 Dot.Com
- 2001 Quest
- 2002 Oracle OEM 10g
- 2005 Embarcadero
  - DB Optimizer
- Delphix



When not being a Geek

- Have a little 4 year old boy who takes up all my time …
  and now a 2 week old !

Typical Architecture

Production

Dev, QA, UAT

Reporting

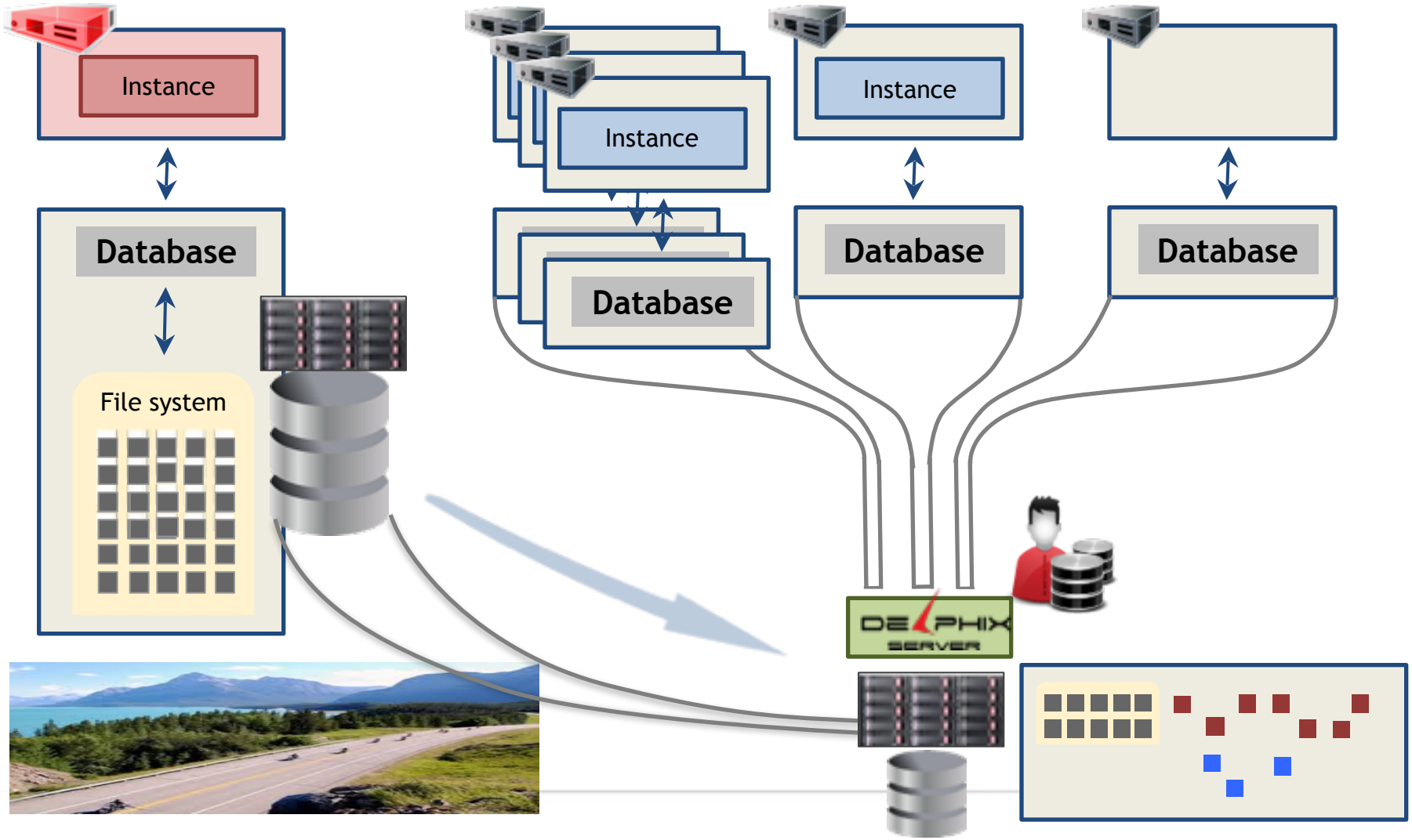
Backup

Instance

Database

File system

With Delphix



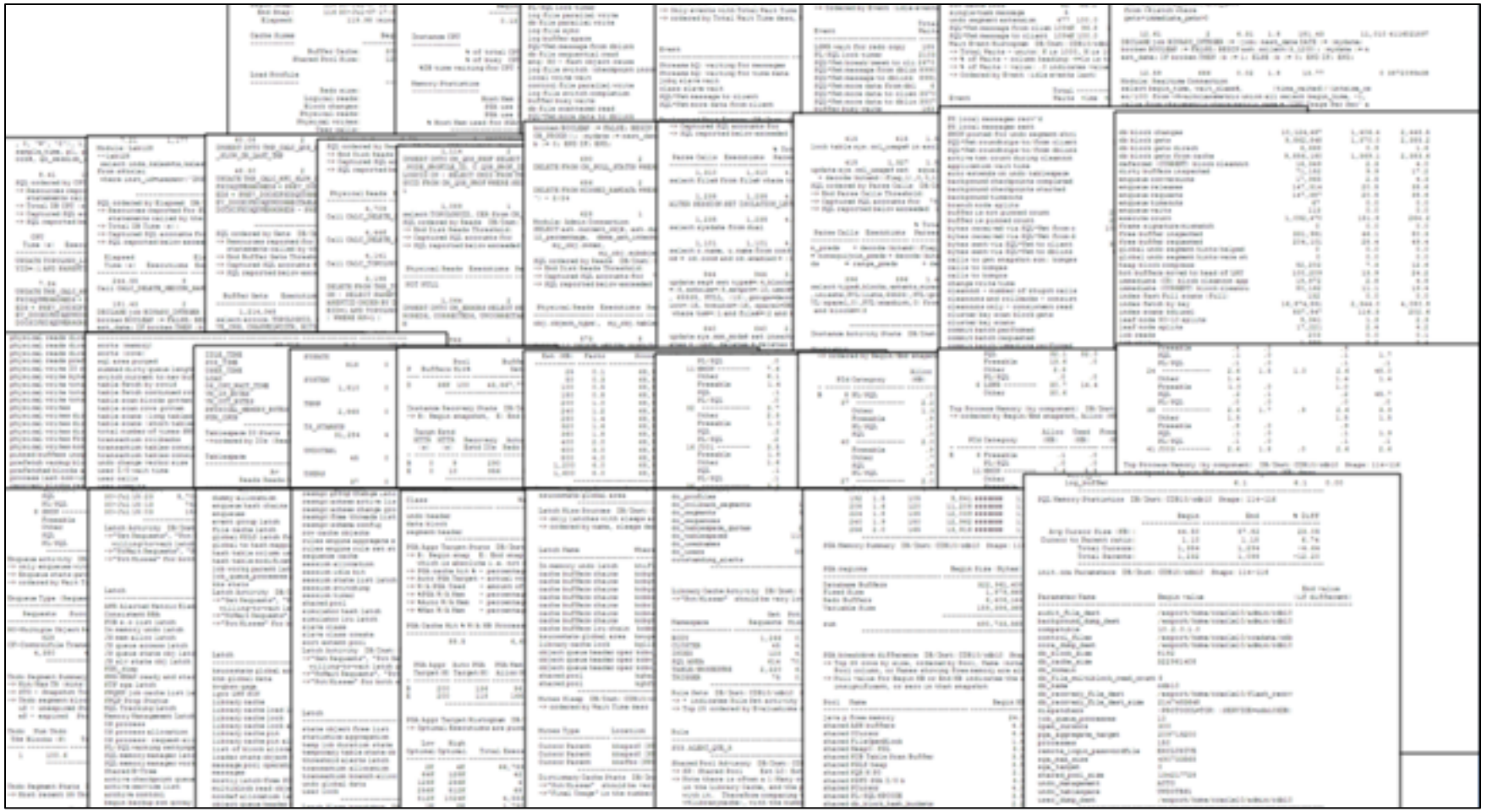
How do you tune a Database?

Database is running slow!

- first step?
- questions?
- tools?
- repeatable method?

DELPHIX

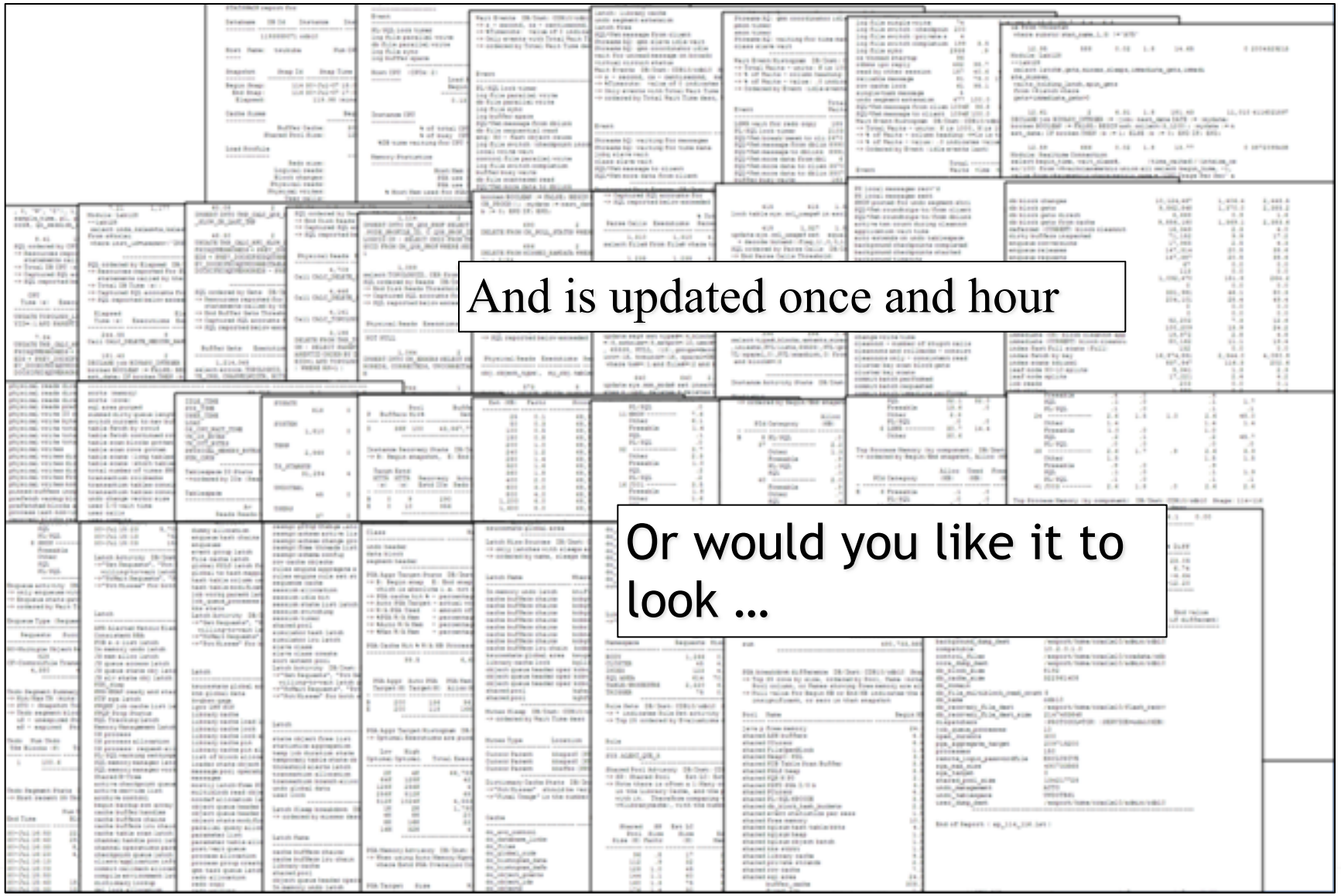
What statistics do you look at?



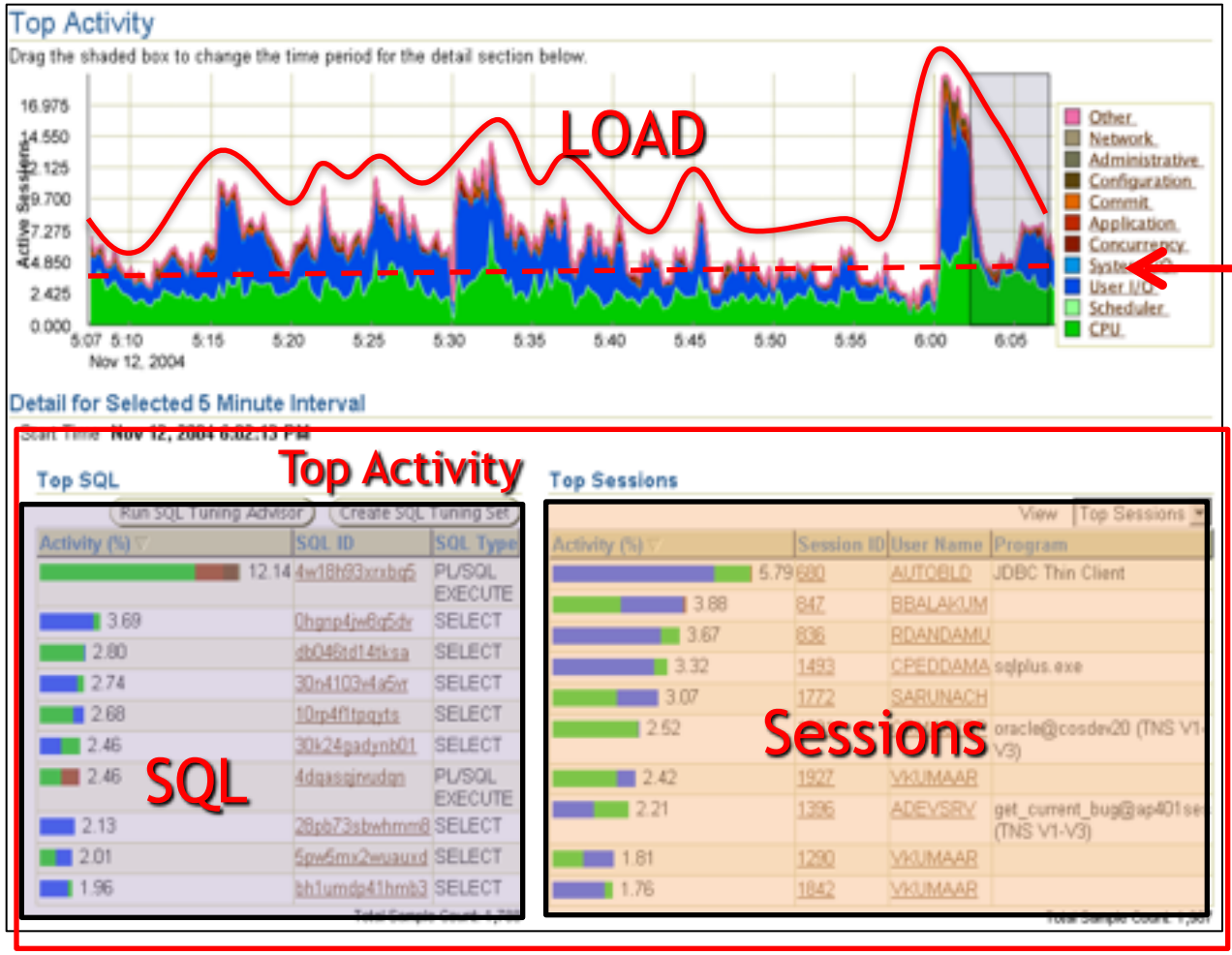
Disastrously difficult

Imagine Trying to Drive your Car

Would you want your dashboard to look like :



How Can We Open the Black Box? ASH

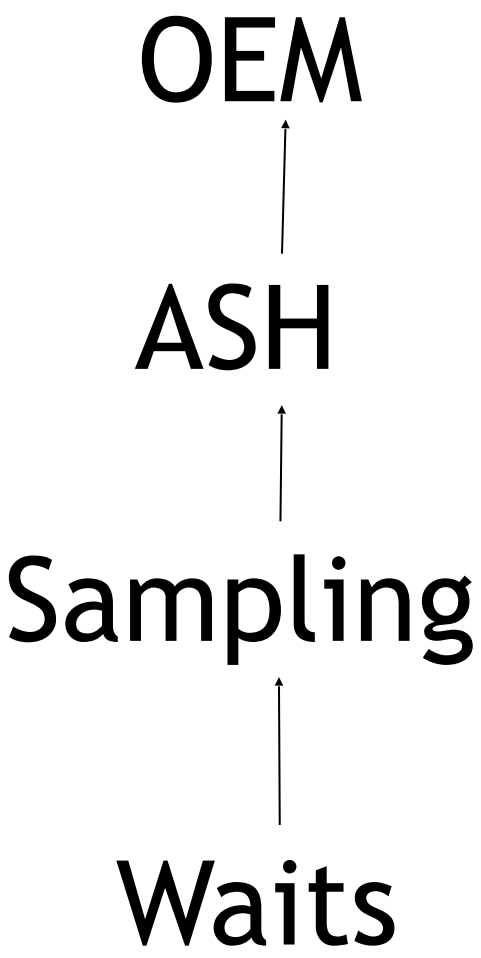


Max CPU
(yard stick)

Don't always have OEM, but always have SQL

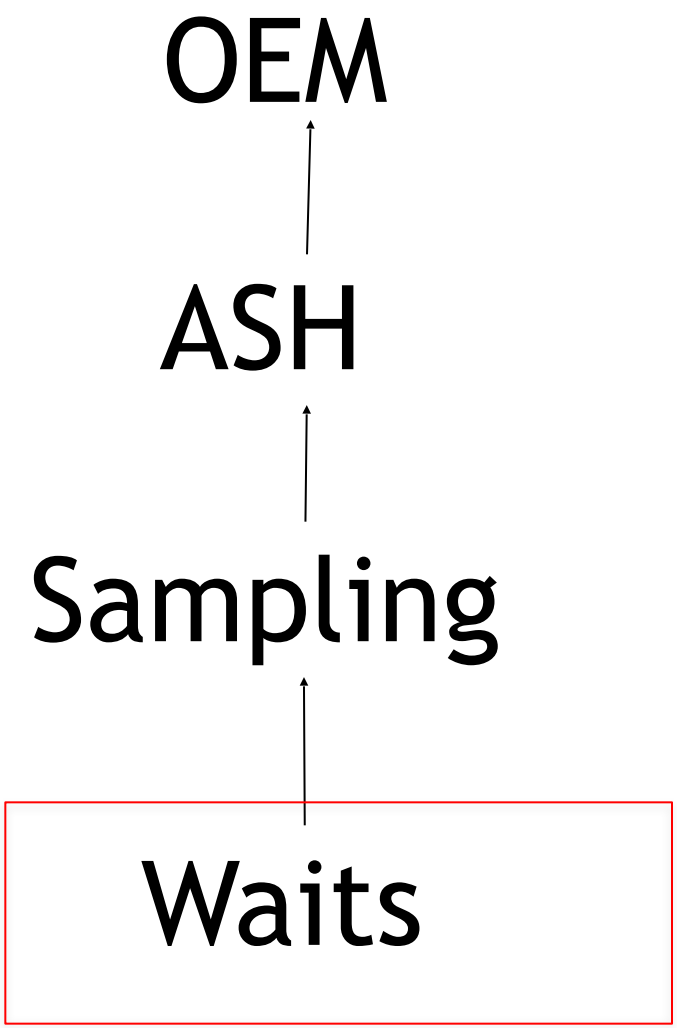
Ideas for Today

OEM doesn't always have the detail



Where does the OEM data come from ?

Ideas for Today



DELPHIX

Statspack: Cheat Sheet

- Install
  - Connect as SYSDBA
  - @?/rdbms/admin/spcreate.sql
- Run
  - Exec statspack.snap;
- Generate Reports
  - @?/rdbms/admin/spreport.sql
  - (AWR : @?/rdbms/admin/awrrpt.sql)

Statspack = always available

DELPHIX

Statspack Sections (10g)

Instance description
Host hardware
Snapshot headline
Cache information
Load profile
Instance efficiency
Shared pool stats
Top timed events
Host CPU load
Host / Instance CPU warning
VM activity
Memory usage
Time model stats
RAC statistics
All wait events
Background wait events
Event histogram
SQL x 9
Instance activity
Log switches
OS stats

Session details x 3
Tablespace I/O
File I/O
File I/O histogram
Buffer pool
Instance Recovery
Buffer pool advisory
Buffer busy waits
Various PGA summaries
PGA histogram
PGA advisory
PGA allocation summary
PGA allocation top N
Enqueue (lock) activity
Undo stats x 2
Latch activity
Latch miss details

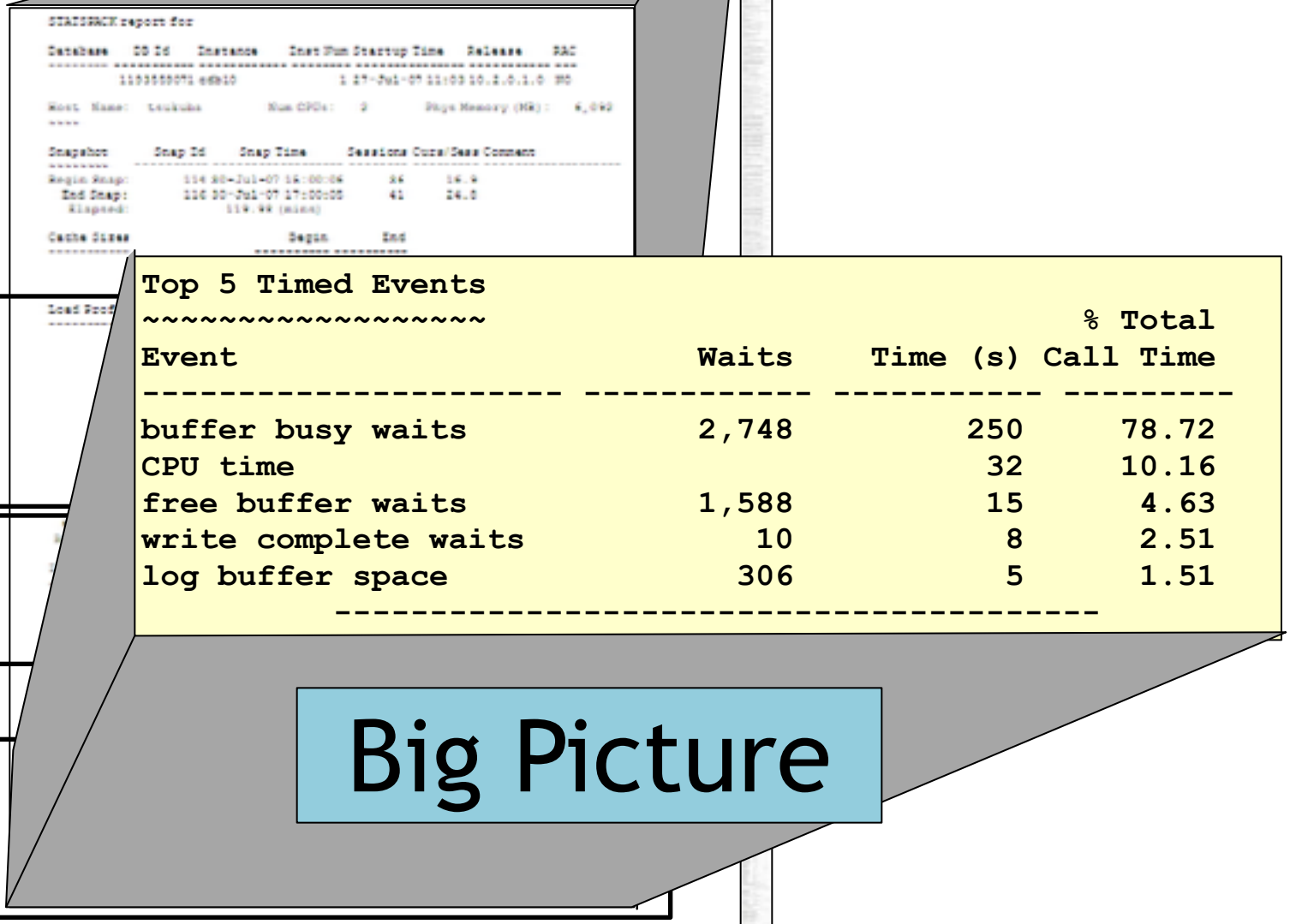
Latch parent and child
Mutex
Segment stats x 8
Dictionary cache
Library cache
RAC (GES)
RAC (CR and CUR served)
RAC (cache xfer x 2)
RAC (Remastering)
Streams x 7
Shared pool advisor
Java pool advisor
SGA resizing
SGA target advisor
SGA summary
SGA detail
SQL memory summary
Resource limits
Parameters

Statspack Method

Who/When

Summary

Waits



Top 5 Timed Events

Event	Waits	Time (s)	% Total Call Time
buffer busy waits	2,748	250	78.72
CPU time		32	10.16
free buffer waits	1,588	15	4.63
write complete waits	10	8	2.51
log buffer space	306	5	1.51

Big Picture

Latch Free

Top 5 Timed Events

Event	Waits	Time (s)	% Total Ela Time
latch free	9,652	760	66.10
CPU time		248	21.62
PL/SQL lock timer	41	123	10.72
SQL*Net message from dblink	681	14	1.22
log file parallel write	128	1	.13

What Latch? There are 100s

Latch Sleep breakdown for DB: CDB Instance: cdb Snaps: 3 -4
-> ordered by misses desc

Latch Name	Requests	Misses	Sleeps	Sleeps 1->4
cache buffers chains	8,448,787	649,484	6,930	0/0/0/0/0
library cache pin	8,405,896	82,915	1,427	81537/1330/47/1/0
library cache	8,435,488	55,645	1,294	54375/1247/22/1/0
shared pool	58,626	7	1	6/1/0/0/0

DELPHIX

Row Locks 10g+

op 5 Timed Events

Event	Waits	Time (s)	Avg wait (ms)	%Total Call Time
enq: TX - row lock contention	59	160	2714	41.8
PL/SQL lock timer	4	117	29291	30.6
CPU time		28		7.2
buffer busy waits	1,217	18	15	4.7
log file parallel write	422	11	27	3.0

- Who is waiting
- Who is blocking
- What is the SQL
- What is the row?

Buffer Busy Wait

Top 5 Timed Events

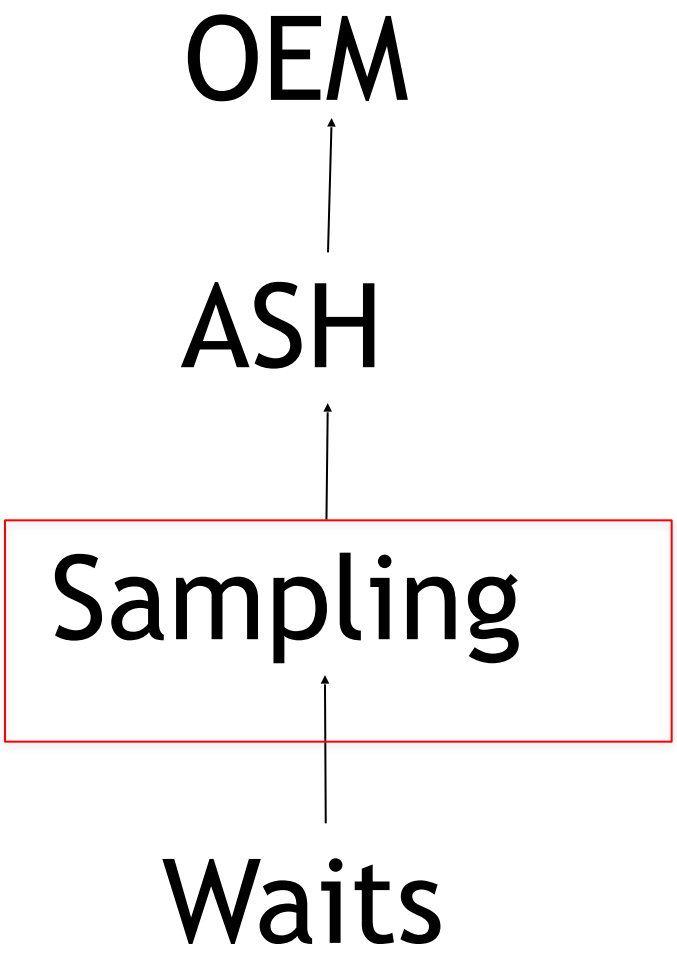
Event	Waits	Time (s)	% Total Call Time
buffer busy waits	2,748	250	78.72
CPU time		32	10.16
free buffer waits	1,588	15	4.63
write complete waits	10	8	2.51
log buffer space	306	5	1.51

Buffer Busy Wait – trying to modify a block

Who blocks?
What object?
what is the SQL?

Statspack fails for analysis

Ideas for Today



Example

- Look at Statspack:

```
Top 5 Timed Events
~~~~~~~~~~~~~~~~~~                                        % Total
Event                         Waits     Time (s) Call Time
--------------------------- ------------ ---------
buffer busy waits            2,748           250      78.72
CPU time                                      32      10.16
free buffer waits            1,588            15       4.63
write complete waits            10             8       2.51
log buffer space               306             5       1.51
-------------------------------
```

What do we do?

- How do we solve “buffer busy wait”
- Need:
  - SQL
  - Sessions
  - Objects
- How do we get that information?
  - Not from Statspack or AWR
  - But from v$session
    - v$session_wait prior to 10g

V$session

- Shows who is waiting
  - For example on “buffer busy waits”
  - Data only exists while waits are happening
  - Data includes
    - Sql
    - Session
    - Object
    - Type of buffer busy wait
    - File and block involved in buffer busy wait
- Problem: Once waits are over, data is gone
- Solution: Sample data all the time

Sample Query

```
select
   nvl(s.username,s.program) username,
   s.sid sid,
   s.serial# serial,
   s.sql_hash_value sql_hash_value,
   substr(decode(w.wait_time,
         0, w.event,
        'ON CPU'),1,15) event ,
   w.p1  p1,
   w.p2  p2,
   w.p3  p3
from        v$session          s,
            v$session_wait     w
where     w.sid=s.sid
and s.status='ACTIVE'
and s.type='USER';
```

SQL identifier

Waiting or on CPU?

This query works since v7

Sampling Output

USERNAME	SID	SERIAL	SQL_HASH_V	EVENT	P1	P2	P3
SYS	64	8717	4116021597	PL/SQL lock timer	300	0	0
SYS	58	19467	961168820	ON CPU	16508152	1	0
STARGUS	71	6251	1311875676	direct path write	201	2155902	127
(CJQ0)	9	1	0	rdbms ipc message	500	0	0

Run sample query every second and save into a table “v$ash”

```
Create table v$ash as select … ;
Insert into v$ash select  … ;
```

Buffer busy wait type

Buffer Busy Wait
P1 = file #
P2 = block #
P3 = block type

v$event_name
Or documentation

```
SQL> Select count(*), p3
     from v$ash
     where event = 'buffer busy waits'
     group by p3;

COUNT(*)         P3
---------- ----
3423              1
```

Buffer busy wait type

Block types come from

```
select rownum n,
       class
  from v$waitstat;
```

```
 N CLASS
-- ------------------
 1 data block
 2 sort block
 3 save undo block
 4 segment header
 5 save undo header
 6 free list
 7 extent map
 8 1st level bmb
 9 2nd level bmb
10 3rd level bmb
11 bitmap block
12 bitmap index block
13 file header block
14 unused
15 system undo header
16 system undo block
17 undo header
18 undo block
```

File and Block #s

```
select count(*),  p1 filen, p2 blockn
from v$ash
where event='buffer busy waits'
group by p1, p2, hash_value;

  COUNT(*)     FILEN   BLOCKN
---------- -------- --------
         1   11         90644
         2   11         90651
         3   11         98233
         1   11        104767
         3   11        113291
         1   11        119842
         1   11        119856
         3   11        121632
         1   11        126334
```

Pick a File and Block to find object

Find Object

```
  COUNT(*)     FILEN    BLOCKN
---------- --------- ---------
         1        11    126334
```

```
column segment_name format a30

select
        owner,
        segment_name,
        segment_type,
        block_id, blocks+block_id
from  dba_extents
where file_id = 11
and 126334 between block_id AND block_id + blocks-1;

OWNER      SEGMENT_NAME       SEGMENT_TY   BLOCK_ID BLOCKS+BLOCK_ID
---------- ------------------ ---------- ---------- ---------------
SYSTEM     TOTO1               TABLE         125201          127249
```

What SQL ?

```
SQL> select count(*), sql_hash_value
2  from v$ash
3  where event='buffer busy waits'
4  group by  hash_value;


COUNT(*)     SQL_HASH_VALUE
----------  --------------
      3423       558666863
```

All the same SQL SQL_HASH_VALUE= 558666863

In version 10g+ use SQL_ID

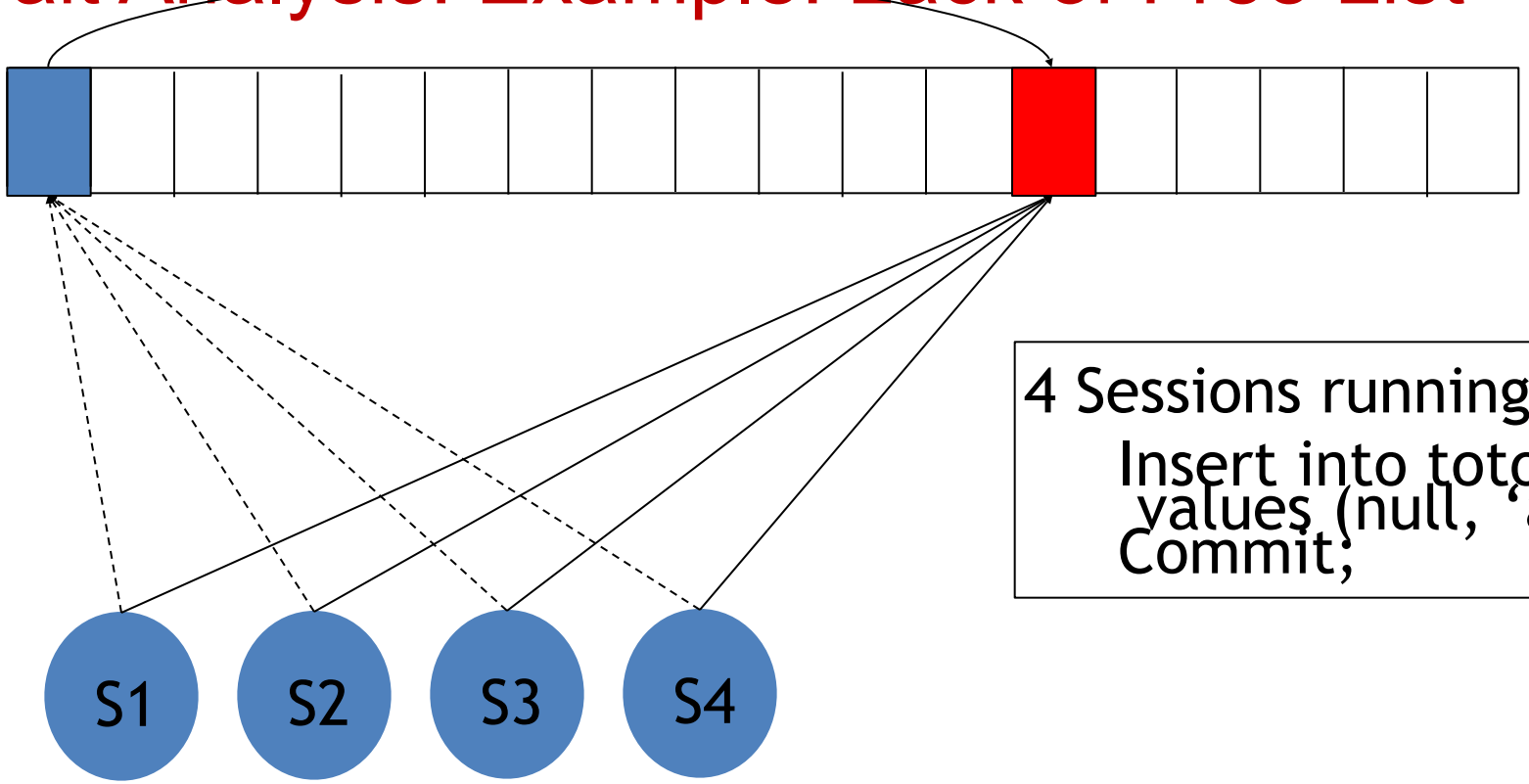
SQL Statement ?

```
select sql_text
from v$sqltext
where hash_value=558666863;

SQL_TEXT
--------------------------------------------------
INSERT into toto1 values (:b1,lpad('a',1000,'a'))
```

- Insert statement
- Problem on a data block on
- table toto1

Solution: Freelists or ASSM

3. Wait Analysis: Example: Lack of Free List



4 Sessions running

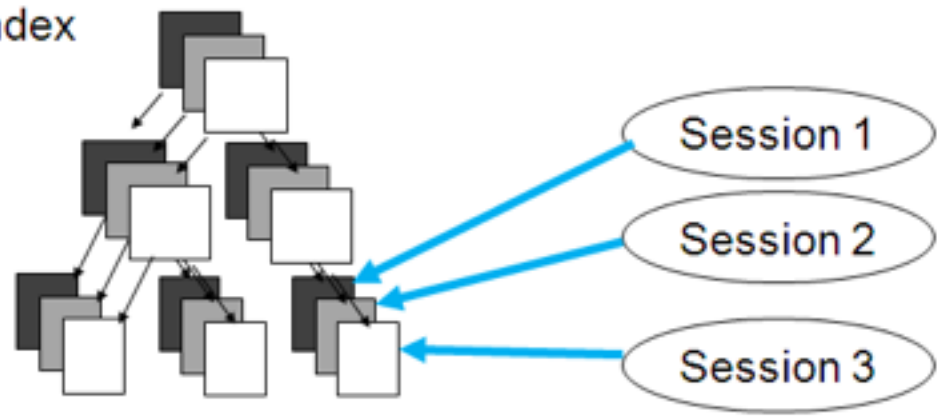
```
Insert into toto1
 values (null, 'a');
Commit;
```

OBJN	OTYPE	FILEN	BLOCKN	SQL_ID	BLOCK_TYPE
54962	TOTO1 TABLE	16	45012	8gz51m9hg5yuf	data block
54962	TOTO1 TABLE	16	161	8gz51m9hg5yuf	segment header

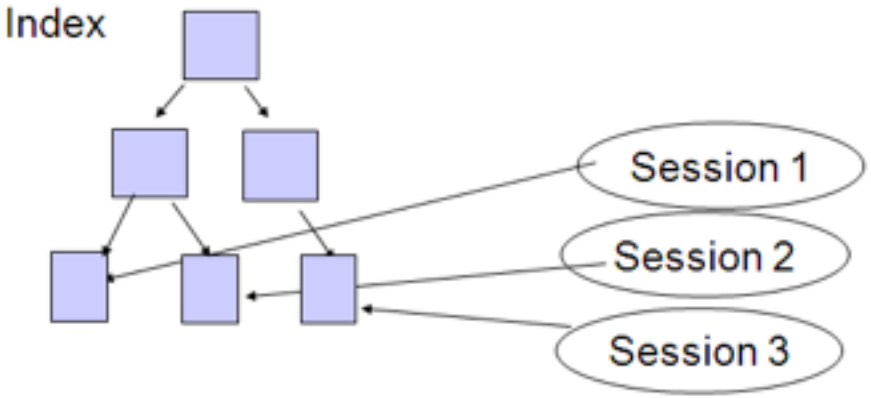
3. Wait Analysis: BBW on Index

Solutions

1. Hash Partitions



2. Reverse Keys

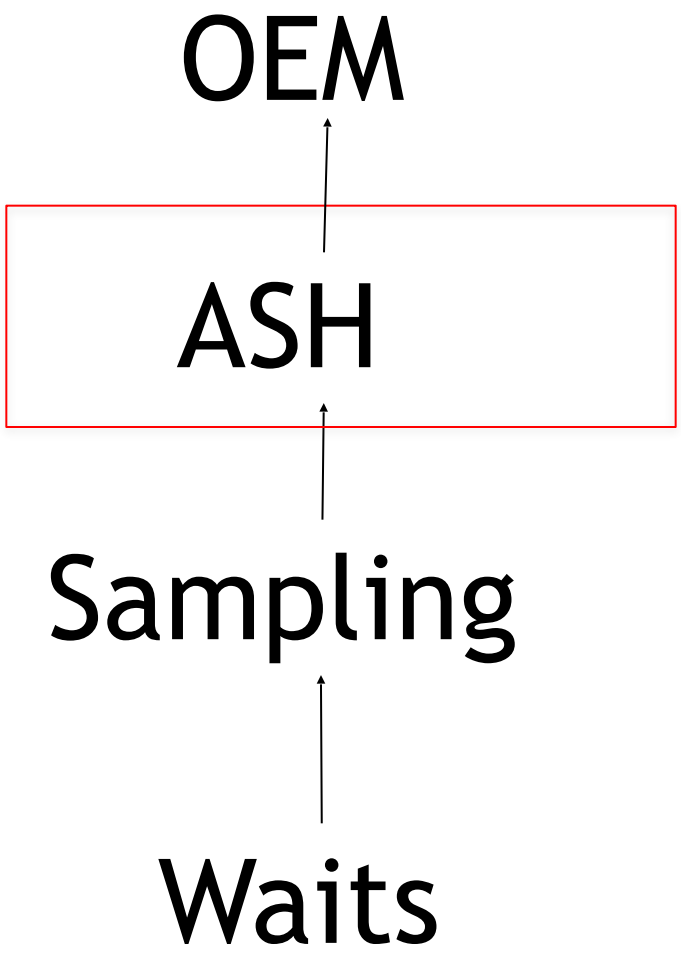


Instead of inserting	We insert the reverse
101	101
102	201
103	301

Sampling Summary

- Sampling v$session
  - When a problem, shows the root causes
  - Active Session History (ASH)
  - v$active_session_history

Ideas for Today

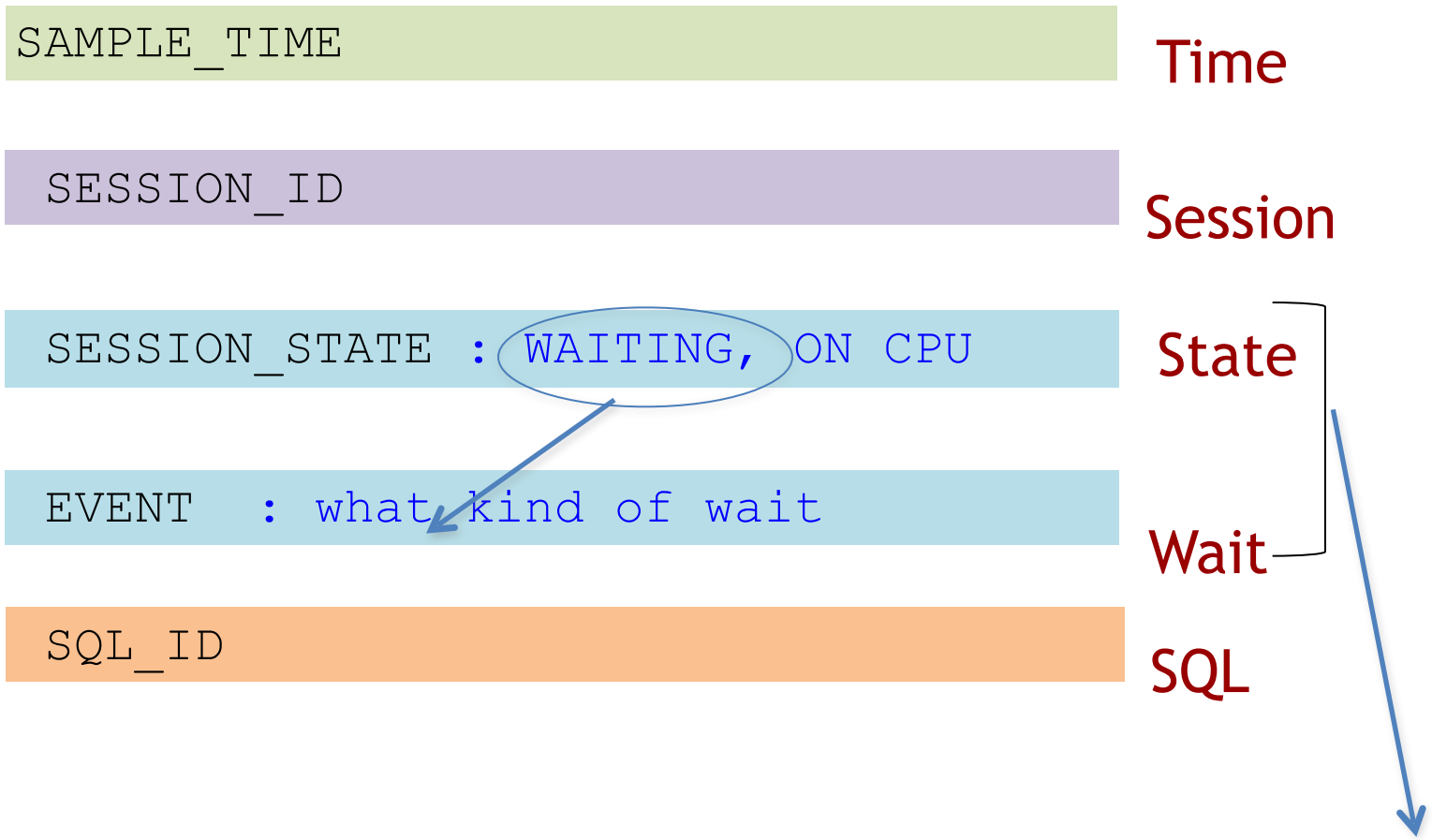


v$active_session_history

When	SAMPLE_ID	NUMBER
	SAMPLE_TIME	TIMESTAMP(3)
Session	SESSION_ID	NUMBER
	SESSION_SERIAL#	NUMBER
	USER_ID	NUMBER
	SERVICE_HASH	NUMBER
	SESSION_TYPE	VARCHAR2(10)
	PROGRAM	VARCHAR2(64)
	MODULE	VARCHAR2(48)
	ACTION	VARCHAR2(32)
	CLIENT_ID	VARCHAR2(64)
State	SESSION_STATE	VARCHAR2(7)
	WAIT_TIME	NUMBER
Wait	EVENT	VARCHAR2(64)
	EVENT_ID	NUMBER
	EVENT#	NUMBER
	SEQ#	NUMBER
	P1	NUMBER
	P2	NUMBER
	P3	NUMBER
	WAIT_TIME	NUMBER
	TIME_WAITED	NUMBER
	CURRENT_OBJ#	NUMBER
	CURRENT_FILE#	NUMBER
	CURRENT_BLOCK#	NUMBER0
SQL	SQL_ID	VARCHAR2(13)
	SQL_CHILD_NUMBER	NUMBER
	SQL_PLAN_HASH_VALUE	NUMBER
	SQL_OPCODE	NUMBER
	QC_SESSION_ID	NUMBER
	QC_INSTANCE_ID	NUMBER
	TIME_WAITED	NUMBER

Primary Fields of ASH

Activity : Who, What, When, How



Groupings – Top Consumer

SESSION_ID SESSION_SERIAL#	(identify SID reuse)
SESSION_TYPE	(FOREGROUND,BACKGROUND)
USER_ID	(SYS, SYSTEM, SCOTT etc)
SERVICE_HASH	(OE,GL,HR)
MODULE.ACTION	(PLSQL tagging)
CLIENT_ID	(identifying users in session pool)
PROGRAM	(SQL, JDBC, Forms etc)
SQL_ID QC_SESSION_ID QC_INSTANCE_ID	(Query Coordinator) (RAC)
EVENT + P1, P2, P3	
CURRENT_OBJ# CURRENT_FILE# CURRENT_BLOCK#	Only for I/O, some locks and buffer busy waits

DELPHIX

Counting is the key to ASH

Seconds ~= count(*)

```
Select count(*) from ASH
where {criteria}  and {time period};
```

AAS ~= count(*)/elapsed => average active sessions

```
Select count(*)/{time period in seconds} AAS
from ASH where …;
```

Top CPU Session

Top CPU consuming Session in last 5 minutes

```
Select
      session_id,
      count(*)
from
      v$active_session_history
where
      session_state= 'ON CPU' and
      SAMPLE_TIME > sysdate – (5/(24*60))
group by
      session_id
order by
        count(*) desc;
```

Who is the rogue session ?

Results Top CPU Session

Not AAS, what does count mean?

SESSION_ID	COUNT(*)
257	299
263	62
256	32
264	9
277	3

Running

Top CPU Session

Top CPU consuming Session in last 5 minutes

Who is the rogue session ?

```
Select
      session_id,
      count(*)/(5*60) – COUNT/ELAPSED=AAS
        100*(count(*)/(5*60) )  -- % active
from
      v$active_session_history
where
      session_state= 'ON CPU' and
      SAMPLE_TIME > sysdate – (5/(24*60))
group by
      session_id
order by
        count(*) desc;
```

Last 5 minutes

Running

Results Top CPU Session

AAS/100 = % active

SESSION_ID	AAS	%busy
257	.99	99
263	.21	21
256	.11	11
264	.03	3
277	.01	1

Making a 0-100% bar

```
Select
   session_id, count(*) ,
   round((count(*)*100)/(5*60),0) "%",
   lpad('*',
              10 *  (count(*)/(5*60)),
        '*') "Bar"
From
   v$active_session_history
Where
    session_state= 'ON CPU' and
    SAMPLE_TIME > sysdate - (5/(24*60))
group by session_id
order by count(*) desc
/
```

0 - 10 range

AAS per session 0-1

CPU with Bars

```
SESSION_ID   COUNT(*)           % Bar
---------- ---------- ---------- ------------
       257        299         99 |**********|
       263         62         21 |**        |
       256         32         11 |*         |
       264          9          3 |          |
       277          3          1 |          |
       258          1          0 |          |
       280          1          0 |          |
```

Bar shows 10% increments

Top Waiting Session

- in last 5 minutes

```
Select
    session_id,
    count(*)
from
    v$active_session_history
where
    session_state='WAITING'  and
    SAMPLE_TIME >  SYSDATE - (5/(24*60))
group by
    session_id
order by
        count(*) desc;
```

Waiting

DELPHIX

Top Waiting Session Results

SESSION_ID	COUNT(*)
272	224
254	8
249	5
276	5
277	4
270	1

Waiting

Top SQL from ASH

```
select
    ash.SQL_ID
    sum(decode(ash.session_state,'ON CPU',1,0))     "CPU",
    sum(decode(ash.session_state,'WAITING',1,0))    -
    sum(decode(ash.session_state,'WAITING', decode(en.wait_class, 'User I/O',1,0),0))    "WAIT"
    sum(decode(ash.session_state,'WAITING', decode(en.wait_class, 'User I/O',1,0),0))    "IO" ,
    sum(decode(ash.session_state,'ON CPU',1,1))     "TOTAL"
from v$active_session_history ash,
        v$event_name en
where SQL_ID is not NULL  and en.event#=ash.event#
group by sql_id
order by sum(decode(session_state,'ON CPU',1,1))   desc
```

Running

Waiting

No time window specified

DELPHIX

Top SQL from ASH Results

SQL_ID	CPU	WAITING	IO	TOTAL
4c1xvq9ufwcjc	**23386**	**0**	**0**	**23386**
6wjw6rz5uvbp3	99	0	23	122
968dm8hr9qd03	97	0	22	119
938jp5gasmrah	90	0	25	115
cv8xnv81kf582	42	0	9	51
6p9bzu19v965k	21	0	0	21
5zu8pxnun66bu	15	0	0	15
db2jr13nup72v	9	0	0	9
7ks5gnj38hghv	8	0	0	8

Top Session

```
select
    ash.session_id,
    ash.session_serial#,
    ash.user_id,
    ash.program,
    sum(decode(ash.session_state,'ON CPU',1,0))     "CPU",
    sum(decode(ash.session_state,'WAITING',1,0))    -
    sum(decode(ash.session_state,'WAITING',
      decode(en.wait_class,'User I/O',1, 0 ), 0))    "WAITING" ,
    sum(decode(ash.session_state,'WAITING',
      decode(en.wait_class,'User I/O',1, 0 ), 0))    "IO" ,
    sum(decode(session_state,'ON CPU',1,1))     "TOTAL"
from v$active_session_history ash,
      v$event_name en
where en.event# = ash.event#
group by session_id,user_id,session_serial#,program
order by sum(decode(session_state,'ON CPU
```

order by

No time window specified

Top Session Results

SESSION_ID	SERIAL#	USER_ID	PROGRAM	CPU	WAITING	IO
247	61970	1	sqlplus	11698	0	0
277	1	0	oracle@labsfrh903 (LGWR)	14	21	0
276	1	0	oracle@labsfrh903 (CKPT)	19	10	0
278	1	0	oracle@labsfrh903 (DBW0)	29	0	0
280	1	0	oracle@labsfrh903 (PMON)	19	0	0
254	22617	5	Executor.exe	13	0	3
255	12877	5	Executor.exe	11	0	5
257	33729	5	Executor.exe	15	0	1
255	13417	5	Executor.exe	14	0	2


Reading


Running


Waiting

Top Session w/ Username

```
select
      decode(nvl(to_char(s.sid),-1),-1,'DISCONNECTED','CONNECTED')
                                    "STATUS",
      topsession.session_id          "SESSION_ID",
      u.name  "NAME",
      topsession.program             "PROGRAM
      max(topsession.CPU)           "CPU",
      max(topsession.WAITING)       "WAITING",
      max(topsession.IO)            "IO",
      max(topsession.TOTAL)         "TOTAL"
   from {previous query}                topsession   ,
                                        v$session    s,
                                        user$        u
  where
              u.user# =topsession.user_id and
             /* outer join to v$session because the session might be disconnected */
             topsession.session_id       = s.sid       (+) and
             topsession.session_serial# = s.serial#   (+)
  group by  topsession.session_id, topsession.session_serial#, topsession.user_id,
             topsession.program, s.username,s.sid,s.paddr,u.name
  order by max(topsession.TOTAL) desc
```

ASH

Connected?

User name

No time window specified

Top Session
Finding a Rogue User

STATUS	SESSION_ID	NAME	PROGRAM	CPU	WAITING	IO
CONNECTED	**247**	**CPU_Monger**	**ChMgr304.exe**	**11704**	**0**	**0**
CONNECTED	277	SYS	oracle@labsfrh903 (LGWR)	14	19	0
CONNECTED	278	SYS	oracle@labsfrh903 (DBW0)	29	0	0
CONNECTED	276	SYS	oracle@labsfrh903 (CKPT)	18	9	0
CONNECTED	280	SYS	oracle@labsfrh903 (PMON)	20	0	0
DISCONNECTED	255	SYSTEM	Executor.exe	11	4	5
DISCONNECTED	257	SYSTEM	Executor.exe	13	0	3
DISCONNECTED	255	SYSTEM	Executor.exe	14	0	2
DISCONNECTED	257	SYSTEM	Executor.exe	13	0	3

Queries can be complicated

- Key to ASH
- Difficulties
- Pitfalls
- Solution

Keys to ASH

1. Seconds =

```
COUNT(*)
```

2. AAS =

```
COUNT(*) / Elapsed
```

3. Combine CPU state with EVENT

```
decode(session_state,
        'ON CPU','ON CPU',
        EVENT)
```

DELPHIX

Queries can be complicated

Difficulties

- Query complicated
- Time windows difficult
- dba_hist_active_sess_history only 1/10 samples kept

Pitfalls

- Ignore time_waited (for the most part)
- ~~WAIT_TIME~~ is time of last wait, > 0 , means on CPU

Solution

repository of pre-written queries

Solution: ASH Masters, Collection of ASH SQL scripts

1. **Load Charts**
2. **Wait Analysis** – locks, latches, bbw
3. **SQL elapsed times**
4. **Top** – sql, session, wait, procedure, object
5. **I/O**

- **Latency** * - not from ASH, but important and related

Before ASH Masters

remember **ashrpt.sql**

1. ASHRPT
   - Based entirely on v$active_session_history
   - @?/rdbms/admin/ashrpt.sql
   - Exec ASH_REPORT_TEXT/HTML

```
select *  from table
(dbms_workload_repository.ash_report_text(
      (select dbid from v$database),
       1,
        sysdate – 1/24,
        sysdate  )) ;
```

ASHRPT

```
ASH Report For TESTDB/testdb
DB Name          DB Id     Inst
------------ ----------- ----
TESTDB        2371570538 test
CPUs             SGA Size
---- ------------------ -----
   2       1,000M (100%)
           Analysis Begin Time
             Analysis End Time
                  Elapsed Time
                  Sample Count
      Average Active Sessions
  Avg. Active Session per CPU
                  Report Target
Top User Events

Event
------------------------------
CPU + Wait for CPU
enq: TX - row lock contention
buffer busy waits
latch: cache buffers chains
```

ASH Report For ORCL/orcl

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
ORCL	1121488884	orcl	1	10.2.0.1.0	NO	KYLEHPD

CPUs	SGA Size	Buffer Cache	Shared Pool	ASH Buffer Size
2	156M (100%)	68M (43.6%)	79M (50.9%)	3.8M (2.4%)

	Sample Time	Data Source
Analysis Begin Time:	14-Dec-07 14:11:38	V$ACTIVE_SESSION_HISTORY
Analysis End Time:	14-Dec-07 14:16:38	V$ACTIVE_SESSION_HISTORY
Elapsed Time:	5.0 (mins)	
Sample Count:	165	
Average Active Sessions:	0.55	
Avg. Active Session per CPU:	0.28	
Report Target:	None specified	

ASH Report

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

ASH RPT

1) General info
2) Top User Events ***
3) Top Background Events
4) Top Event P1/P2/P3 Values
5) Top Service/Module
6) Top Client IDs
7) Top SQL Command Types
8) Top SQL Statements ***

9) Top SQL using literals
10) Top Sessions ***
11) Top Blocking Sessions
12) Top Sessions running PQs
13) Top DB Objects
14) Top DB Files
15) Top Latches
16) Activity Over Time ***

ASHRPT over Time

- Waits over Time
  - Not in AAS
- Difficult but better than nothing

Slot Time (Duration)	Slot Count	Event	Event Count	% Event
14:11:38 (22 secs)	11	CPU + Wait for CPU	6	3.64
		db file sequential read	5	3.03
14:12:00 (1.0 min)	32	CPU + Wait for CPU	23	13.94
		db file sequential read	6	3.64
		log file parallel write	1	0.61
14:13:00 (1.0 min)	37	CPU + Wait for CPU	21	12.73
		db file sequential read	15	9.09
		log file parallel write	1	0.61
14:14:00 (1.0 min)	27	CPU + Wait for CPU	18	10.91
		db file sequential read	7	4.24
		latch free	2	1.21
14:15:00 (1.0 min)	28	CPU + Wait for CPU	16	9.70
		db file sequential read	8	4.85
		log file parallel write	3	1.82
14:16:00 (38 secs)	30	CPU + Wait for CPU	17	10.30
		db file sequential read	12	7.27
		log file parallel write	1	0.61

ASH Masters

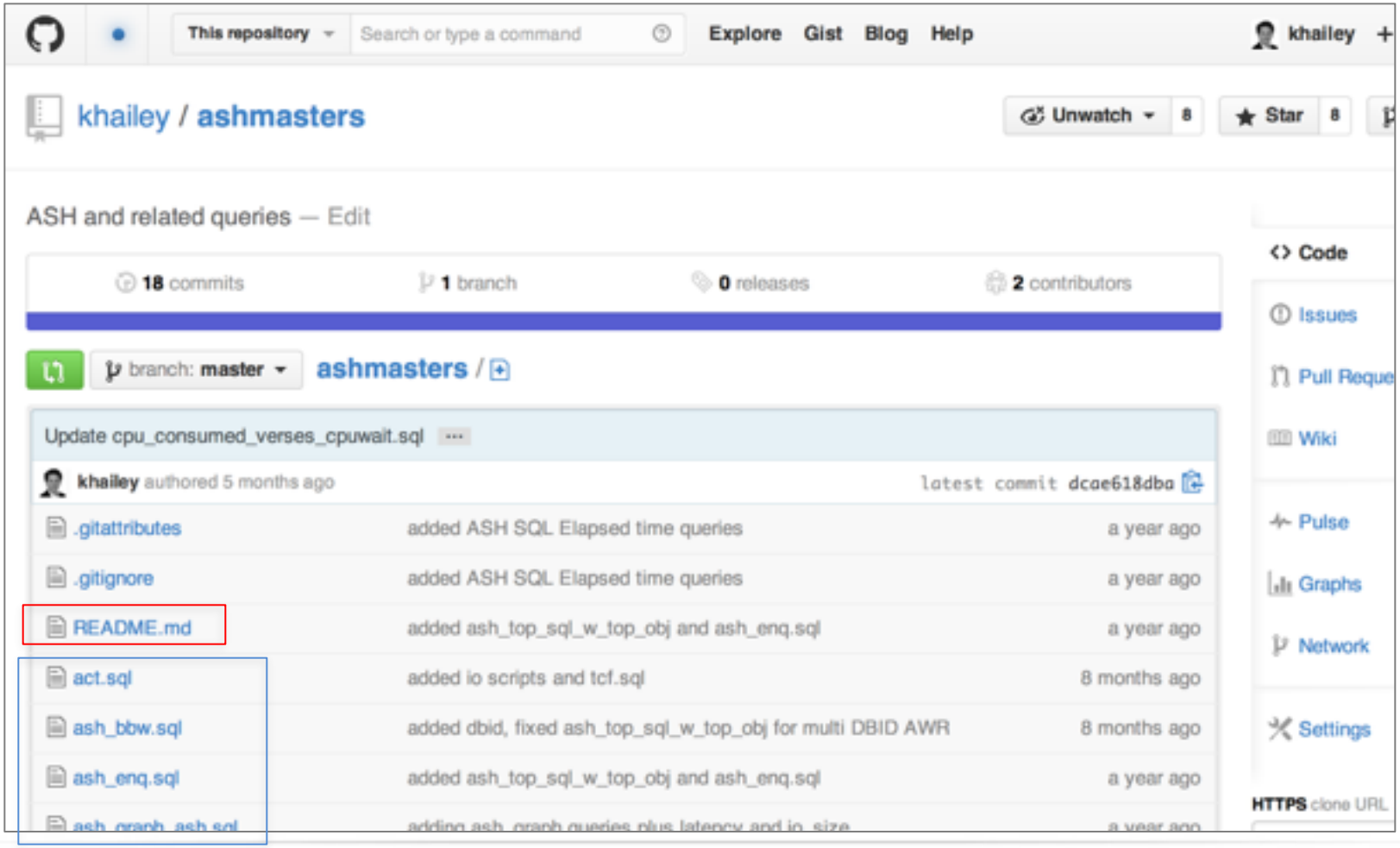
intro to **github**

1. **Load Charts**
2. **Wait Analysis** – locks, latches, bbw
3. **SQL elapsed times**
4. **Top** – sql, session, wait, procedure, object
5. **I/O** – size, object

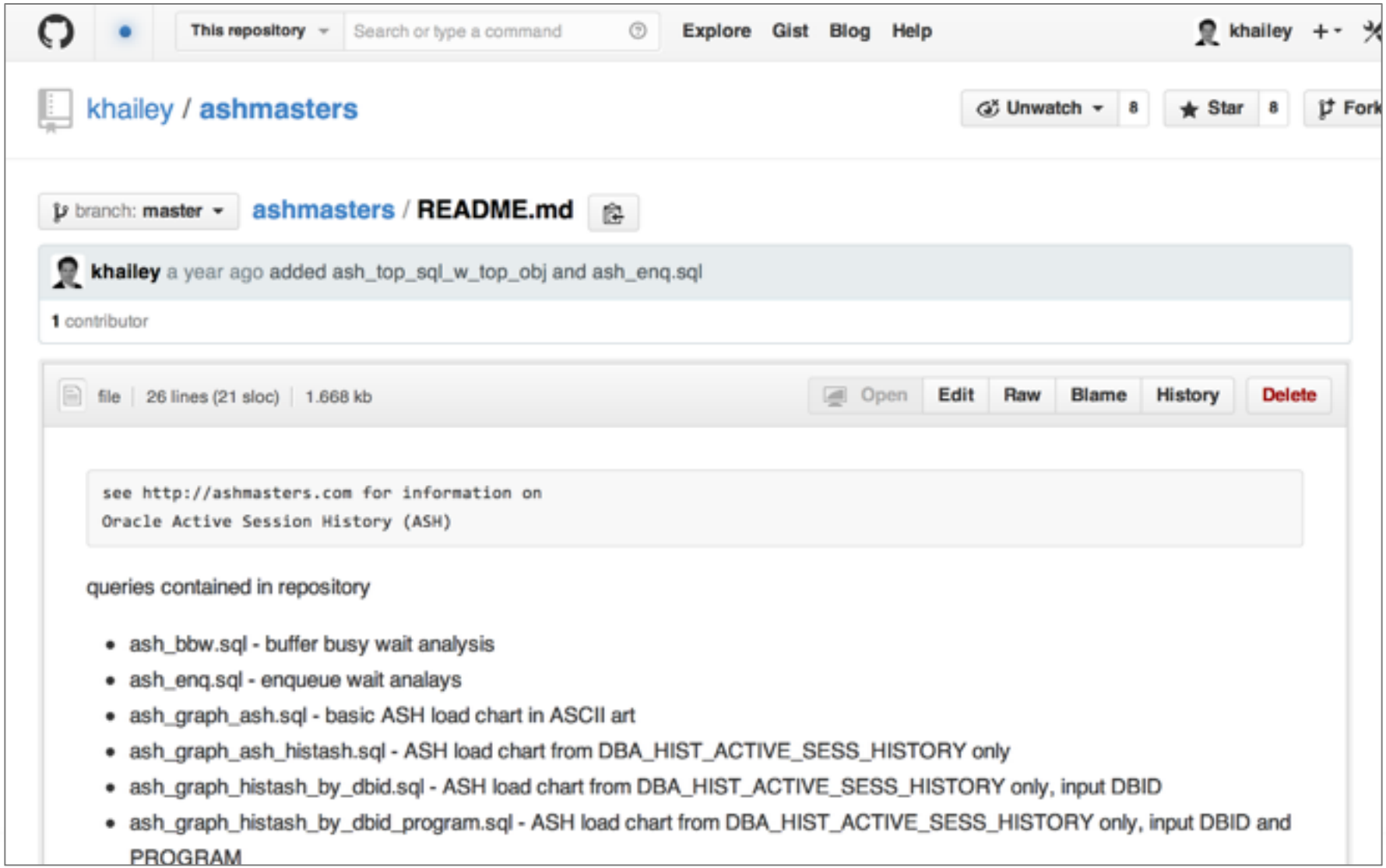
Extra: Latency * - not from ASH, but important

ASH Masters on github.com

- https://github.com/khailey/ashmasters



ASH Masters README.md



1. Load Charts

How to get a quick overview of all the data in the ASH ?

Reproduce Active Session OEM tab in SQL

- Load graph
  - Raw data
  - Create graphics
- Top waits
- History from dba_hist_active_sess_history

1. Load Charts: ash_graph.sql

```
@ash_graph
TIME                AAS      GRAPH
----------------  -------  ----------------------
06-AUG  14:00:00     2.24  ++--------2---
06-AUG  15:00:00     6.67  ++++------2----------
06-AUG  16:00:00     2.59  ++--------2----
06-AUG  17:00:00     1.26  ++-----    2
06-AUG  18:00:00     1.38  +++----    2
06-AUG  19:00:00     1.74  ++-------  2
06-AUG  20:00:00      .99  +----      2
06-AUG  21:00:00     1.22  ++-----    2
06-AUG  22:00:00     1.66  ++------   2
06-AUG  23:00:00     1.08  +----      2
07-AUG  00:00:00      .83  +---       2
07-AUG  01:00:00     1.74  ++-------  2
07-AUG  02:00:00     2.47  ++--------2----
07-AUG  03:00:00     6.59  +++-------2----------
```

What is this ?

1. Load Charts: ash_graph.sql

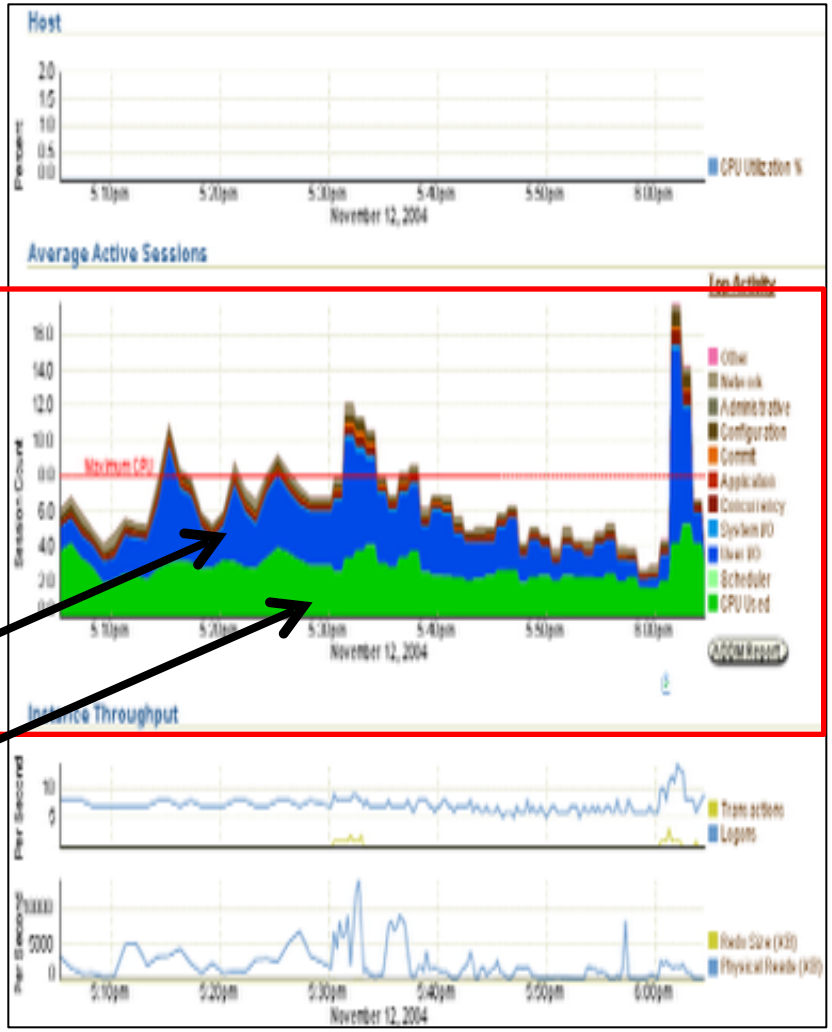
```
@ash_graph
TIME                  AAS      GRAPH
----------------    -------   ----------------------
06-AUG  13:00:00        .33   +-        2
06-AUG  14:00:00       2.24   ++--------2---
06-AUG  15:00:00       6.67   ++++------2----------
06-AUG  16:00:00       2.59   ++--------2----
06-AUG  17:00:00       1.26   ++-----   2
06-AUG  18:00:00       1.38   +++----   2
06-AUG  19:00:00       1.74   ++------- 2
06-AUG  20:00:00        .99   +----     2
06-AUG  21:00:00       1.22   ++-----   2
06-AUG  22:00:00       1.66   ++------  2
06-AUG  23:00:00       1.08   +----     2
07-AUG  00:00:00        .83   +---      2
07-AUG  01:00:00       1.74   ++------- 2
07-AUG  02:00:00       2.47   ++--------2----
07-AUG  03:00:00       6.59   +++-------2----------
07-AUG  04:00:00       1.95   ++++++--- 2
07-AUG  05:00:00       3.08   +++++-----2------
```

1. Load Charts: ash_graph.sql

```
TM           NPTS  AAS GRAPH
-----------  ----  ----------
27-MAR       2641  .04 --                                2
27-MAR       3501  .05 --                                2
27-MAR          1  .00                                   2
27-MAR       2781  .05 +--                               2
27-MAR       3581  .11 ++-----                           2
27-MAR       3581  .16 ++------                          2
27-MAR       3581  .16 ++------                          2
27-MAR       3581  .17 ++------                          2
27-MAR       3201  .19 ++--------                        2
27-MAR       3551  .21 +---------                        2
27-MAR       3561  .31 ++---------                       2
27-MAR        641  .01                                   2
27-MAR       2041  .00                                   2
27-MAR         11  .00                                   2
27-MAR       1361  .02 -                                 2
27-MAR       4351  .00                                   2
28-MAR        419  .01                                   2
28-MAR       3543  .46 ++                                2
28-MAR       3325  1.83 +++-------                       2
28-MAR       3577  .32 -                                 2
28-MAR       3589  .31 -                                 2
28-MAR       2948  .40 --                                2
28-MAR        419  .00                                   2
```

"-" = WAIT

"+" = CPU



which waits ?

1. Load Charts: ash_graph_waits.sql ash_graph_ash.sql adf

TO_CHAR(STA	AAS1	FIRST	AAS2	SECOND	GRAPH	
27 06:00:00	.15	db file sequent	.11	enq: TX - alloc	--	2
27 07:00:00	.23	db file sequent	.11	enq: TX - alloc	--	2
27 08:00:00	.00	CPU				2
27 09:00:00	.24	db file sequent	.11	enq: TX - alloc	+--	2
27 10:00:00	.52	db file sequent	.34	CPU	++----	2
27 11:00:00	.82	db file sequent	.40	log file sync	++------	2
27 12:00:00	.79	db file sequent	.38	log file sync	++------	2
27 01:00:00	.84	db file sequent	.51	log file sync	++------	2
27 02:00:00	.89	db file sequent	.48	log file sync	++--------2-	
27 03:00:00	1.25	db file sequent	.57	log file sync	+---------2-	
27 04:00:00	1.65	db file sequent	1.03	log file sync	++--------2-------	
27 07:00:00	.03	db file sequent	.01	CPU		2
27 08:00:00	.00	CPU	.00	db file sequent		2
27 09:00:00						2
27 10:00:00	.06	db file scatter	.06	db file sequent	-	2
27 11:00:00	.00	control file se	.00	db file sequent		2
28 12:00:00	.00	db file sequent	.00	CPU		2
28 01:00:00	.36	CPU	.07	latch: cache bu	++	2
28 02:00:00	.53	CPU	.47	log file sync	+++------	2
28 03:00:00	.16	db file sequent	.06	CPU	-	2
28 04:00:00	.16	db file sequent	.11	direct path wri	-	2
28 05:00:00	.35	direct path wri	.02	CPU	--	2
28 06:00:00	.00	db file sequent	.00	CPU		2

Top Two Waits

DELPHIX

1. Load Charts: basics, fake query

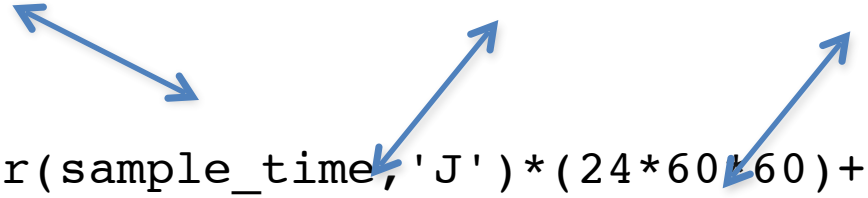
```
Select pseudo columns
          start_time
        ,  bucket_id
        ,  count(cpu)
        ,  count(waits)
      from
         v$active_session_history ash

START_TIME BUCKET_ID              CPU       WAITS
---------- ------------ ---------- ----------
25-NOV-13            55        397       28231
25-NOV-13            56         80        5631
```

1. Load Charts: components

ID

ID = truncate (seconds in date / seconds in bucket)

((Julian days * seconds in a day) + (seconds in last day)) / bucket size in seconds



```
trunc((to_char(sample_time,'J')*(24*60*60)+
to_char(sample_time,'SSSSS'))/&v_secs)
```

Counts

```
, sum(decode(session_state,'ON CPU',1,0))     cpu
, sum(decode(session_state,'WAITING',1,0))    waits
```

1. Load Charts: real query

```
select
         trunc((to_char(sample_time,'J')*(24*60*60)+
                to_char(sample_time,'SSSSS'))/&v_secs) id
       , sum(decode(session_state,'ON CPU',1,0))        cpu
       , sum(decode(session_state,'WAITING',1,0))       waits
    from
       v$active_session_history ash
       -- dba_hist_active_sess_history
    group by
       trunc((to_char(sample_time,'J')*(24*60*60)+
              to_char(sample_time,'SSSSS'))/&v_secs)
```

ID

CPU

Waits

Group by

ID	CPU	WAITS
4.2432E+10	1	4
4.2432E+10	9	18
4.2432E+10	8	25
4.2432E+10	9	26

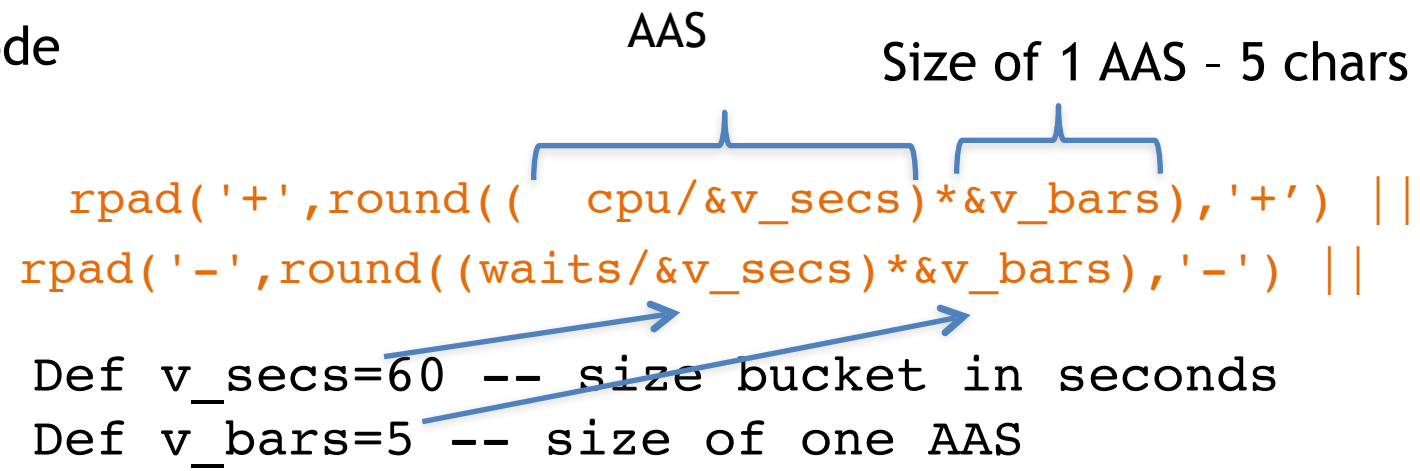
1. Load Charts: how do you draw a load line?

If the value of 1 AAS is 5 characters wide then

++++++-----

= AAS of 2 (1 wait, 1 CPU)

- code



What about a yard stick ?

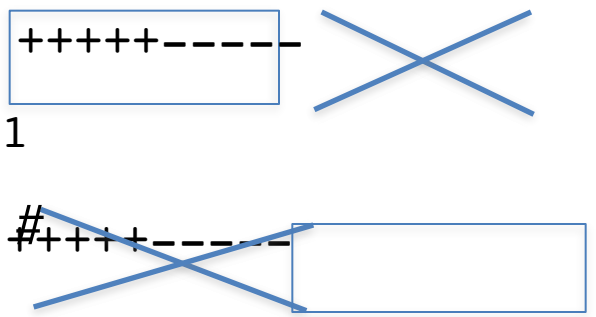
1. Load Charts: basics : graphics

+++++1-----

```
rpad('+',round((  cpu/&v_secs)*&v_bars),'+') ||
 rpad('-',round((waits/&v_secs)*&v_bars),'-') ||
rpad(' ',p.value * &v_bars,' '),                    --
blanks
```

Trick, add in the number cores

1. Draw full line, cut at core #
2. Add core #
3. Draw full line again cut before first core #



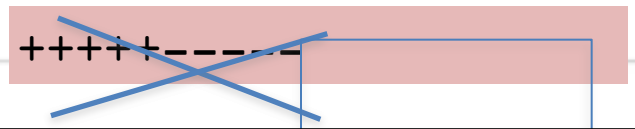
1. Load Charts: graphics

```
Def v_bars=5 -- size of one AAS in characters
Def v_graph=80
substr(
      substr(
            rpad('+',round((cpu*&v_bars)/&v_secs),'+') ||
            rpad('-',round((waits*&v_bars)/&v_secs),'-')  ||
            rpad(' ',p.value * &v_bars,' '),
         0,(p.value * &v_bars)) ||
       p.value  ||
       substr(
            rpad('+',round((cpu*&v_bars)/&v_secs),'+') ||
            rpad('-',round((waits*&v_bars)/&v_secs),'-')  ||
           rpad(' ',p.value * &v_bars,' '),
       (p.value * &v_bars))
  ,0,&v_graph)
```

First half of line

Add core count

2nd half of line



1. Load Charts: basics

```
select
        to_char(to_date( trunc((id*&v_secs)/ (24*60*60)) || ' ' ||   -- Julian days
                         mod((id*&v_secs), (24*60*60))               -- seconds in the day
                , 'J SSSSS' ), 'MON DD YYYY HH24:MI:SS') start_time,
        substr( substr(
                  rpad('+',round((  cpu/&v_secs)*&v_bars),'+') ||
                  rpad('-',round((waits/&v_secs)*&v_bars),'-') ||
                  rpad(' ',p.value * &v_bars,' ') ,0,(p.value * &v_bars)) ||
             p.value  ||
             substr(
                  rpad('+',round((  cpu/&v_secs)*&v_bars),'+') ||
                  rpad('-',round((waits/&v_secs)*&v_bars),'-') ||
                  rpad(' ',p.value * &v_bars,' '), (p.value * &v_bars))
           ,0,&v_graph) graph
from (
   select  trunc((to_char(sample_time,'J')*(24*60*60)+to_char(sample_time,'SSSSS'))/&v_secs) id
         , sum(decode(session_state,'ON CPU',1,0))    cpu
         , sum(decode(session_state,'WAITING',1,0))   waits
      from  v$active_session_history ash
     group by
        trunc((to_char(sample_time,'J')*(24*60*60)+to_char(sample_time,'SSSSS'))/&v_secs)
) aveact,
  v$parameter p
where p.name='cpu_count'
order by id
/
```

Date (from ID)

Bar

ID

CPU

WAITS

1. Load Charts : ash_graph.sql

```
START_TIME              GRAPH
--------------------    --------------------------
NOV 03 2013 22:02:00    ++---1------
NOV 03 2013 22:03:00    ++---1------
NOV 03 2013 22:04:00    +++--1-------
NOV 03 2013 22:05:00    +----1-
NOV 03 2013 22:06:00    +++--1--
```

When waiting,
which are the top waits ?

1. Load Charts : events ash_graph_waits.sql

```
Select
     trunc((to_char(sample_time,'J')*(24*60*60)+to_char(sample_time,'SSSSS'))/&v_secs)   id
   , decode(session_state,'ON CPU','ON CPU',ash.event)                                   event
   , count(*)                                                                            event_count
From
    v$active_session_history ash
group by
      trunc((to_char(sample_time,'J')*(24*60*60)+to_char(sample_time,'SSSSS'))/&v_secs)
    , decode(session_state,'ON CPU','ON CPU',ash.event)
```

ID	EVENT	EVENT_COUNT
3537592817	control file heartbeat	2
3537592818	ADR block file read	1
3537592818	ARCH wait for process start 3	3
3537592818	ON CPU	9
3537592818	control file parallel write	4

1. Load Charts : per bucket, rank the waits

```
select
        id
      , event
      , row_number() over ( partition by id order by event_count desc ) rank
      , ratio_to_report( event_count ) over ( partition by id )          pct
      , sum(decode(event,'ON CPU',event_total,0))                        cpu
      , sum(decode(event,'ON CPU',0,event_total))                        waits
   from ( Previous Query  )  chunks
   group by id,  event, event_total
```

Rank

Rank

ID	EVENT	RANK	PCT	CPU	WAITS
3537592818	ON CPU	1	.290	9	0
3537592818	db file sequential read	2	.161	0	5
3537592818	log file switch (checkpoint i	3	.161	0	5
3537592818	control file parallel write	4	.129	0	4
3537592818	ARCH wait for process start	5	.096	0	3
3537592818	ADR block file read	6	.032	0	1

%

1. Load Charts : raw data ash_graph_waits.sql

```
select id
  , round(max(decode(top.rank,1,pct,null)),2) fpct
  , max(
     decode(top.rank,1,decode(top.event,'ON CPU','CPU',event),null)) first
  , round(max(decode(top.rank,2,pct,null)),2) spct
  , max(
     decode(top.rank,2,decode(top.event,'ON CPU','CPU',event),null)) second
  , sum(waits) waits
  , sum(cpu) cpu
…
```

ID	FPCT	FIRST	SPCT	SECOND	WAITS	CPU
3537592817	1.00	control file he			2	0
3537592818	.29	CPU	.16	db file sequent	22	9
3537592819	.72	control file pa	.14	CPU	37	6
3537592820	.75	CPU	.25	control file pa	1	3
3537592821	.91	CPU	.09	direct path rea	1	10
3537592823	.75	CPU	.25	db file paralle	1	3

1. Load Charts : ash_graph_waits.sql

```
TO_CHAR( PCT1 FIRST           PCT2 SECOND          GRAPH
-------- ---- -------------- ---- -------------- ----------------------------------------
15 19:00   64 CPU             21 db file sequent ++o                  4
15 20:00   63 CPU             19 read by other s ++++o-               4
15 21:00   31 db file sequent  24 CPU             ++ooo----           4
15 22:00   35 CPU             24 db file scatter +++++ooooooo---     4
15 23:00   29 log file sync    25 db file sequent ++++ooooooooo-------4-------------
16 00:00   52 db file sequent  27 CPU             +++++++++ooooooooo4ooooooooooooooo--
16 01:00   57 CPU             36 db file sequent ++++++++++ooooooo 4
16 02:00   38 db file sequent  21 CPU             ++++++ooooooooooo--4---------
16 03:00   69 db file sequent  20 CPU             +++ooooooooooo      4
16 04:00   45 db file sequent  28 CPU             o                   4
16 05:00   58 db file sequent  24 CPU             +ooo                4
16 06:00   41 db file sequent  39 CPU             +oo                 4
```

o = I/O
+ = cpu
- = wait

OK, what about yesterday?

1. Load Charts : dba_hist_active_sess_history?

dba_hist_active_sess_history

- Week of ASH data !!
- Only 1 in 10 v$active_session_history rows kept
- Has DBID
  - Can query different databases in same repository

1. Load Charts: queries with history

- V$ACTIVE_SESSION_HISTORY - live
  - ash_graph_waits.sql – minute buckets

Fast, in memory

- V$ACTIVE_SESSION_HISTORY + DBA_HIST_ACTIVE_SESS_HISTORY – combine
  - ash_graph_waits_histash.sql – minute buckets

slower

- DBA_HIST_ACTIVE_SESS_HISTORY - history only (DBID)
  - ash_graph_histash_by_dbid.sql - input DBID
  - ash_graph_histash_by_dbid_program.sql - input DBID and PROGRAM
  - ash_graph_histash_by_dbid_sqlid.sql - input DBID and a SQL_ID

repositories

2. SQL Elapsed times (11g+)

ASH 11g added

sql_exec_id
sql_exec_start

for every sample
When the same SQL is still executing
ASH it will have the same sql_exec_id and sql_exec_start
and same sql_id

2. SQL Elapsed times

- ash_sql_elapsed.sql
  - longest running SQL

```
SQL_ID          COUNT(*)       MX         AV         MIN
------------- ---------- ------- --------- ----------
0fvrpk7476b7y          26    3068     133.1          0
1pjp66rxcj6tg          15    3106     767.7         57
```

- ash_sql_elapsed_hist.sql
  - with histogram of execution times

```
SQL_ID              CT         MX         MN         AV       1       2       3       4       5
------------- ---------- ---------- ---------- -------- ------ ------ ------- ------- -------
5k7vccwjr5ahd       2653       1963          0     33.4   2623     15       8       4       3
ds8cz0fb8w147        161       2531         13    273.8    136     18       5       1       1
```

- ash_sql_elapsed_hist_longestid.sql
  - execution id of longest running query

```
SQL_ID            CT         MX   MN       AV MAX_RUN_TIME                          LONGEST_SQ      1       2       3       4   5
------------- ------ ---------- ---- -------- ------------------------------------ ---------- ------ ------- ------- ------- ---
2spgk3k0f7quz    251      29607    0    546.0 11-04-12 12:11:47 11-04-12 20:25:14 16781748      247       2       0       0   2
990m08w8xav7s    591       7681    0     52.0 11-04-13 00:39:27 11-04-13 02:47:28 16786685      587       0       0       2   2
```

2. SQL Elapsed times: ash_sql_elapsed.sql

SQL_ID	Executions	MX	AV	MIN
0fvrpk7476b7y	26	3068	133.1	0
1pjp66rxcj6tg	15	3106	767.7	57
8r5wuxk1dprhr	39	3510	841.0	24
0w5uu5kngyyty	21	3652	442.3	0
0hbv80w9ypy0n	161	4089	1183.9	0
71fwb4n6a92fv	49	4481	676.9	30
0bujgc94rg3fj	604	4929	24.7	0
64dqhdkkw63fd	1083	7147	7.2	0
990m08w8xav7s	591	7681	51.8	0
2n5369dsuvn5a	16	10472	5726.8	303
2spgk3k0f7quz	251	29607	546.1	0
36pd759xym9tc	12	37934	23861.9	1391
497wh6n7hu14f	49	69438	5498.2	0

2. SQL Elapsed times : raw data

seconds query had been running = **Sample_time - sql_exec_start**

```
select
      sql_id,
      sql_exec_id,
      (cast(sample_time id ate) -  cast(sql_exec_start as date Seconds running
   from
         v$active_session_history
   where sql_exec_id is not null
```

SQL_ID	SQL_EXEC_ID	TM
acc988uzvjmmt	16777220	3
acc988uzvjmmt	16777220	2
acc988uzvjmmt	16777220	1
acc988uzvjmmt	16777220	0

now select max seconds for each sql_exec_id

2. SQL Elapsed times

```
select
    sql_id,
    sql_exec_id,
    max(tm) tm
from ( previous query )
```

Final execution time is max of all execs

DELPHIX

2. SQL Elapsed times

```
select
        sql_id,
        count(*),
        max(tm) mx,
        avg(tm) av,
        min(tm) min
from ( previous query )
```

each SQL_ID
- max
- min
- average

execution times

2. SQL Elapsed times: ash_sql_elapsed.sql

SQL_ID	COUNT(*)	MX	AV	MIN
0fvrpk7476b7y	26	3068	133.1	0
1pjp66rxcj6tg	15	3106	767.7	57
8r5wuxk1dprhr	39	3510	841.0	24
0w5uu5kngyyty	21	3652	442.3	0
0hbv80w9ypy0n	161	4089	1183.9	0
71fwb4n6a92fv	49	4481	676.9	30
0bujgc94rg3fj	604	4929	24.7	0
64dqhdkkw63fd	1083	7147	7.2	0

What was the distribution?

2. SQL Elapsed times: ash_sql_elapsed_hist.sql

Histogram buckets: elapsed time spread

SQL_ID	CT	MX	MN	AV	1	2	3	4	5
401ayw4r7n6kz	1	1983	1983	1983.0	0	0	0	0	1
30hzp85f3qtxj	179	3029	29	111.0	177	1	0	0	1
9k86k2zvht9pt	1	3052	3052	3052.0	0	0	0	0	1
bnddu47dqmzqd	10	3138	68	732.3	7	1	1	0	1
gc7khrc2mx86m	1	3618	3618	3618.0	0	0	0	0	1
8rua4c9agcqkb	978	820	0	72.7	830	116	21	9	2

Bucket interval = max time / 5

820/5 = 164

1. 0 - 164 830 executions
2. 165 - 328 115 executions
3. 329 - 492 21 executions
4. 493 - 656 9 executions
5. 657 - 820 2 executions

2. SQL Elapsed Times ash_sql_elapsed_hist_longestid.sql

OK found the longest but

- When did it happen?
- What was the sql_exec_id of the longest?

SQL_ID	CT	MX	MN	AV	MAX_RUN_TIME	LONGEST_SQ	1	2	3	4	5
30hzp85f3qtxj	179	3029	29	111.0	11-11-07 13:46:12 11-11-07 14:36:41	16777250	177	1	0	0	1
C61wk6d7ssxxc	20	1801	258	743.0	11-11-07 12:01:02 11-11-07 12:31:03	16777216	13	2	0	0	5
769uu28qm4thw	17	1427	0	479.0	11-11-07 14:04:02 11-11-07 14:27:49	16777762	10	2	2	1	2
Fuzcbdt08xjcd	74	315	33	248.0	11-11-07 12:10:05 11-11-07 12:15:20	16790567	5	4	9	5	51
Bkv51bug8ag7c	29	307	1	36.0	11-11-07 11:58:30 11-11-07 12:03:37	16790749	26	2	0	0	1
1wgsn7mmf6kqc	131	299	0	6.0	11-11-07 12:01:08 11-11-07 12:06:07	16784143	130	0	0	0	1
3b7q0hd1q8pw0	17	288	1	118.0	11-11-07 09:42:21 11-11-07 09:47:09	16794374	7	4	0	1	5
1v6yyfy630rkj	13	135	42	65.0	11-11-07 07:49:12 11-11-07 07:51:27	16816140	7	4	1	0	1
1t715k5p9uxxx	50	95	0	32.0	11-11-07 05:17:55 11-11-07 05:19:30	16777225	32	1	1	0	16
A98fbc69py0us	11	91	59	74.0	11-11-07 14:55:55 11-11-07 14:57:26	16777389	1	4	3	2	1
0g53kf4gr3vrg	25	53	32	46.0	11-11-07 15:35:31 11-11-07 15:36:24	16777588	1	0	5	14	5

start

End

2. SQL Elapsed Times – lookup by SQL ID

```
Select
      decode(session_state,'ON CPU','ON CPU', event)
From
     v$active_session_history
Where
    sql_exec_id = 16777217
Order by sample_id
```

```
DECODE(SESSION_STATE,'ONCPU','ONCPU',EVENT)
----------------------------------------
db file parallel read
db file sequential read
ON CPU
ON CPU
ON CPU
ON CPU
```

3. Wait Analysis: v$active_session_history

- Buffer busy waits
- Enqueue transaction waits

3. Wait Analysis: buffer busy waits

Top 5 Timed Events

Event	Waits	Time (s)	% Total Call Time
buffer busy waits	2,748	250	78.72
CPU time		32	10.16
free buffer waits	1,588	15	4.63
write complete waits	10	8	2.51
log buffer space	306	5	1.51

Buffer Busy Wait – trying to modify a block
Who blocks?
What object?
what is the SQL?

AWR fails

3. Wait Analysis: buffer busy waits

To solve Buffer Busy Waits, need

- Block type
- Object
  - Table w/ data block => ASM or Freelists,
  - Index w/data block => reverse or hash partition
- File
  - Temp files problem with extents

ash_bbw.sql

```
OBJN                    OTYPE     FILEN  BLOCKN SQL_ID        BLOCK_TYPE
----------------------- ---------- ----- ------- ------------- ------------
53218 BBW_INDEX_VAL_I   INDEX         1   64826 97dgthz60u28d data block 1
53218 BBW_INDEX_VAL_I   INDEX         1   64826 gypmcfzruu249 data block 1
53218 BBW_INDEX_VAL_I   INDEX         1   64826 2vd1w5kgnfa5n data block 1
53218 BBW_INDEX_VAL_I   INDEX         1   64826 3p3qncvp2juxs data block 1
53218 BBW_INDEX_VAL_I   INDEX         1   64826 6avm49ys4k7t6 data block 1
```

3. Wait Analysis: Joining ASH with v$waitstat

```
select
       o.object_name obj,
       o.object_type otype,
       ash.SQL_ID,
       w.class
from v$active_session_history ash,
     ( select rownum class#, class from v$waitstat ) w,
      all_objects o
where event='buffer busy waits'
   and w.class#(+)=ash.p3
   and o.object_id (+)= ash.CURRENT_OBJ#
Order by sample_time;
```

```
OBJ    OTYPE  SQL_ID        CLASS
------ ------ ------------- ------------------
TOTO1  TABLE  8gz51m9hg5yuf data block
TOTO1  TABLE  8gz51m9hg5yuf data block
TOTO1  TABLE  8gz51m9hg5yuf segment header
TOTO1  TABLE  8gz51m9hg5yuf data block
```

3. Wait Analysis: How to get Class Name

P3 = class#, how do we get class name?

```
select * from v$event_name

where name = 'buffer busy waits'
```

```
NAME                      P1     P2     P3
------------------------- ------ ------ ------
buffer busy waits         file#  block# class#
```

```
select rownum n,ws.class
from v$waitstat;
```

```
  N CLASS
--- ------------------
  1 data block
  2 sort block
  3 save undo block
  4 segment header
  5 save undo header
  6 free list
  7 extent map
  8 1st level bmb
  9 2nd level bmb
 10 3rd level bmb
 11 bitmap block
 12 bitmap index block
 13 file header block
 14 unused
 15 system undo header
 16 system undo block
 17 undo header
 18 undo block
```

3. Wait Analysis: tx enqueues

Top 5 Timed Events

Event	Waits	Time (s)	Avg wait (ms)	%Total Call Time
enq: TX - row lock contention	**59**	**160**	**2714**	**41.8**
PL/SQL lock timer	4	117	29291	30.6
CPU time		28		7.2
buffer busy waits	1,217	18	15	4.7
log file parallel write	422	11	27	3.0

Who is waiting?
Who is blocking?
What is the SQL?
What is the row?

Not in AWR report

3. Wait Analysis: buffer busy waits

v$active_session_history

- Waiter
  - SESSION_ID
  - SESSION_SERIAL#
  - USER_ID
- Object
  - CURRENT_OBJ#
  - CURRENT_FILE#
  - CURRENT_BLOCK#
- SQL Waiting
  - SQL_ID
- Blocker
  - BLOCKING_SESSION
  - BLOCKING_SESSION_STATUS
  - BLOCKING_SESSION_SERIAL#
- Lock Type and Mode
  - Event = Type (name)
  - P1 = Type | Mode

3. Wait Analysis: tx enqueue waits

ash_enq.sql

```
select
        to_char(sample_time,'HH MI') st,
        substr(event,0,20)      lock_name,
        ash.session_id          waiter,
        mod(ash.p1,16)          lmode,
        ash.p2                  p2,
        ash.p3                  p3,
        o.object_name           object,
        o.object_type           otype,
        CURRENT_FILE#           filen,
        CURRENT_BLOCK#          blockn,
        ash.SQL_ID              waiting_sql,
        BLOCKING_SESSION        blocker
from
        v$active_session_history ash,
        all_objects o
where
        event like 'enq: T%'
    and o.object_id (+)= ash.CURRENT_OBJ#
/
```

3. Waits: enq TX 4

➢Mode 4, unique key

```
ST    EVENT                    SID  LM     P2   P3 OBJ   OTYPE  FN BLOCKN SQL_ID         BSID
----- ---------------------- --- --- ------ ---- ----- ----- --- ------ --
10:39 enq: TX - row lock c   141   4 655406 6672 -1            0      0 bjvx94vnxtxgv    158
10:39 enq: TX - row lock c   141   4 655406 6672 -1            0      0 bjvx94vnxtxgv    158
10:39 enq: TX - row lock c   141   4 655406 6672 -1            0      0 bjvx94vnxtxgv    158
10:39 enq: TX - row lock c   141   4 655406 6672 -1            0      0 bjvx94vnxtxgv    158
```

➢Mode 4, foreign key

```
ST    EVENT                    SID  LM     P2   P3 OBJ   OTYPE  FN BLOCKN SQL_ID         BSID
----- ---------------------- --- --- ------ ---- ----- ----- --- ------ --
10:41 enq: TX - row lock c   144   4 179681 7074 CHILD TABLE   1  60954 ahm7c9rupbz9r    1
10:41 enq: TX - row lock c   144   4 179681 7074 CHILD TABLE   1  60954 ahm7c9rupbz9r    1
10:41 enq: TX - row lock c   144   4 179681 7074 CHILD TABLE   1  60954 ahm7c9rupbz9r    1
```

➢Mode 4, bitmap

```
ST    EVENT                    SID  LM     P2   P3 OBJ   OTYPE  FN BLOCKN SQL_ID         BSID
----- ---------------------- --- --- ------ ---- ----- ----- --- ------ --
10:41 enq: TX - row lock c   143   4 966081 4598 I1    INDEX   0      0 azav296xxqcjx    144
10:41 enq: TX - row lock c   143   4 966081 4598 I1    INDEX   0      0 azav296xxqcjx    144
10:41 enq: TX - row lock c   143   4 966081 4598 I1    INDEX   0      0 azav296xxqcjx    144
10:41 enq: TX - row lock c   143   4 966081 4598 I1    INDEX   0      0 azav296xxqcjx    144
```

4. Top

- Procedure
  - ash_top_procedure.sql
- Session
  - ash_top_session.sql - wait, I/O and CPU time
- SQL
  - ash_top_sql.sql - wait, I/O and CPU time
  - ash_top_sql_w_top_obj.sql - with top OBJ per SQL

4. Top: PROCEDURE ash_top_procedure.sql

10.2.0.3 ASH added

PLSQL_ENTRY_OBJECT_ID
PLSQL_ENTRY_SUBPROGRAM_ID
PLSQL_OBJECT_ID
PLSQL_SUBPROGRAM_ID

ALL_PROCEDURES view

object_id = plsql_object_id

subprogram_id = plsql_subprogram_id

4. Top: PROCEDURE ash_top_procedure.sql

```
 COUNT(*) SQL_ID        calling_code
--------- ------------- ------------------------------------
        2 1xxksrhwtz3zf ORDERENTRY.NEWORDER  => DBMS_RANDOM.VALUE
        2 1xxksrhwtz3zf ORDERENTRY.NEWORDER  => DBMS_LOCK.SLEEP
       13 1xxksrhwtz3zf ORDERENTRY.NEWORDER
       16 0bzhqhhj9mpaa ORDERENTRY.NEWCUSTOMER
       45 41zu158rqf4kf ORDERENTRY.BROWSEANDUPDATEORDERS
```

4. Top: SESSION ash_top_session.sql

STATUS	SID	NAME	PROGRAM	CPU	WAITING	IO	TOTAL
DISCONNECTED	54	SYS	oracle@source (J000)	1227	540	152	1919
DISCONNECTED	57	SYS	oracle@source (J001)	725	160	18	903
DISCONNECTED	71	SYS	sqlplus@source (TNS V1-V3	535	60	36	631
DISCONNECTED	67	SYSTEM	LAB128.exe	187	182	148	517
CONNECTED	10	SYS	oracle@source (DBW0)	267	171	0	438
CONNECTED	11	SYS	oracle@source (LGWR)	10	357	0	367
DISCONNECTED	44	SYS	sqlplus@source (TNS V1-V3	103	158	15	276
CONNECTED	53	SYSTEM	JDBC Thin Client	129	33	0	162
CONNECTED	36	SYSMAN	OMS	6	114	0	120

4. Top: SQL ash_top_sql.sql

SQL_ID	PLAN_HASH	TYPE	CPU	WAIT	IO	TOTAL
fdtr6ds1nxm5r	1269593971	SELECT	9345	9	84277	93631
5aa7r665a07n6	3761618565	SELECT	82040	6	850	82896
2x8kgb0s9q1zh	0	PL/SQL EXE	75967	1575	1186	78728
4qraawku9303p	2256852101	SELECT	46995	45	3196	50236
3h7agx5ndadrf	1006614515	SELECT	2068	1	24914	26983
8p2cyq3gdgau5	2108428761	SELECT	22345	489	313	23147
7vpqr5zhjm08v	4089415459	SELECT	422	0	19655	20077
7sf51jrzq6y3c	485652470	INSERT	1727	1418	15471	18616
6cy7mc2kj0u4z	3041431021	SELECT	14272	0	0	14272

4. Top: SQL ash_top_sql_w_top_obj.sql

SQL_ID	SQLPLANHASH	AUD_ACT	CPU	WAIT	IO	TOTAL	PCT_IO	TOP_OBJ
dr1fpksws4nv9	3458999899	SELECT	3791	0	0	3791		
dc9fkz2t3b9p8	1909389838	SELECT	3685	0	5	3690	60	PS_F_ABS_EXPTN_CA
50spnwj8tdnnh	2849751558	SELECT	1981	57	0	2038		
fuzcbdt08xjcd	2618832091	SELECT	0	1864	0	1864		
30hzp85f3qtxj	3802366046	SELECT	568	0	921	1489	100	PS_TAX_BALANCE
c61wk6d7ssxxc	702141750	SELECT	117	6	1364	1487	98	PS_PAY_DEDUCTION
c3y6kdm1uzkb7	2395607145	SELECT	39	0	1375	1414	87	PS_PAY_CHECK
6wsy8rpd0bw26	252818247	SELECT	1363	0	14	1377	64	PS_F_TIALAST_HIST

5. I/O – lots you can do

I/O by

- SQL
- Object
- File & Table Space
- Parallel Query master and slaves
- By type of I/O
  - Db file sequential read
  - Db file scattered read
  - Direct path read
  - Direct path read temp
  - Direct path write
  - Direct path write temp

5. I/O : iosql.sql

I/O by SQL and top Objects for that sql

TCNT	SQL_ID	CNT	AAS	OBJN	OBJ	P1	TABLESPACE_NAME
11	3zqmt52j08cby	2	0	-1	0	7	AWR
		2	0	521	SYS_LOB0000000520C00	1	SYSTEM
		4	0	-1	0	1	SYSTEM
12	a3rgayp753z59	1	0	6342	WRH$_SQL_PLAN_PK	2	SYSAUX
		11	0	6339	WRH$_SQL_PLAN	2	SYSAUX
14	52tyrgvbph5fc	14	0	77259	77259	7	AWR

iosql.sql

```
select
      sum(cnt) over ( partition by io.sql_id order by sql_id ) tcnt,
      io.sql_id, io.cnt cnt, io.aas aas, io.objn objn, io.obj obj, io.p1 p1,
      f.tablespace_name tablespace_name
from
(
  select
       sql_id,
       count(*) cnt,
       round(count(*)/(&v_minutes*60),2) aas,
       CURRENT_OBJ# objn,
       nvl(o.object_name,decode(CURRENT_OBJ#,-1,0,CURRENT_OBJ#)) obj,
       o.object_type otype,
       ash.p1
   from v$active_session_history ash
       ,all_objects o
   where ( event like 'db file s%' or event like 'direct%' )
     and o.object_id (+)= ash.CURRENT_OBJ#
     and sample_time > sysdate - &v_minutes/(60*24)
   group by
       CURRENT_OBJ#,o.object_name ,o.object_type , ash.p1, sql_id
) io,
   dba_data_files f
where
   f.file_id = io.p1
Order by tcnt, io.sql_id, io.cnt
/
```

Extra : Latency

Can't get latency from: TIME_WAITED
So tempting

- latency_waitclassmetric.sql each 60 seconds WAIT_CLASS
  - V$WAITCLASSMETRIC
- latency_eventmetric.sql each 60 seconds EVENT
  - V$EVENTMETRIC
- latency_system_event.sql each hour EVENT
  - DBA_HIST_SYSTEM_EVENT

Latency: Last 60 seconds for I/O waitclass

latency_waitclassmetric.sql

```
select
         10*time_waited/nullif(wait_count,0) avg_io_ms
        -- convert centi-seconds to milliseconds
from   v$waitclassmetric  m
     where wait_class_id= 1740759767 --  User I/O
/
```

```
 AVG_IO_MS
----------
     2.032
```

latency: latency_eventmetric.sql

Last 60 seconds

NAME	TIME_WAITED	WAIT_COUNT	AVGMS
log file parallel write	3.856	12	3.213
log file sync	2.809	6	4.682
db file sequential read	0	0	
db file scattered read	0	0	
direct path read	0	0	
direct path read temp	0	0	
direct path write	0	0	
direct path write temp	0	0	

latency: last 60 seconds

latency_eventmetric.sql

```
select -- m.intsize_csec,
       n.name ,
       round(m.time_waited,3) time_waited,
       m.wait_count,
       round(10*m.time_waited/nullif(m.wait_count,0),3) avgms
from v$eventmetric m,
        v$event_name n
where m.event_id=n.event_id
  and n.name in (
                'db file sequential read',
                'db file scattered read',
                'direct path read',
                'direct path read temp',
                'direct path write',
                'direct path write temp',
                'log file sync',
                'log file parallel write'
```

v$event_metric only had event ids ☹
Join to v$event_name

I/O Events

Latency: each hour

```
BTIME                    AVG_MS
------------------------ ----------
01-MAY-13 21:46           6.133
01-MAY-13 21:57            .642
01-MAY-13 23:00            .199
02-MAY-13 00:00            .023
02-MAY-13 01:00            .031
02-MAY-13 02:00            .006
02-MAY-13 03:00            .017
02-MAY-13 04:00            .015
02-MAY-13 05:00            .013
02-MAY-13 06:00            .019
02-MAY-13 07:00            .017
```

latency_system_event.sql

DBA_HIST_SYSTEM_EVENT
DBA_HIST_SNAPSHOT
Input event name

END

DELPHIX

11.1g ASH extras

SQL Elapsed

SQL_EXEC_ID
SQL_EXEC_START

SQL Row Source

SQL_PLAN_LINE_ID
SQL_PLAN_OPERATION
SQL_PLAN_OPTIONS

Which instance requested block?

REMOTE_INSTANCE#

Recursive SQL

TOP_LEVEL_SQL_ID
TOP_LEVEL_SQL_OPCODE

Operation bit vector – non timed ops

IN_CONNECTION_MGMT
IN_PARSE
IN_HARD_PARSE
IN_SQL_EXECUTION
IN_PLSQL_EXECUTION
IN_PLSQL_RPC
IN_PLSQL_COMPILATION
IN_JAVA_EXECUTION
IN_BIND
IN_CURSOR_CLOSE

ETC

CURRENT_ROW#
EVENT#
QC_SESSION_SERIAL#
CONSUMER_GROUP_ID
FLAGS

11.2g ASH extras

Statistics

TM_DELTA_TIME
TM_DELTA_CPU_TIME
TM_DELTA_DB_TIME
DELTA_TIME
DELTA_READ_IO_REQUESTS
DELTA_WRITE_IO_REQUESTS
DELTA_READ_IO_BYTES
DELTA_WRITE_IO_BYTES
DELTA_INTERCONNECT_IO_BYTES
PGA_ALLOCATED
TEMP_SPACE_ALLOCATE

ASH Across Versions

When

Who

SQL

ASH instrumentation is exploding across the versions

Waits

Statistics

Blocker

I/O Sizes

- ash_io_sizes.sql - I/O sizes from ASH

I/O

- io.sql
- io_timestamps.sql
- ioag.sql
- ioag_orig.sql
- iodfpr.sql
- ionoobj.sql
- ionosql.sql
- ioobj.sql
- ioobjdec.sql
- iop.sql
- iosql.sql
- iotbs.sql
- iotbs1.sql
- iotx.sql
- jb_sql.txt

io.sql - basic I/O

```
EVENT              P1      P2 P3 OBJ                    OTYPE      SQL_ID
-------------- -----   ---- -- -------------------- ---------- -------------
db file sequent     1   22377  1 -1                              7wt7phk4xns75
db file sequent     2   75849  1 -1                              13fnb572x6z9j
db file sequent     2   78039  1 -1                              8tfvwyvfm5cjn
db file sequent     1   91095  1 -1                              d15cdr0zt3vtp
db file sequent     2  321005  1 WRH$_ACTIVE_SESSION_ TABLE PART 25a6sjj8zdbr7
db file scatter     2  331679  3 WRH$_ACTIVE_SESSION_ TABLE PART 25a6sjj8zdbr7
```

```
Select substr(event,0,15) event,
       ash.p1,
       ash.p2,
       ash.p3 p3,
       nvl(o.object_name,CURRENT_OBJ#) obj,
       o.object_type otype,
       --CURRENT_FILE# filen,
       --CURRENT_BLOCK# blockn,
       ash.SQL_ID
from v$active_session_history ash,
      all_objects o
where event like 'db file s%'
   and o.object_id (+)= ash.CURRENT_OBJ#
   and sample_time > sysdate - &minutes/(60*24)
Order by sample_time
/
```

io_pqo.sql

```
select
      ash.SQL_ID,
      QC_SESSION_ID qsid,
      count(*) cnt,
      count (distinct session_id) deg,
      nvl(o.object_name,to_char(CURRENT_OBJ#))  obj,
      o.object_type otype,
      decode(session_state, 'WAITING',event,'CPU') event
from   v$active_session_history ash,
       all_objects o
where  o.object_id (+)= ash.CURRENT_OBJ#
  and qc_session_id is not null
group by qc_session_id, sql_id, o.object_name,
       o.object_type, CURRENT_OBJ#, event, session_state
Order by q
```

SQL_ID	QSID	CNT	DEG	OBJ	OTYPE	EVENT
7p3jt75phub2d	144	386	4	WAIT_OBJECTS	TABLE	PX Deq Credit: send blkd
	144	4	3	WAIT_OBJECTS	TABLE	PX qref latch
	144	37	1	WAIT_OBJECTS	TABLE	db file sequential read
	144	3	2	WAIT_OBJECTS	TABLE	direct path read
	144	70	1	WAIT_OBJECTS	TABLE	CPU
	144	21	4	0		PX Deq Credit: send blkd
	144	12	4	0		db file sequential read

io_timestamps.sql: time range for I/O by Obj

CNT	DELTA	MNT	MXT	EVENT	OBJN	OTYPE	
15	.0382	04/13/05 18:23:16	04/13/05 19:18:15	direct path wri	232 I_PROCEDUREINFO1	INDEX	
15	.0143	04/13/05 18:24:50	04/13/05 18:45:27	direct path wri	222 PROCEDUREINFO$	TABLE	
23	.0117	04/13/05 18:25:43	04/13/05 18:42:36	db file scatter	6339 WRH$_SQL_PLAN	TABLE	
26	.0302	04/13/05 18:32:28	04/13/05 19:15:57	direct path wri	0		
59	.1930	04/13/05 18:22:43	04/13/05 23:00:35	db file sequent	-1		

```
select
       count(*) cnt,
       cast(max(sample_time) as date) - cast(min(sample_time) as date) delta,
       to_char(cast(min(sample_time) as date),'DD/YY/MM HH24:mi:ss') mnt,
       to_char(cast(max(sample_time) as date),'DD/YY/MM HH24:mi:ss') mxt,
       substr(event,0,15) event,
       CURRENT_OBJ#||' '||o.object_name objn,
       o.object_type otype
from v$active_session_history ash,
      all_objects o
where ( event like 'db file s%' or event like 'direct%' )
      and o.object_id (+)= ash.CURRENT_OBJ#
    --and sample_time > sysdate - &minutes/(60*24)
group by
       substr(event,0,15) ,
       CURRENT_OBJ#, o.object_name ,
       o.object_type
Order by cnt
/
```

ioag.sql - I/O by File# an Tablespace

CNT	AAS	EVENT	OBJ	P1	TABLESPACE_NAME
10	0	db file sequent	0	2	SYSAUX
12	0	db file sequent	WRH$_SEG_STAT	2	SYSAUX
13	0	db file scatter	WRH$_ACTIVE_SESSION_	2	SYSAUX
14	0	db file scatter	77259	7	AWR
22	0	db file sequent	0	1	SYSTEM
23	0	db file scatter	WRH$_SQL_PLAN	2	SYSAUX

ioag.sql - I/O tablespace

```
select &minutes f_minutes from dual;
select
       io.cnt cnt,
       io.aas aas,
       io.event event,
       substr(io.obj,1,20) obj,
       io.p1 p1,
       f.tablespace_name tablespace_name
from
(
  select
        count(*) cnt,
        round(count(*)/(&v_minutes*60),2) aas,
        substr(event,0,15) event,
        nvl(o.object_name,decode(CURRENT_OBJ#,-1,0,CURRENT_OBJ#)) obj,
        ash.p1,
        o.object_type otype
   from v$active_session_history ash,
        all_objects o
   where ( event like 'db file s%' or event like 'direct%' )
      and o.object_id (+)= ash.CURRENT_OBJ#
      and sample_time > sysdate - &v_minutes/(60*24)
   group by
       substr(event,0,15) ,
       CURRENT_OBJ#, o.object_name ,
       o.object_type ,
       ash.p1
) io,
  dba_data_files f
where
   f.file_id = io.p1
Order by io.cnt
/
```

ioag_orig.sql

```
CNT AAS OBJN                          OTYPE
--- --- ------------------------- --------------
 13   0 77448 WRH$_ACTIVE_SESSION_HI TABLE PARTITION
 15   0 222 PROCEDUREINFO$           TABLE
 23   0 6339 WRH$_SQL_PLAN           TABLE
 30   0 0
 34   0 232 I_PROCEDUREINFO1         INDEX
 37   0 -1
```

```
select cnt,
       --event,
       round(cnt/nullif(((
              to_date(beg,'DD/MM/YY HH24:MI:SS')-
              to_date(end,'DD/MM/YY HH24:MI:SS'))*
              24*60*60),0)
         ,2) aas,
       objn,
       otype
from (
select
       count(*) cnt,
       to_char(nvl(min(sample_time),sysdate),
              'DD/MM/YY HH24:MI:SS') end,
       to_char(nvl(max(sample_time),sysdate),
               'DD/MM/YY HH24:MI:SS') beg,
       substr(event,0,15) event,
       CURRENT_OBJ#||' '||o.object_name objn,
       o.object_type otype
from v$active_session_history ash,
     all_objects o
where ( event like 'db file s%' or event like 'direct% )
   and o.object_id (+)= ash.CURRENT_OBJ#
   and sample_time > sysdate - &minutes/(60*24)
group by
       substr(event,0,15) ,
       CURRENT_OBJ#, o.object_name ,
       o.object_type
)
Order by cnt
/
```

iodfpr.sql – db file parallel read

CNT	P1	P2	P3	OBJ	OTYPE	SQL_ID
1	1	127	127	WRH$_SQL_PLAN	TABLE	8jk220sm5y171
1	1	43	43	WRH$_PARAMETER_PK	INDEX PAR	03u0y15q547qq
1	1	127	127	WRH$_SQL_PLAN_PK	INDEX	bvvzxsyxxbnkp
1	1	9	9	WRH$_ROWCACHE_SUMMAR	INDEX PAR	2vvqz21zckj64
1	1	113	113	WRH$_LATCH_MISSES_SU	INDEX PAR	5c3gr4vusmgcn
1	1	117	117	WRH$_SQL_PLAN	TABLE	a3rgayp753z59

```
Select  count(*),
      ash.p1,
      ash.p2,
      to_char(ash.p3) p3 ,
      nvl(o.object_name,CURRENT_OBJ#) obj,
      o.object_type otype,
      ash.SQL_ID
from v$active_session_history ash,
     all_objects o
where event like 'db file p%'
  and o.object_id (+)= ash.CURRENT_OBJ#
  and sample_time > sysdate - &minutes/(60*24)
group by
      ash.p1, ash.p2, ash.p3 ,
      o.object_name,
      CURRENT_OBJ#,
      o.object_type ,
      ash.SQL_ID
order by count(*)
/
```

ionoobj.sql – I/O with no Object ID

```
COUNT(*)     P1     P2
-------  -----  ----------
      1      2       11522
      1      1       24517
      1      1       18536
      1    201       12422
      1    201       11264
      2    201       11780
      2    201       12416
      2    201       11776
```

```
select
      count(*),
      ash.p1,
      ash.p2
from v$active_session_history ash
where ( event like 'db file s%' or event like 'direct%' )
   and sample_time > sysdate - &minutes/(60*24)
   and session_state='WAITING'
   and ( current_obj# = -1 or current_obj#=0 )
group by
      ash.p1,
      ash.p2
Order by count(*)
/
```

ionosql.sql - I/O with no SQL_ID

```
       CNT USERNAME        OBJ
---------- --------------- ------------------
         1 SYS                           0
```

```
select
      count(*) cnt
      ,nvl(u.username,ash.user_id) username
      ,nvl(o.object_name,decode(CURRENT_OBJ#,-1,0,CURRENT_OBJ#)) obj
from v$active_session_history ash,
       all_objects o,
     dba_users u
where ( event like 'db file s%' or event like 'direct%' )
  and sample_time > sysdate - &minutes/(60*24)
  and session_state='WAITING'
  and (sql_id is null or sql_id ='')
     and o.object_id (+)= ash.CURRENT_OBJ#
  and u.user_id (+) = ash.user_id
group by
      ash.current_obj#
      ,o.object_name
      ,u.username
      ,ash.user_id
Order by count(*)
/
```

ioobj.sql - I/O by Object

```
CNT       AAS OBJN                          OTYPE
---------- -------------------------- ---------------
     15     0 222 PROCEDUREINFO$          TABLE
     17     0 77448 WRH$_ACTIVE_SESSION  TABLE PARTITION
     24     0 6339 WRH$_SQL_PLAN          TABLE
     32     0 0
     36     0 232 I_PROCEDUREINFO1       INDEX
     40     0 -1
```

```
select &minutes f_minutes from dual;
select
       count(*) cnt,
       round(count(*)/(&v_minutes*60),2) aas,
       CURRENT_OBJ#||' '||o.object_name objn,
       o.object_type otype
from v$active_session_history ash,
      all_objects o
where ( event like 'db file s%' or event like 'direct%' )
   and o.object_id (+)= ash.CURRENT_OBJ#
   and sample_time > sysdate - &v_minutes/(60*24)
   and session_state='WAITING'
group by
       CURRENT_OBJ#, o.object_name ,
       o.object_type
Order by count(*)
/
```

ioobjdec.sql - myextents

x

```
drop table myextents;
l
create table myextents as select * from dba_extents;
l
select
       count(*),
       ext.owner,
       ext.segment_name,
       ext.partition_name,
       ext.segment_type
from v$active_session_history ash,
     myextents ext
where ( event like 'db file s%' or event like 'direct%' )
   and sample_time > sysdate - &minutes/(60*24)
   and session_state='WAITING'
   and ( current_obj# = -1 or current_obj#=0 )
   and  ext.file_id(+)=ash.p1 and
        ash.p2 between  ext.block_id and ext.block_id + ext.blocks
group by
       ext.owner,
       ext.segment_name,
       ext.partition_name,
       ext.segment_type
Order by count(*)
/
```

iop.sql – I/O P1,P2,P3

```
NAME                           P1           P2       P3
---------------------- ------------ --------- ----------
read by other session           file#     block#     class#
db file sequential read         file#     block#     blocks
db file scattered read          file#     block#     blocks
db file parallel read           files     blocks   requests
direct path read          file number first dba block cnt
direct path read temp     file number first dba block cnt
direct path write         file number first dba block cnt
direct path write temp    file number first dba block cnt
```

```
select name,
       parameter1 p1,
       parameter2 p2,
       parameter3 p3
from v$event_name
where name in (
'db file sequential read',
'db file scattered read',
'db file parallel read',
'read by other session',
'direct path read',
'direct path write',
'direct path read temp',
'direct path write temp',
'direct path write (lob)'
)
;
```

iosql.sql – I/O by SQL_ID breakdown OBJ

TCNT	SQL_ID	CNT	AAS	OBJN	OBJ	P1	TABLESPACE_NAME
11	3zqmt52j08cby	2	0	-1	0	7	AWR
		2	0	521	SYS_LOB0000000520C00	1	SYSTEM
		4	0	-1	0	1	SYSTEM
12	a3rgayp753z59	1	0	6342	WRH$_SQL_PLAN_PK	2	SYSAUX
		11	0	6339	WRH$_SQL_PLAN	2	SYSAUX
14	52tyrgvbph5fc	14	0	77259	77259	7	AWR

iosql_aas.sql – I/O by SQL_ID break by OBJ

AAS	SQL_ID	CNT	PCT	OBJ	SUB_OBJ	OTYPE	EVENT	F#	TBSP	CONTENTS
.000	a3rgayp753z59	1	8	WRH$_SQL_PLAN_PK		INDEX	sequentia	2	SYSAUX	PERMANENT
.000	a3rgayp753z59	11	92	WRH$_SQL_PLAN		TABLE	scattered	2	SYSAUX	PERMANENT
.000	2hgxq2u3v0qjc	13	100	WRH$_FILESTATXS_PK	WRH$_FILES	INDEX PART	sequentia	2	SYSAUX	PERMANENT
.000	2whm2vvjb98k7	6	22				scattered	2	SYSAUX	PERMANENT
		21	78				sequentia	2	SYSAUX	PERMANENT

iosql.sql

```
TCNT SQL_ID        CNT AAS OBJN OBJ                    P1 TABLESPACE_NAME
----- ---------- ---- ---- --- -------------------- -- --------------
11 3zqmt52j08cby    2    0   -1 0                    7 AWR
                    2    0  521 SYS_LOB0000000520C00 1 SYSTEM
                    4    0   -1 0                    1 SYSTEM
12 a3rgayp753z59    1    0 6342 WRH$_SQL_PLAN_PK     2 SYSAUX
                   11    0 6339 WRH$_SQL_PLAN        2 SYSAUX
14 52tyrgvbph5fc   14    077259 77259                7 AWR
```

```
select
      sum(cnt) over ( partition by io.sql_id order by sql_id ) tcnt,
      io.sql_id, io.cnt cnt, io.aas aas, io.objn objn, io.obj obj, io.p1 p1,
      f.tablespace_name tablespace_name
from
(
  select
       sql_id,
       count(*) cnt,
       round(count(*)/(&v_minutes*60),2) aas,
       CURRENT_OBJ# objn,
       nvl(o.object_name,decode(CURRENT_OBJ#,-1,0,CURRENT_OBJ#)) obj,
```

iotbs1.sql

```
    CNT EVENT           TABLESPACE_NAME
------- --------------- -------------
     13 direct path wri AWR
     18 db file sequent AWR
     38 db file scatter AWR
     46 db file sequent SYSTEM
     58 db file scatter SYSAUX
     83 db file sequent SYSAUX
```

```
select
      tf.cnt,
      tf.event,
      f.tablespace_name
from (
     select
           count(*) cnt,
           substr(event,0,15) event,
           ash.p1 p1
      from   v$active_session_history ash
      where ( event like 'db file s%' or event like 'direct%' )
           and sample_time > sysdate - &minutes/(60*24)
      group by
           substr(event,0,15) ,
           ash.p1
     ) tf,
       dba_data_files f
where
   f.file_id = tf.p1
Order by tf.cnt
/
```

iotx.sql

```
SQL_ID         XID
-------------- ----------------
fty6ptn572tt8
fzxxhn9yjhtqm 0900030011070000
g3tvrzsb8rb7j
g48azr9sj7ud2
g7rjqg0t81tv6 05000E006E060000
gh2hf1fqppaxr
gq841w157yvsp
```

```
Select
       ash.sql_id
      ,ash.xid
from v$active_session_history ash
where ( event like 'db file s%' or event like 'direct%' )
   and sample_time > sysdate - &minutes/(60*24)
   and session_state='WAITING'
order by sql_id, xid
/
```

DELPHIX

io_seq.sql

x

io_scat.sql

EVENT	P3	OBJN	OTYPE	FILEN	BLOCKN	SQL_ID
db file scattered read	5	5904 WRI$_ADV_USAGE	TABLE	2	3691	7aum6ufgy2xjq
db file scattered read	31	77403 WRH$_LATCH_PK	INDEX PART	2	228293	1zkmvr5gjshqk
db file scattered read	8	78168 WRH$_SQLSTAT	TABLE PART	2	360352	040bmjsqca000
db file scattered read	128	-1		0	0	2whm2vvjb98k7
db file scattered read	124	-1		0	0	2whm2vvjb98k7
db file scattered read	128	-1		0	0	2whm2vvjb98k7
db file scattered read	4	-1		0	0	2whm2vvjb98k7
db file scattered read	6	-1		0	0	2whm2vvjb98k7
db file scattered read	128	-1		0	0	2whm2vvjb98k7

```
Select event,
       ash.p3,
       CURRENT_OBJ#||' '||o.object_name objn,
       o.object_type otype,
       CURRENT_FILE# filen,
       CURRENT_BLOCK# blockn,
       ash.SQL_ID
from v$active_session_history ash,
     all_objects o
where event like 'db file scattered read'
  and o.object_id (+)= ash.CURRENT_OBJ#
Order by sample_time;
```

jb_sql.txt

x

cpu_consumed_verses_cpuwait.sql

```
QL> @cpu_consumed_verses_cpuwait.sql

 CPU_TOTAL     CPU_OS  CPU_ORA CPU_ORA_WAIT   COMMIT     READIO     WAIT
---------- ---------- ---------- ------------ ---------- ---------- ----------
      .088    .086         .002     0            0   0    .002
```

Exporting AWR

http://gavinsoorma.com/2009/07/25/exporting-and-importing-awr-snapshot-data/

SQL> CREATE DIRECTORY AWR_DATA AS '/u01/oracle/';

SQL> @?/rdbms/admin/awrextr.sql

Enter value for directory_name: AWR_DATA
Using the dump directory: AWR_DATA
Enter value for file_name: awrexp

Importing AWR : create tablespace

```
create BIGFILE tablespace AWR
   datafile '/home/oracle/oradata/${ORACLE_SID}/awr_01.dbf'
   size 1G
   NOLOGGING ONLINE
   PERMANENT
   EXTENT MANAGEMENT LOCAL
   AUTOALLOCATE
   SEGMENT SPACE MANAGEMENT AUTO ;
 alter tablespace AWR autoextend on
     next 200m maxsize unlimited;
```

DELPHIX

Importing AWR: drop/recreate user

```
-- create tablespace AWR datafile '&DATAFILE' size 200M;
-- drop directory AWR_DMP;
   create directory AWR_DMP AS '&AWR_DMP_LOCATION';
-- drop every time
   drop user awr_stage cascade;
   create user awr_stage
            identified by awr_stage
            default tablespace awr
             temporary tablespace temp;
   grant  connect to awr_stage;
   alter    user awr_stage quota unlimited on awr;
   alter    user awr_stage temporary tablespace temp;
```

Importing AWR : import dmp file

```
-- Import dump file into stage
 begin
     dbms_swrf_internal.awr_load(
       schname  => 'AWR_STAGE',
        dmpfile  => '&DMP_FILE', -- file w/o .dmp
         dmpdir   => 'AWR_DMP');
   end;
/
-- change dbid, give it a new #
   def dbid=&DBID;
   @awr_change_dbid
```

Importing AWR : move from Stage, change DBID

```
-- move from stage to SYSAUX
def schema_name='AWR_STAGE'
select  '&schema_name' from dual;
variable schname varchar2(30);
   begin
     :schname := '&schema_name';
     dbms_swrf_internal.move_to_awr(schname => :schname);
   end;
/

-- check new DBID
   col host_name for a30
   select distinct dbid, version, db_name, instance_name, host_name from
   dba_hist_database_instance;
```

- Wait Analysis
  - ash_bbw.sql - buffer busy
  - ash_enq.sql - enqueue
- Load Charts
  - V$ACTIVE_SESSION_HISTORY
    - ash_graph_ash.sql – basic
  - DBA_HIST_ACTIVE_SESS_HISTORY
    - ash_graph_ash_histash.sql –
    - ash_graph_histash_by_dbid.sql - input DBID
    - ash_graph_histash_by_dbid_program.sql - input DBID and PROGRAM
    - ash_graph_histash_by_dbid_sqlid.sql - input DBID and a SQL_ID
- SQL Elapsed
  - ash_sql_elapsed.sql - longest running SQL
  - ash_sql_elapsed_hist.sql – “” with histogram of execution times
  - ash_sql_elapsed_hist_longestid.sql - “” execution id of longest running query
- Top
  - ash_top_procedure.sql
  - ash_top_session.sql - wait, I/O and CPU time
  - ash_top_sql.sql - wait, I/O and CPU time
  - ash_top_sql_w_top_obj.sql -I/O and CPU time, include top object for I/O waits
- Latency
  - latency_eventmetric.sql - wait event latency from V$EVENTMETRIC, ie last 60 seconds
  - latency_system_event.sql - wait event latency from DBA_HIST_SYSTEM_EVENT
  - latency_waitclassmetric.sql - User I/O latency from V$WAITCLASSMETRIC, ie over last 60 seconds
- I/O Sizes
  - ash_io_sizes.sql - I/O sizes from ASH

Cool stuff

- Querying across RAC nodes
- Extrapolating wait times from wait counts

act.sql- activity

```
@act
Analysis Begin Time :   2007-07-24 11:04:48
Analysis End   Time :   2007-07-24 11:19:45
Start time, mins ago:           15
Request Duration    :           15
Collections         :          528
Data Values         :         3327
Elapsed Time:  15 mins

WAIT_EVENT                          CNT    % Active Ave_Act_Sess
----------------------------------- ---------- ------------
latch free                           10         .3          .02
log buffer space                     13        .39          .02
buffer busy waits                    14        .42          .03
db file scattered read               15        .45          .03
library cache pin                    78       2.34          .15
log file sync                       213       6.40          .40
ON CPU                              726      21.82         1.38
enqueue                             855      25.70         1.62
db file sequential read            1399      42.05         2.65
                                                      ------------
sum                                                         6.30
```