

Troy's Tuning Tips

Troy Ligon

Topics:

- ⇒ The REAL Execution Plan
- ⇒ Histograms
- ⇒ Child Cursors
- ⇒ Automatic Memory Management
- ⇒ SQLPlus Startup Slow

Topics:

⇒ The REAL Execution Plan

⇒ Histograms

⇒ Child Cursors

⇒ Automatic Memory Management

⇒ SQLPlus Startup Slow

The REAL Execution Plan

Execution Plan from Interactive SQL:

Usage:

```
SET AUTOT[RACE] {OFF | ON | TRACE[ONLY]}  
[EXP[LAIN]] [STAT[ISTICS]]
```

The REAL Execution Plan

Execution Plan from Interactive SQL:

```
set autotrace on explain
select sysdate from dual;
```

SYSDATE

2010-08-26 09:41:01

Execution Plan

Plan hash value: 1546270724

Id	Operation	Name	Rows	Cost (%CPU)	Time
0	SELECT STATEMENT		1	2 (0)	00:00:01
1	FAST DUAL		1	2 (0)	00:00:01

The REAL Execution Plan

Execution Plan from Interactive SQL:

```
set autotrace traceonly statistics
```

```
select sysdate from dual;
```

Statistics

```
-----  
0 recursive calls  
0 db block gets  
0 consistent gets  
0 physical reads  
0 redo size  
526 bytes sent via SQL*Net to client  
520 bytes received via SQL*Net from client  
2 SQL*Net roundtrips to/from client  
0 sorts (memory)  
0 sorts (disk)  
1 rows processed
```

The REAL Execution Plan

Execution Plan from Interactive SQL:

```
explain plan for &SQL_TEXT ;
```

```
select * from table(  
  dbms_xplan.display  
);
```

The REAL Execution Plan

Execution Plan from Interactive SQL:

```
explain plan for &SQL_TEXT ;
```

```
Enter value for sql_text: select sid,serial# from v$session
```

Explained.

```
select * from table(dbms_xplan.display);
```

Plan hash value: 2334513008

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		675	20250	1 (100)	00:00:01
1	NESTED LOOPS		675	20250	1 (100)	00:00:01
2	NESTED LOOPS		675	8775	0 (0)	00:00:01
3	FIXED TABLE FULL	X\$KSLWT	675	6075	0 (0)	00:00:01
* 4	FIXED TABLE FIXED INDEX	X\$KSLED (ind:2)	1	4	0 (0)	00:00:01
* 5	FIXED TABLE FIXED INDEX	X\$KSUSE (ind:1)	1	17	0 (0)	00:00:01

The REAL Execution Plan

Execution Plan from SQL in SGA:

```
select * from table(  
  dbms_xplan.display_cursor(  
    '&sql_id',&child_number,'TYPICAL')  
  );
```

The REAL Execution Plan

Execution Plan from SQL in SGA:

```
connect sys
select /*twl*/ sid from v$session;
connect system
select /*twl*/ sid from v$session;
```

```
select sql_id,child_number,sql_text
from v$sql where sql_text like '%twl%';
```

SQL_ID	CHILD_NUMBER	SQL_TEXT
69qw2qcgmvmtx	0	select /*twl*/ sid from v\$session
69qw2qcgmvmtx	1	select /*twl*/ sid from v\$session

The REAL Execution Plan

Execution Plan from SQL in SGA:

```
connect sys  
select sysdate from dual;  
connect system  
select sysdate from dual;
```

```
select last_active_time,sql_id,child_number,  
substr(sql_text,1,50) sql_text      from v$sql  
where last_active_time>sysdate-(10/(24*60*60))  
order by 2,3  
/
```

The REAL Execution Plan

Execution Plan from SQL in SGA:

LAST_ACTIVE_TIME	SQL_ID	CHILD_NUMBER
-----	-----	-----
SQL_TEXT		

2010-08-26 10:58:05	7h35uxf5uhmm1	0
select sysdate from dual		
2010-08-26 10:58:05	7h35uxf5uhmm1	1
select sysdate from dual		

The REAL Execution Plan

Execution Plan from SQL in SGA:

@twl_find_all_plans_from_vdollar.sql

Enter value for sql_id: 69qw2qcgmvmtx

```
/* SYS      2010-08-24 22:09:00 cost=1 */ select * from table(
DBMS_XPLAN.DISPLAY_CURSOR('69qw2qcgmvmtx','0','TYPICAL')) ;
```

SQL_ID 69qw2qcgmvmtx, child number 0

select /*twl*/ sid from v\$session

Plan hash value: 2334513008

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				1 (100)	
1	NESTED LOOPS		675	18225	1 (100)	00:00:01
2	NESTED LOOPS		675	8775	0 (0)	
3	FIXED TABLE FULL	X\$KSLWT	675	6075	0 (0)	
* 4	FIXED TABLE FIXED INDEX	X\$KSLED (ind:2)	1	4	0 (0)	
* 5	FIXED TABLE FIXED INDEX	X\$KSUSE (ind:1)	1	14	0 (0)	

The REAL Execution Plan

Execution Plan from SQL in AWR:

```
select * from table(  
  dbms_xplan.display_awr(  
    '&sql_id',&plan_hash_value,  
    &dbid,'TYPICAL')  
);
```

The REAL Execution Plan

Execution Plan from SQL in AWR:

@twl_find_all_plans_from_AWR.sql

Enter value for sql_id: 0fxuy3rqrz1k4

```
/* ARCUSTAX 2010-04-23 03:11:05 cost=          1 */ select * from table(
DBMS_XPLAN.DISPLAY_AWR('0fxuy3rqrz1k4','2343943101','1652853939','TYPICAL'));
/* USTAX      2010-03-27 06:03:58 cost=          1 */ select * from table(
DBMS_XPLAN.DISPLAY_AWR('0fxuy3rqrz1k4','4234951013','1652853939','TYPICAL'));
/* USTRAIN    2010-03-27 06:03:31 cost=          1 */ select * from table(
DBMS_XPLAN.DISPLAY_AWR('0fxuy3rqrz1k4','1849462728','1652853939','TYPICAL'));
```

SQL_ID 0fxuy3rqrz1k4

```
select all dm_group.r_object_id from dm_group_sp  dm_group where
((dm_group.group_name="SYS_B_0"))
```

Plan hash value: 4234951013

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				1 (100)	
1	TABLE ACCESS BY INDEX ROWID	DM_GROUP_S	1	40	1 (0)	00:00:01
2	INDEX RANGE SCAN	D_1F00D60B80000018	1		1 (0)	00:00:01

The REAL Execution Plan

Bind Values from SQL in AWR:

@twl_show_binds_from_AWR.sql

```
select SNAP_ID, POSITION, LAST_CAPTURED, NAME, VALUE_STRING
from DBA_HIST_SQLBIND where SQL_ID='&sql_id' and rownum < 20
order by SNAP_ID desc, POSITION
/
```

Enter value for sql_id: 0fxuy3rqrz1k4

SNAP_ID	POSITION	LAST_CAPTURED	NAME	VALUE_STRING
2106	1	2010-07-02 08:41:38	:SYS_B_0	reg_0b00d60b80038b74
2106	1	2010-07-02 08:07:42	:SYS_B_0	imp_tls_au_delta_admin
2106	1	2010-07-02 06:05:02	:SYS_B_0	imp_tls_au_acl
2104	1	2010-07-02 06:54:53	:SYS_B_0	imp_tls_security_admin
2104	1	2010-07-02 06:06:10	:SYS_B_0	imp_tls_au_delta_admin
2104	1	2010-07-02 06:05:02	:SYS_B_0	imp_tls_au_acl

The REAL Execution Plan

Bind Values from SQL in SGA:

@twl_show_binds_from_vdollar.sql

```
select POSITION, LAST_CAPTURED, NAME, VALUE_STRING from gv$sql_bind_capture
where sql_id='&sql_id' and child_number=&child_number order by 1,2
/
```

Enter value for sql_id: 0fxuy3rqrz1k4

Enter value for child_number: 0

POSITION	LAST_CAPTURED	NAME	VALUE_STRING
1	2010-08-25 21:39:00	:SYS_B_0	reg_0b00d60b81bf2018
1	2010-08-25 23:08:32	:SYS_B_0	imp_tls_security_admin

The REAL Execution Plan

Execution Plan from SQL Plan Baselines:

```
select plan_table_output  
from table(  
  dbms_xplan.display_sql_plan_baseline(  
    '&sql_handle','&plan_name','TYPICAL')  
  );
```

The REAL Execution Plan

Execution Plan from SQL PLAN BASELINES:

```
@twl_find_all_plans_from_baselines.sql
/* USTAX      24-AUG-10 03.12.14.000000 PM cost=          5
enabled=YES  accepted=YES  fixed=NO */
select * from table(DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE
('SYS_SQL_02b0c738ac895e79','SYS_SQL_PLAN_ac895e79fa08e7e6',
'TYPICAL'));

/* USTAX      24-AUG-10 03.18.08.000000 PM cost=          1
enabled=YES  accepted=YES  fixed=NO */
select * from table(DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE
('SYS_SQL_e7f56f3aeacd2432','SYS_SQL_PLAN_eacd24321fe77486',
'TYPICAL'));
```

The REAL Execution Plan

Execution Plan from SQL PLAN BASELINES :

```
select * from table(DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE  
( 'SYS_SQL_02b0c738ac895e79', 'SYS_SQL_PLAN_ac895e79fa08e7e6', 'TYPICAL' ));
```

SQL handle: SYS_SQL_02b0c738ac895e79

SQL text: select distinct dm_document.r_object_id from dm_document_sp dm_document
where ((dm_document.i_chronicle_id="SYS_B_00")) and
dm_document.i_is_deleted = "SYS_B_01" and ((exists (select "SYS_B_02"
from dm_acl_s ACL_S0, dm_acl_r ACL_R where ACL_S0.r_object_id =
ACL_R.r_object_id and dm_document.acl_domain = ACL_S0.owner_name and
dm_document.acl_name = ACL_S0.object_name and ((ACL_R.r_accessor_name
in ("SYS_B_03","SYS_B_04") or (ACL_R.r_is_group = "SYS_B_05" and
(ACL_R.r_accessor_name in ("SYS_B_06","SYS_B_07","SYS_B_08","SYS_B_0
9","SYS_B_10","SYS_B_11","SYS_B_12","SYS_B_13","SYS_B_14","SYS_B_1
5","SYS_B_16","SYS_B_17","SYS_B_18","SYS_B_19","SYS_B_20")))) or
(ACL_R.r_accessor_name = "SYS_B_21")) and ((ACL_R.r_permit_type =
"SYS_B_22" or ACL_R.r_permit_type is null) and
(((ACL_R.r_accessor_permit >= "SYS_B_23")))))))) and
dm_document.r_object_type = "SYS_B_24"

The REAL Execution Plan

Execution Plan from SQL PLAN BASELINES :

Plan name: SYS_SQL_PLAN_ac895e79fa08e7e6

Enabled: YES Fixed: NO Accepted: YES Origin: MANUAL-LOAD

Plan hash value: 2634024941

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	205	5 (20)	00:00:01
1	SORT UNIQUE		1	205	5 (20)	00:00:01
2	NESTED LOOPS					
3	NESTED LOOPS		1	205	4 (0)	00:00:01
4	NESTED LOOPS		1	166	3 (0)	00:00:01
5	NESTED LOOPS		1	112	2 (0)	00:00:01
* 6	TABLE ACCESS BY INDEX ROWID	DM_SYSOBJECT_S	1	95	1 (0)	00:00:01
* 7	INDEX RANGE SCAN	D_1F00D60B8000000E	1		1 (0)	00:00:01
* 8	INDEX UNIQUE SCAN	D_1F00D60B8000013F	1	17	1 (0)	00:00:01
9	TABLE ACCESS BY INDEX ROWID	DM_ACL_S	1	54	1 (0)	00:00:01
* 10	INDEX UNIQUE SCAN	D_1F00D60B80000103	1		1 (0)	00:00:01
* 11	INDEX RANGE SCAN	D_1F00D60B80000102	5		1 (0)	00:00:01
* 12	TABLE ACCESS BY INDEX ROWID	DM_ACL_R	1	39	1 (0)	00:00:01

The REAL Execution Plan

I have included the following sqls used above in the session download:

twl_find_all_plans_from_vdollar.sql
twl_find_all_plans_from_AWR.sql
twl_find_all_plans_from_baselines.sql

twl_show_binds_from_vdollar.sql
twl_show_binds_from_AWR.sql

unpack_clob.sql

Topics:

⇒ The REAL Execution Plan

⇒ Histograms

⇒ Child Cursors

⇒ Automatic Memory Management

⇒ SQLPlus Startup Slow

HISTOGRAMS

What is a Histogram?

How do I create one?

How do I see what Histograms exist?

How do I know if they are helping?

HISTOGRAMS

Histograms give the optimizer information about the skew of the data so it can make different decisions based on different bind values.

HISTOGRAMS

-- How to Create Histograms:

```
exec sys.dbms_stats.gather_table_stats( ownname=>'SCOTT', degree=>6,  
tabname=>'EMP', method_opt=>'FOR COLUMNS LAST_NAME size 254',  
cascade => TRUE);
```

```
exec sys.dbms_stats.gather_table_stats( ownname=>'SCOTT', degree=>6,  
tabname=>'DEPT', method_opt=>'FOR COLUMNS DEPT_NO size 100',  
cascade=> TRUE);
```

```
exec sys.dbms_stats.gather_table_stats( ownname=>'SCOTT', degree=>6,  
tabname=>'DEPT', method_opt=>'FOR ALL INDEXED COLUMNS size  
skewonly', cascade=> TRUE);
```

HISTOGRAMS

```
-- What Histograms Exist?  
col owner          format a10  
col table_name     format a20  
col column_name    format a30  
select owner,table_name,column_name,  
       max(endpoint_number),count(*) buckets  
from dba_histograms  
where endpoint_number>1 and owner in  
      ('USTAX','ARCUSTAX','USTRAIN')  
group by owner,table_name,column_name  
/
```

HISTOGRAMS

OWNER	TABLE_NAME	COLUMN_NAME	MAX(ENDPOINT_NUMBER)	BUCKETS
USTAX	DM_ACL_S	OWNER_NAME	75	8
USTAX	DM_ACL_S	OBJECT_NAME	75	74
USTAX	DM_FOLDER_R	R_FOLDER_PATH	254	225
USTAX	DM_SYSOBJECT_R	I_FOLDER_ID	254	236
USTAX	DM_SYSOBJECT_R	R_VERSION_LABEL	100	13
USTAX	DM_SYSOBJECT_S	I_CHRONICLE_ID	254	253
ARCUSTAX	DM_ACL_S	OWNER_NAME	75	71
ARCUSTAX	DM_ACL_S	OBJECT_NAME	75	74
ARCUSTAX	DM_FOLDER_R	R_FOLDER_PATH	254	79
ARCUSTAX	DM_SYSOBJECT_R	I_FOLDER_ID	254	94
ARCUSTAX	DM_SYSOBJECT_R	R_VERSION_LABEL	100	17

HISTOGRAMS

Are these histograms helping? Show all sql_id's with multiple plans:

```
col sqltext format a100
col object_owner format a10
break on sql_id
select sql_id,object_owner,min(cost),max(cost),sum(cost),count(*),sqltext
from ( select distinct p.sql_id,p.object_owner,p.plan_hash_value,
unpack_clob(t.sql_text,100) sqltext,
(select cost from dba_hist_sql_plan c where p.dbid=c.dbid and p.sql_id=c.sql_id
and p.plan_hash_value=c.plan_hash_value and c.id=0) cost
  from dba_hist_sql_plan p, dba_hist_sqltext t
  where p.object_owner in ('USTAX','ARCUSTAX','USTRAIN')
  and p.sql_id=t.sql_id(+)
  and unpack_clob(t.sql_text,15)<>'/* SQL Analyze(' )
where object_owner is not null
group by sql_id,object_owner,sqltext
having count(*) > 1 order by 1,2;
```

HISTOGRAMS

SQL_ID	OBJECT_OWN	MIN(COST)	MAX(COST)	SUM(COST)	COUNT(*)
03pcdt6v6n88r	USTAX	13	13	26	2
0ct6cabbk22jm	USTAX	10	11	21	2
0jvpnnj87bn2m	ARCUSTAX	5	6	17	3
	USTAX	5	6	11	2
0mvsdq6ar9160	USTAX	1270	52734	54004	2
0q7tdjbd5z88f	USTAX	183	1036	1219	2
0uszypunyzrb1	USTAX	10	11	31	3
289f72abzvz8	USTAX	5	48	58	3
2u74nxv4xwt3h	ARCUSTAX	18	39	115	5
	USTAX	18	20	58	3
2x10u27c67464	USTAX	3	4567	4574	3
38zk3jrk064qk	USTAX	13	13	39	3
3bf9dv4ym5qpx	USTAX	379	12572	34407	5
3nmjmn4x3u7kk	USTAX	5302	8635	13937	2
56bamkhbbbtkg	USTAX	9	168	177	2
57cf4qqc3vvfu	USTAX	17381	228512	245893	2
7bjdpdjy2r299	USTAX	4240	213454	217694	2

HISTOGRAMS

Look at the plans from above sql_id's:

```
-- twl_find_all_plans_from_AWR.sql
select distinct
  /*'||rpad(p.object_owner,8)||' '||p.timestamp||' cost=||
  to_char((select c.cost from dba_hist_sql_plan c
    where p.dbid=c.dbid and p.sql_id=c.sql_id
    and p.plan_hash_value=c.plan_hash_value and c.id=0),'99999999')||' */'||
  'select * from table(DBMS_XPLAN.DISPLAY_AWR(''||p.sql_id||'', ''||
  p.plan_hash_value||'', ''||p.dbid||'', 'TYPICAL'));' cmd
from dba_hist_sql_plan p
where p.sql_id in ('&sql_id')
and p.object_owner is not null
order by 1
/
```

HISTOGRAMS

sqlplus "/ as sysdba" @twl_find_all_plans_from_AWR.sql
Enter value for sql_id: 2u74nxv4xwt3h

CMD

```
-----  
/* ARCUSTAX 2010-07-29 11:52:13 cost=      39 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','3839802276','1483500327','TYPICAL'));  
/* ARCUSTAX 2010-08-02 10:18:07 cost=      20 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','2837950775','1483500327','TYPICAL'));  
/* ARCUSTAX 2010-08-02 10:26:14 cost=      20 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','3289804744','1483500327','TYPICAL'));  
/* ARCUSTAX 2010-08-10 12:46:17 cost=      18 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','2356683203','1483500327','TYPICAL'));  
/* ARCUSTAX 2010-08-11 09:55:31 cost=      18 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','1324208847','1483500327','TYPICAL'));  
/* USTAX      2010-07-29 09:56:42 cost=      20 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','560044146','1483500327','TYPICAL'));  
/* USTAX      2010-07-30 13:20:17 cost=      20 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','1412673116','1483500327','TYPICAL'));  
/* USTAX      2010-08-05 00:10:05 cost=      18 */ select * from  
table(DBMS_XPLAN.DISPLAY_AWR('2u74nxv4xwt3h','2433123436','1483500327','TYPICAL'));
```


Topics:

⇒ The REAL Execution Plan

⇒ Histograms

⇒ Child Cursors

⇒ Automatic Memory Management

⇒ SQLPlus Startup Slow

CHILD CURSORS

Example from earlier:

```
connect sys
select /*twl*/ sid from v$session;
connect system
select /*twl*/ sid from v$session;
```

```
select sql_id,child_number,sql_text from v$sql where sql_text like
'%twl%';
```

SQL_ID	CHILD_NUMBER	SQL_TEXT
-----	-----	-----
69qw2qcgmvmtx	0	select /*twl*/ sid from v\$session
69qw2qcgmvmtx	1	select /*twl*/ sid from v\$session

CHILD CURSORS

Why are these cursors not shared? Look in v\$sql_shared_cursor:

UNBOUND_CURSOR,
SQL_TYPE_MISMATCH,
OPTIMIZER_MISMATCH,
OUTLINE_MISMATCH,
STATS_ROW_MISMATCH,
LITERAL_MISMATCH,
FORCE_HARD_PARSE,
EXPLAIN_PLAN_CURSOR,
BUFFERED_DML_MISMATCH,
PDML_ENV_MISMATCH,
INST_DRTL_MISMATCH,
SLAVE_QC_MISMATCH,
TYPECHECK_MISMATCH,
AUTH_CHECK_MISMATCH,
BIND_MISMATCH,
DESCRIBE_MISMATCH,
LANGUAGE_MISMATCH,
TRANSLATION_MISMATCH,
ROW_LEVEL_SEC_MISMATCH,
INSUFF_PRIVS,
INSUFF_PRIVS_REM,
REMOTE_TRANS_MISMATCH,
LOGMINER_SESSION_MISMATCH,

INCOMP_LTRL_MISMATCH
OVERLAP_TIME_MISMATCH,
EDITION_MISMATCH,
MV_QUERY_GEN_MISMATCH,
USER_BIND_PEEK_MISMATCH,
TYPCHK_DEP_MISMATCH,
NO_TRIGGER_MISMATCH,
FLASHBACK_CURSOR,
ANYDATA_TRANSFORMATION,
INCOMPLETE_CURSOR,
TOP_LEVEL_RPI_CURSOR,
DIFFERENT_LONG_LENGTH,
LOGICAL_STANDBY_APPLY,
DIFF_CALL_DURN,
BIND_UACS_DIFF,
PLSQL_CMP_SWITCHS_DIFF,
CURSOR_PARTS_MISMATCH,
STB_OBJECT_MISMATCH,
CROSSEDITION_TRIGGER_MISMATCH,
PQ_SLAVE_MISMATCH,
TOP_LEVEL_DDL_MISMATCH,
MULTI_PX_MISMATCH,
BIND_PEEKED_PQ_MISMATCH,

MV_REWRITE_MISMATCH,
ROLL_INVALID_MISMATCH,
OPTIMIZER_MODE_MISMATCH
PX_MISMATCH,
MV_STALEOBJ_MISMATCH,
FLASHBACK_TABLE_MISMATCH,
LITREP_COMP_MISMATCH,
PLSQL_DEBUG,
LOAD_OPTIMIZER_STATS,
ACL_MISMATCH,
FLASHBACK_ARCHIVE_MISMATCH,
LOCK_USER_SCHEMA_FAILED,
REMOTE_MAPPING_MISMATCH,
LOAD_RUNTIME_HEAP_FAILED,
HASH_MATCH_FAILED

CHILD CURSORS

@twl_why_not_shared_by_count.sql

Enter value for childcnt: 1

A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
A21	A22	A23	A24	A25	A26	A27	A28	A29	A30
A31	A32	A33	A34	A35	A36	A37	A38	A39	A40
A41	A42	A43	A44	A45	A46	A47	A48	A49	A50
A51	A52	A53	A54	A55	A56	A57	A58	A59	A60
A61									
0	9	6	0	0	0	0	0	0	0
0	0	0	11219	2895	0	7	11213	7	0
0	0	0	370	0	0	0	0	0	0
0	0	130	3	0	0	3	73	0	0
0	0	14	0	1	0	0	379	521	1
0	0	0	0	5	0	0	0	0	0
45109									

CHILD CURSORS

@twl_why_not_shared_by_childcnt.sql

Enter value for childcnt: 3000

[illegible]

CHILD CURSORS

Why do I care about all these children?

CHILD CURSORS

Why do I care about all these children?

Library Latch Contention

cursor: mutex S

cursor: mutex X

cursor: mutex S wait on X

CHILD CURSORS

What do I do about all these children?

CHILD CURSORS

What do I do about all these children?

```
alter system flush shared_pool;
```

CHILD CURSORS

What do I do about all these children?

```
alter system flush shared_pool;
```

```
dbms_shared_pool.purge  
( 'address,hash_value', 'C' );
```

CHILD CURSORS

Manually Flush a Particular SQL_ID

```
var name varchar2(50)
accept sql_id prompt 'Enter value for sql_id: '
alter session set events '5614566 trace name context forever';
begin
  select address||','||hash_value into :name
  from v$sqlarea where sql_id like '&sql_id';
  dbms_shared_pool.purge(:name,'C',1);
end;
/
```

CHILD CURSORS

Automatically Flush at 500 Children:

```
sqlplus -s "/ as sysdba" <<EOF
var name varchar2(50)
alter session set events '5614566 trace name context forever';
begin
  for s in (select sql_id,version_count
            from v$sqlarea where version_count>=500)
  loop
    select address||','||hash_value into :name
    from v$sqlarea where sql_id=s.sql_id;
    dbms_shared_pool.purge(:name,'C',1);
  end loop;
end;
/
exit
EOF
```

Topics:

⇒ The REAL Execution Plan

⇒ Histograms

⇒ Child Cursors

⇒ Automatic Memory Management

⇒ SQLPlus Startup Slow

Automatic Memory Management

What is AMM?

db_cache_size, log_buffer,
large_pool_size, java_pool_size, etc...

Automatic Memory Management

How do I turn on AMM?

```
alter system set memory_max_target=150G;  
alter system set memory_target=100G;
```

Automatic Memory Management

Am I using AMM?

```
show parameter memory_max_target  
show parameter memory_target
```


Automatic Memory Management

**And This is a
GOOD
Thing.**

RIGHT?

Automatic Memory Management

And This is **IT**
OR IS IT
This?

RIGHT?

Automatic Memory Management

What is it doing?

GV\$MEMORY_RESIZE_OPS
GV\$SGA_RESIZE_OPS

```
select COMPONENT,OPER_TYPE,START_TIME,  
       (INITIAL_SIZE-TARGET_SIZE)/1024/1024 MB  
from dba_hist_memory_resize_ops order by 3;
```

Automatic Memory Management

COMPONENT	OPER_TYPE	START_TIME	MB
-----	-----	-----	-----
DEFAULT buffer cache	SHRINK	2010-07-02 00:16:01	256
shared pool	GROW	2010-07-02 00:16:01	-256
shared pool	SHRINK	2010-07-02 00:16:01	256
DEFAULT buffer cache	GROW	2010-07-02 00:16:01	-256
DEFAULT buffer cache	SHRINK	2010-07-02 00:16:56	-256
shared pool	GROW	2010-07-02 00:16:56	256
shared pool	SHRINK	2010-07-02 00:16:57	-256
DEFAULT buffer cache	GROW	2010-07-02 00:16:57	256
DEFAULT buffer cache	SHRINK	2010-07-02 00:20:28	-256
shared pool	GROW	2010-07-02 00:20:28	256
shared pool	SHRINK	2010-07-02 00:20:30	-256
DEFAULT buffer cache	GROW	2010-07-02 00:20:30	256
DEFAULT buffer cache	SHRINK	2010-07-02 00:23:19	-256
shared pool	GROW	2010-07-02 00:23:19	256
shared pool	SHRINK	2010-07-02 00:23:30	-256
DEFAULT buffer cache	GROW	2010-07-02 00:23:30	256
DEFAULT buffer cache	SHRINK	2010-07-02 00:23:30	-256

Automatic Memory Management

What does this cost you?

⇒ **CPU**

⇒ **Waits:**

SGA: MMAN sleep for component shrink

SGA: allocation forcing component growth

Wait for shrink lock

Wait for shrink lock2

⇒ **Invalidation of Child Cursors**

ORA-7445

Additional Hard Parses

Execution Plan Instability

Automatic Memory Management

So What's a DBA to Do?

- ⇒ Small DB's with plenty memory:
turn on AMM
set minimum sizes for all memory
components**
- ⇒ Larger DB's:
turn off AMM
manually set all memory components**

Automatic Memory Management

What are the current values?

```
select inst_id,nvl(pool,name)||'='||  
       trim(to_char(sum(bytes)/1024/1024,'999999'))||'M' parm  
from gv$sgastat  
group by inst_id,nvl(pool,name)  
union  
select inst_id,'pga_aggregate_target='||  
       trim(to_char(value/1024/1024,'999999'))||'M' parm  
from gv$pgastat where name='aggregate PGA target parameter'  
order by 1,2;
```

Automatic Memory Management

INST_ID PARM

```
-----  
1  buffer_cache=23552M  
   fixed_sga=2M  
   java pool=1536M  
   large pool=1536M  
   log_buffer=6M  
   pga_aggregate_target=20480M  
   shared pool=8704M  
   streams pool=258M  
  
2  buffer_cache=23552M  
   fixed_sga=2M  
   java pool=1536M  
   large pool=1536M  
   log_buffer=6M  
   pga_aggregate_target=20480M  
   shared pool=8704M  
   streams pool=256M
```


Automatic Memory Management

```
create pfile='twl_pfile.ora' from spfile;  
shutdown immediate  
vi twl_pfile.ora  
create spfile from pfile='twl_pfile.ora';  
startup
```

Topics:

- ⇒ The REAL Execution Plan
- ⇒ Histograms
- ⇒ Child Cursors
- ⇒ Automatic Memory Management
- ⇒ SQLPlus Startup Slow

SQLPLUS STARTUP SLOW

Ever notice a lag between entering your password and actually getting the SQL> prompt?

How do you figure out what it is doing?

SQLPLUS STARTUP SLOW

Ever notice a lag between entering your password and actually getting the SQL> prompt?

How do you figure out what it is doing?

If you are on a Unix platform: TRUSS

SQLPLUS STARTUP SLOW

```
truss -o /tmp/truss_sqlplus_delay.out -f -a -D  
-E sqlplus "/ as sysdba"
```

```
tail -f /tmp/truss_sqlplus_delay.out
```

SQLPLUS STARTUP SLOW

```
14924: 0.0002 0.0000 getrlimit(RLIMIT_STACK, 0xFFFFFFFF7FFF7840) = 0
14924: 0.0001 0.0000 setrlimit(RLIMIT_STACK, 0xFFFFFFFF7FFF7840) = 0
14924: 0.0001 0.0000 times(0xFFFFFFFF7FFF8BB0) = 18274341
14924: 0.0002 0.0000 getuid() = 20001 [20001]
14924: 0.0000 0.0000 getuid() = 20001 [20001]
14924: 0.0001 0.0000 door_info(6, 0xFFFFFFFF7FFF6238) = 0
14924: 0.0002 0.0000 door_call(6, 0xFFFFFFFF7FFF6330) = 0
14924: 0.0001 0.0000 getgid() = 1000 [1000]
14921: read(12, 0x100232416, 8208) (sleeping...)
14924: 10.5851 0.7747 open("/opt/app/oracle/admin/dtusp1st/adump/dtusp1st2_ora_14924_1.aud",
O_RDWR|O_CREAT|O_EXCL, 0660) = 16
14924: 0.0004 0.0001 write(16, " A u d i t   f i l e   ", 11) = 11
14924: 0.0010 0.0000 write(16, " / o p t / a p p / o r a"..., 62) = 62
14924: 0.0001 0.0000 write(16, "\n", 1) = 1
14924: 0.0003 0.0000 write(16, " O r a c l e   D a t a b"..., 118) = 118
14924: 0.0001 0.0000 write(16, "\n", 1) = 1
14924: 0.0002 0.0000 write(16, " O R A C L E _ H O M E   "..., 49) = 49
14924: 0.0001 0.0000 write(16, " S y s t e m   n a m e : "..., 19) = 19
```

It appears to be a 10second sleep after getting the O/S groupid, then a read of something from parent process (note the 14921 vs. 14924) then opening the audit file for output.

SQLPLUS STARTUP SLOW

Hmmm...so it looks like the read of the adump directory slow...let's verify:

```
truss -f -a -D -E ls /opt/app/oracle/admin/dtus01st/adump.old
```

```
22480: 0.0015 0.0002 lstat64("/opt/app/oracle/admin/dtus01st/adump", 0xFFBFF428) = 0
22480: 0.0001 0.0000 openat(-3041965, "/opt/app/oracle/admin/dtus01st/adump.old",
O_RDONLY|O_NDELAY|O_LARGEFILE) = 3
22480: 0.0001 0.0000 fcntl(3, F_SETFD, 0x00000001) = 0
22480: 0.0001 0.0000 fstat64(3, 0xFFBFF308) = 0
22480: 0.0130 0.0002 getdents64(3, 0xFF384000, 8192) = 480
22480: 0.0001 0.0000 getdents64(3, 0xFF384000, 8192) = 240
22480: 13.3572 0.3404 getdents64(3, 0xFF384000, 8192) = 1008
22480: 0.0077 0.0002 getdents64(3, 0xFF384000, 8192) = 1248
22480: 0.0139 0.0002 getdents64(3, 0xFF384000, 8192) = 672
22480: 0.0001 0.0000 brk(0x00045150) = 0
22480: 0.0001 0.0000 brk(0x0004F150) = 0
22480: 0.0242 0.0002 getdents64(3, 0xFF384000, 8192) = 48
22480: 0.0004 0.0001 getdents64(3, 0xFF384000, 8192) = 384
22480: 0.0069 0.0002 getdents64(3, 0xFF384000, 8192) = 1008
22480: 0.0002 0.0000 getdents64(3, 0xFF384000, 8192) = 0
22480: 0.0001 0.0000 close(3) = 0
```

SQLPLUS STARTUP SLOW

Noticed that the size of the adump directory was huge (evidently from a previous time when there were a lot of audit files). Even though the files were removed, the directory file itself was still huge:

```
==> 1 /opt/app/oracle/admin/dtus01st
total 144806
drwxr-x---  2 oracle  oinstall    1024 Oct  9  2009 scripts
drwxr-x---  2 oracle  oinstall      96 Oct  9  2009 dpdump
drwxr-x---  2 oracle  oinstall      96 Oct  9  2009 hdump
drwxr-x---  2 oracle  oinstall      96 Oct  9  2009 pfile
drwxr-x---  3 oracle  oinstall      96 May  1 10:44 ..
drwxr-x---  2 oracle  oinstall 73973760 Aug 11 05:00 adump
drwxr-x---  8 oracle  oinstall    1024 Aug 11 09:10 .
```


SQLPLUS STARTUP SLOW

So rename the dir, create new dir, copy audit files into new dir, problem solved:

```
cd /opt/app/oracle/admin/dtus01st
mv adump adump.old
mkdir adump
chmod 750 adump
cp adump.old/* adump
```

```
==> l /opt/app/oracle/admin/dtus01st
drwxr-x--- 2 oracle oinstall 1024 Oct 9 2009 scripts
drwxr-x--- 2 oracle oinstall 96 Oct 9 2009 dpdump
drwxr-x--- 2 oracle oinstall 96 Oct 9 2009 hdump
drwxr-x--- 2 oracle oinstall 96 Oct 9 2009 pfile
drwxr-x--- 3 oracle oinstall 96 May 1 10:44 ..
drwxr-x--- 2 oracle oinstall 73973760 Aug 11 05:00 adump.old
drwxr-x--- 8 oracle oinstall 1024 Aug 11 09:10 .
drwxr-x--- 2 oracle oinstall 9216 Aug 11 09:11 adump
```

SQLPLUS STARTUP SLOW

Let's verify that this fixed it:

```
truss -f -a -D -E ls /opt/app/oracle/admin/dtus01st/adump.old
```

```
28682: 0.0010 0.0008 lstat64("/opt/app/oracle/admin/dtus01st/adump", 0xFFBFF428) = 0
28682: 0.0008 0.0000 openat(-3041965, "/opt/app/oracle/admin/dtus01st/adump",
O_RDONLY|O_NDELAY|O_LARGEFILE) = 3
28682: 0.0002 0.0000 fcntl(3, F_SETFD, 0x00000001) = 0
28682: 0.0001 0.0000 fstat64(3, 0xFFBFF308) = 0
28682: 0.0001 0.0001 getdents64(3, 0xFF384000, 8192) = 480
28682: 0.0001 0.0000 getdents64(3, 0xFF384000, 8192) = 240
28682: 0.1041 0.1040 getdents64(3, 0xFF384000, 8192) = 1008
28682: 0.0003 0.0000 getdents64(3, 0xFF384000, 8192) = 1248
28682: 0.0002 0.0000 getdents64(3, 0xFF384000, 8192) = 672
28682: 0.0001 0.0000 brk(0x00045150) = 0
28682: 0.0001 0.0000 brk(0x0004F150) = 0
28682: 0.0001 0.0000 getdents64(3, 0xFF384000, 8192) = 48
28682: 0.0001 0.0000 getdents64(3, 0xFF384000, 8192) = 384
28682: 0.0002 0.0000 getdents64(3, 0xFF384000, 8192) = 1008
28682: 0.0001 0.0000 getdents64(3, 0xFF384000, 8192) = 0
28682: 0.0001 0.0000 close(3) = 0
```

SOUG TechDay

Wednesday – October 6, 2010

Confirmed Speakers:
Craig Shallahamer
Robert Freeman

19 sessions
4 tracks

<http://soug.org>



Troy's Tuning Tips

Q & A

Troy Ligon
tligon@soug.org