

Analyzing SQL Statement with the SQL_ID Data Collector

Martin Frauendorfer, SAP Active Global Support

martin.frauendorfer@sap.com

May 2010

Agenda



1. **SQL Statement Tuning**
2. **The SQL_ID Data Collector**
3. **Output Sections of the SQL_ID Data Collector**
4. **Example**

Agenda



1. **SQL Statement Tuning**
2. **The SQL_ID Data Collector**
3. **Output Sections of the SQL_ID Data Collector**
4. **Example**

Motivation

- **There are many ways to optimize database performance, e.g.:**
 - Proper parameter settings
 - Efficient SGA and PGA setup
 - Implementation of recommended patches
 - Proper CBO statistics setup
 - Minimization of fragmentation
 - Elimination of system bottlenecks (CPU, paging, I/O, server network)
 - Reduction of database load from application perspective
 - Timed events analysis and optimization
 - SQL statement tuning
- **See SAP Note 618868 (“FAQ: Oracle Performance”) for more information.**
- **Experience shows that usually **SQL statement tuning is most important** – and most time consuming.**

Which SQL statements should be tuned?

- **Application independent tuning to optimize general database performance:**
 - Top SQL statements in terms of elapsed time (which is most important to reduce the overall database time)
 - Top SQL statements in terms of disk reads
 - Top SQL statements in terms of buffer gets
- **Tuning of a certain application**
 - Most critical SQL statements determined with an application SQL trace
- **Here we will focus on the pure technical approach**

How does the technical tuning normally work in SAP environments?

- **Transaction ST04 / DBACOCKPIT is used to display the SQL statements available in the current shared cursor cache (V\$SQL).**
- **Dependent on the demands this list can be filtered and sorted by e.g. elapsed time, disk reads or buffer gets.**
- **Further information available in ST04 / DBACOCKPIT (e.g. execution plan, coding location) and in other transactions (e.g. SE16 for table definition) is used to check for tuning approaches.**

What are the restrictions of this normal SAP approach?

- **The content of the shared cursor cache is not representative, because for several reasons parts can be invalidated (so that the performance counters start from 0 again):**
 - SQL statements accessing tables that get new CBO statistics
 - SQL statements accessing tables that are somehow else touched with a DDL operation (e.g. index rebuild)
 - SQL statements that are replaced due to shared pool space requirements
 - All SQL statements if database is restarted
- **No information is available for several important details, e.g.:**
 - Previous execution plans
 - Timed events related to SQL statement
 - Contribution of different tables and indexes to total SQL statement runtime
 - PGA and PSAPTEMP consumption

How can these restrictions be eliminated?

- With the **SQL_ID Data Collector!**
- It is a single **SELECT** statement that is available via **SAP Note 1257075** for Oracle **>= 10g** and that collects comprehensive information for a certain **SQL** statement
- The only input parameter is the **SQL_ID** of the **SQL** statement.
- **What's a SQL_ID?**
 - It's a unique short identifier linked to a certain **SQL** text.
 - As long as the **SQL** text remains the same, also the **SQL_ID** remains the same.
 - The same **SQL** text results in the same **SQL_ID** independent of time, release or system.
 - It is available as of Oracle 10g.
 - It is part of many Oracle performance views.
 - If you know the **SQL_ID**, you can easily retrieve all information from these current and historic performance views.

How to identify the SQL_IDs to be analyzed?

- In ST04 / DBACOCKPIT they are displayed in the execution plan (of currently running Oracle sessions and the shared cursor cache).
- For a structured analysis of the top SQL statements in history you can use e.g. SQL command `SQL_TopSQLInAWR.txt` available via SAP Note 1438410.
- This command will return the top SQL statements in terms of elapsed time, disk reads, buffer gets (and others).

Agenda



1. SQL Statement Tuning
2. The SQL_ID Data Collector
3. Output Sections of the SQL_ID Data Collector
4. Example

How is the SQL_ID Data Collector executed?

- At first the SQL_ID place holder (typically &&sql_id) has to be replaced at all locations with the SQL_ID to be analyzed.
- Then it can be copied into the „SQL Command Editor“ available in transaction DBACOCKPIT and executed.
- Then it is important to be patient – it collects a lot of information and so runtimes of several minutes are normal.
- When it is finished you can switch to the monospaced „Print Preview“ in order to optimize readability.

What are the restrictions of the SQL_ID Data Collector?

- **The performance can sometimes be really bad, particularly in the following situations:**

- Oracle parameters not set according to SAP Note 830576
- Outdated Oracle patch level
- Missing fixed objects statistics
- DBA_SEGMENTS not optimized for performance (SAP Note 871455)
- Highly active system with large AWR history retention

- **If the performance remains bad, certain output sections can be deactivated or reduced based on settings in the BASIS_INFO section of the statement:**

```
' X' SEGMENT_INFO,  
' ' COMPLEX_INDEX_STATS,  
' X' ASH_INFO,  
' X' FRAGMENTATION_INFO,  
' X' VIEW_INFO,  
' X' INDEX_COLUMNS,  
' X' COMPLEX_SQL_TEXT  
SYSDATE - 70 AWR_BEGIN_DATE,  
SYSDATE AWR_END_DATE,
```

What are the restrictions of the SQL_ID Data Collector?

- **Dependent on the SAP release only 80 characters are displayed per column, so the output is divided in columns with 80 characters**
- **Particularly with older versions of ST04 / ST04N / DBACOCKPIT parsing of the SQL_ID Data Collector sometimes fails**
- **In this case you have the following options:**
 - Use SQL_SQL_ID_DataCollector_Fallback.txt available in SAP Note 1438410 (simplified output)
 - Run it via a remote DBACOCKPIT connection from a system with a recent support package level (see SAP Note 1256322)
 - Run it with another client tool like SQLPLUS (instructions available in the header of the SQL_ID Data Collector command)

Let's now focus on the output provided by the SQL_ID Data Collector...

Agenda



1. SQL Statement Tuning
2. The SQL_ID Data Collector
3. Output Sections of the SQL_ID Data Collector
4. Example

Output Sections of the SQL_ID Data Collector



SQL STATEMENT (V\$SQLSTATS, DBA_HIST_SQLTEXT, V\$SQL):

```
SELECT * FROM "AUSP" WHERE "MANDT" = :A0 AND "OBJEK" IN ( :A1 , :A2 , :A3 , :A4
, :A5 ) AND "MAFID" = :A6 AND "KLART" = :A7 AND "ATINN" IN ( :A8 , :A9 , :A10 ,
:A11 , :A12 , :A13 , :A14 ) AND "DATUV" <= :A15
```

- **SQL text with proper line breaks after <= 80 characters (sounds simple, but was the most complex task when developing the SQL_ID Data Collector!)**
- **Copy and paste possible**

Output Sections of the SQL_ID Data Collector



SQL CACHE (V\$SQLSTATS, DBA_HIST_SQLSTAT):

END_INTERVAL_TIME	PLAN_HASH	EXECUTIONS	ELA_TIME_MS	ELA_TIME_MS/EXEC	CPU_TIME_MS	RECORDS	REC/EXEC
V\$SQL	1851864160	27173	1177462	43.33	600875	280741880	10331.65
V\$SQL	1851864160	0	4127	0.00	2910	0	0.00
V\$SQL	1851864160	0	1898407	0.00	866188	0	0.00
V\$SQLSTATS	1851864160	27173	3079996	113.35	1469973	280741880	10331.65
TOTAL (HISTORY)	1851864160	225126	66049366	293.39	30427871	2679720984	11903.21
TOTAL (HISTORY)	2496132331	45714	1316155	28.79	817011	555037873	12141.53
2010-03-24 09: 00: 43	1851864160	3008	1113831	370.29	503399	48503606	16124.87
2010-03-24 08: 00: 39	1851864160	4030	858931	213.13	423407	35439760	8793.99
2010-03-24 07: 00: 26	1851864160	891	315964	354.62	143212	14381539	16140.90
2010-03-24 06: 00: 22	1851864160	14	429	30.67	353	0	0.00

- **V\$SQL, V\$SQLSTATS: Current information**
- **Multiple V\$SQL lines in case of several child cursors**
- **TOTAL (HISTORY): Summarized history information (per execution plan hash)**
- **2010-03-24 0...: History information for defined AWR interval**
- **If execution plan changes, different PLAN_HASH values are displayed**
- **Useful for understanding general and time related behavior**

Output Sections of the SQL_ID Data Collector



SOURCE	PLAN_HASH	CPU_%	USER_I/O_%	APPLICATION_%	CONCURRENCY_%	OTHER_%
V\$SQLSTATS	1851864160	48	0	0	0	52
DBA_HIST_SQLSTAT	1851864160	46	0	0	0	54
DBA_HIST_SQLSTAT	2496132331	62	0	0	0	38

FETCHES_PER_EXEC	END_OF_FETCH_%	BUFFER_QUALITY_%	DISK_READ_TIME_MS
3.04	100.00	100.00	0.65
3.37	100.00	100.00	1.07
3.40	100.01	100.00	3.80

CHILD_CURSOR_REASON	CURSORS_AFFECTED
TOP_LEVEL_RPI_CURSOR	1
BIND_UACS_DIFF	2
PQ_SLAVE_MISMATCH	1

- High level wait time distribution (CPU, I/O, ...)
- If “END_OF_FETCH_%” is much smaller than 100 % it can mean that the client usually doesn’t fetch the whole result set.
- DISK_READ_TIME_MS can indicate I/O bottlenecks if significantly higher than expected (e.g > 10 ms)
- CHILD_CURSOR_REASON indicates root causes for multiple child cursors

Output Sections of the SQL_ID Data Collector



VIEW DEFINITIONS (V\$SQL, V\$OBJECT_DEPENDENCY, DBA_OBJECTS, DBMS_METADATA.GET_DDL):

RSBKDATA_V:

```
CREATE OR REPLACE FORCE VIEW "SAPR3"."RSBKDATA_V" ("REQUI D", "DATAPAKI D", "ST  
EPI D", "TSTMP", "RELI D", "SRTF2", "REQUI D30", "STEPI D10", "TSTPNM", "DELTODATE")  
AS SELECT T2."REQUI D", T1."DATAPAKI D", T2."STEPI D", T2."TSTMP", T1."RELI D",  
T1."SRTF2", T1."REQUI D30", T1."STEPI D10", T2."TSTPNM", T2."DELTODATE" FROM "RSBK  
DATA" T1, "RSBKDATAI NFO" T2 WHERE T1."REQUI D30" = T2."REQUI D30" AND T1."DATAPAKI  
D" = T2."DATAPAKI D" AND T1."STEPI D10" = T2."STEPI D10"
```

- **Definition of used views**
- **Only available if statement is currently in V\$SQL**
- **Not available for SYS views**
- **Truncated to <= 4000 byte**

Output Sections of the SQL_ID Data Collector



EXECUTION PLAN (V\$SQL_PLAN, DBA_HIST_SQL_PLAN):

PLAN_HASH	CHILD	TIMESTAMP	ID	SCOL	CPU_%	COST	CARD	ACTION
1851864160	0	2010-03-24 08:34:36	0	0	0	2	0	SELECT STATEMENT
			1	0	0	0	0	PX COORDINATOR
			2	0	0	1	3	PX SEND QC (RANDOM) (:TQ10000)
			3	0	0	1	3	NESTED LOOPS
			4	0	0	0	0	PX BLOCK ITERATOR
			5	1	0	0	3	INDEX FAST FULL SCAN (RSBKDATA~0)
			6	0	0	0	1	TABLE ACCESS BY INDEX ROWID (RSBKDATAINFO)
			7	3	0	0	1	INDEX UNIQUE SCAN (RSBKDATAINFO~0)
2496132331	0	2010-03-08 00:06:01	0	0	0	2	0	SELECT STATEMENT
			1	0	0	1	11	NESTED LOOPS
			2	0	0	1	1	TABLE ACCESS BY INDEX ROWID (RSBKDATAINFO)
			3	1	0	0	1	INDEX RANGE SCAN (RSBKDATAINFO~0)
			4	4	0	0	11	INDEX RANGE SCAN (RSBKDATA~0)

→ Current and historic execution plans

→ Mapping to time frames via PLAN_HASH, because TIMESTAMP can be misleading

Output Sections of the SQL_ID Data Collector



SQL WORKAREAS (V\$SQL_WORKAREA):

OPERATION_ID	OPERATION_TYPE	LAST_MEMORY_USED	LAST_TEMPSEG_SIZE	MAX_TEMPSEG_SIZE
1	SORT (v2)		0	
4	SORT (v2)		3659776	
5	HASH-JOIN		1254400	
14	SORT (v2)		5079040	
16	SORT (v2)		5530624	

- **Current and historic SQL workareas (PGA and PSAPTEMP areas used for operations like sort, hash or bitmap)**
- **OPERATION_ID maps to ID in the execution plan overview**
- **Based on this information it is possible to understand in which part of a complex execution plan significant efforts are required for workarea operations.**

Output Sections of the SQL_ID Data Collector



TEMPORARY SEGMENTS (V\$TEMPSEG_USAGE):

TABLESPACE_NAME	CONTENTS	SEGTYPE	SIZE_MB
PSAPTEMP	TEMPORARY	LOB_DATA	6885.00

- **Displays information about temporary segments associated to SQL statement**
- **Example: Temporary LOBs in J2EE environments**

Output Sections of the SQL_ID Data Collector



SQL INFO, TIMED EVENTS AND ACCESSED OBJECTS (DBA_HIST_ACTIVE_SESS_HISTORY):

USER_NAME	SQL_TYPE	MODULE	MOD_ACTION
SAPR3	SELECT	SAPLRSBATCH	8278

- **Basic information like SQL_TYPE extracted from Active Session History**
- **Sometimes the only information source to understand what was going on (e.g. if CREATE INDEX is not recorded in shared cursor cache)**

Output Sections of the SQL_ID Data Collector



PLAN_HASH	OBJECT_NAME	SAMPLES	PLAN_PCT	TOTAL_PCT
1851864160	not available	6917	99.88	98.14
1851864160	RSBKDATA~0	8	0.12	0.11
2496132331	not available	115	100.00	1.63

PLAN_HASH	ACTION	SAMPLES	PLAN_PCT	TOTAL_PCT
1851864160	CPU	4775	68.95	67.75
1851864160	PX Deq Credit: send blkd	1934	27.93	27.44
1851864160	SQL*Net more data to client	145	2.09	2.06
1851864160	PX qref latch	59	0.85	0.84
2496132331	CPU	98	85.22	1.39
2496132331	SQL*Net more data to client	17	14.78	0.24

PLAN_HASH	OBJECT_NAME	ACTION	SAMPLES	PLAN_PCT	TOTAL_PCT	BLOCKS_PER_WAIT
1851864160	not available	CPU	4775	68.95	67.75	
1851864160	not available	PX Deq Credit: send blkd	1934	27.93	27.44	
1851864160	not available	SQL*Net more data to client	145	2.09	2.06	
1851864160	not available	PX qref latch	59	0.85	0.84	
2496132331	not available	CPU	98	85.22	1.39	
2496132331	not available	SQL*Net more data to client	17	14.78	0.24	

- **Active Session History samples aggregated by accessed objects and / or timed events**
- **Very helpful to understand with what activity and related to which segment the time is lost**
- **Restriction: Objects are only recorded for certain timed events (like I/O or enqueues)**

Output Sections of the SQL_ID Data Collector



TOP HOURS (DBA_HIST_ACTIVE_SESS_HISTORY):

PLAN_HASH	HOUR	PLAN_PCT	TOTAL_PCT
3061063750	2010-03-17 11: 00: 00	7. 78	7. 78
3061063750	2010-03-16 12: 00: 00	7. 62	7. 62
3061063750	2010-03-16 13: 00: 00	7. 02	7. 02
3061063750	2010-03-16 11: 00: 00	6. 09	6. 09
3061063750	2010-03-19 17: 00: 00	5. 11	5. 11
3061063750	2010-03-22 10: 00: 00	5. 11	5. 11

→ Hours with highest activity of analyzed SQL statement according to ASH

Output Sections of the SQL_ID Data Collector



SEGMENT STATISTICS (V\$SEGMENT_STATISTICS):

SEGMENT_NAME	LOG_READS	PHYS_READS	DIR_READS	PHYS_WRITES	DIR_WRITES
RSBKDATA	3883168	149462	0	627154	0
RSBKDATAI NFO	290975568	109	0	236	0
RSBKDATAI NFO~0	292972736	72	0	258	0
RSBKDATAI NFO~001	10080	12	0	140	0
RSBKDATA~0	82183968	9031	0	16853	0

SEGMENT STATISTICS (DBA_HIST_SEG_STAT):

SEGMENT_NAME	LOG_READS	PHYS_READS	DIR_READS	PHYS_WRITES	DIR_WRITES
RSBKDATA	41234976	1407491	0	6864974	0
RSBKDATAI NFO	2644048448	423	0	2954	0
RSBKDATAI NFO~0	2668055344	436	0	3355	0
RSBKDATAI NFO~001	20592	16	0	470	0
RSBKDATA~0	670068976	46621	0	203544	0

- **Current and historic segment statistics of tables and their indexes accessed by SQL statement**
- **Global values, not SQL statement specific**
- **Can sometimes still be useful if a top SQL statement contributes significantly to the overall figures**

Output Sections of the SQL_ID Data Collector



INDEX COLUMNS (DBA_INDEXES):

TABLE_NAME	INDEX_NAME	COLUMN_NAME	NUM_DISTINCT	AVG_COL_LEN	COLUMN_POSITION
RSBKDATA	RSBKDATA~0	RELID	1	3	1
		REQUI D30	1	31	2
		DATAPAKID	1	7	3
		STEPI D10	1	10	4
		SRTF2	3	3	5
RSBKDATAINFO	RSBKDATAINFO~0	REQUI D30	1	31	1
		DATAPAKID	1	7	2
		STEPI D10	1	10	3
	RSBKDATAINFO~001	REQUI D	1	5	1

→ Indexes and indexed columns of tables related to SQL statement

Output Sections of the SQL_ID Data Collector



CBO STATISTICS (DBA_TAB_STATISTICS, DBA_INDEXES, DBA_TAB_COLUMNS):

TABLE STATISTICS:

OWNER	TABLE_NAME	NUM_ROWS	BLOCKS	AVG_ROW_LEN	LAST_ANALYZED	USER_STATS	INI_TRANS
SAPR3	RSBKDATA	3	55025	2925	2010-03-21 06:05:57	NO	1
SAPR3	RSBKDATAINFO	1	21	83	2010-03-21 06:05:57	NO	1

INDEX STATISTICS (INDEX PARTITIONS ARE NOT CONSIDERED):

INDEX_NAME	NUM_ROWS	LEAF_BLOCKS	DISTINCT_KEYS	CLUST_FACTOR	BLEVEL	LAST_ANALYZED	SAMPLE_SIZE
RSBKDATA~0	3	1	3	2	2	2010-03-21 06:05:57	3
RSBKDATAINFO~0	1	1	1	1	1	2010-03-21 06:05:57	1
RSBKDATAINFO~001	1	1	1	1	0	2010-03-21 06:05:57	1

→ CBO statistics for involved tables and their indexes

→ Some other minor information like INI_TRANS or COMPRESS also included

Output Sections of the SQL_ID Data Collector



CBO STATISTICS (DBA_TAB_STATISTICS, DBA_INDEXES, DBA_TAB_COLUMNS):

COLUMN STATISTICS:

TABLE_NAME	COLUMN_NAME	NUM_DISTINCT	DENSITY	NUM_BUCKETS	AVG_COL_LEN	NULLABLE	NUM_NULLS	DATA_TYPE
RSBKDATA	CLUSTD	0	0.00000	0	2868	Y	0	BLOB
	CLUSTR	2	0.50000	1	4	N	0	NUMBER
	DATAPAKID	1	1.00000	1	7	N	0	VARCHAR2
	RELI D	1	1.00000	1	3	N	0	VARCHAR2
	REQUI D30	1	1.00000	1	31	N	0	VARCHAR2
	SRTF2	3	0.33333	1	3	N	0	NUMBER
	STEPI D10	1	1.00000	1	10	N	0	VARCHAR2
RSBKDATAINFO	DATAPAKID	1	1.00000	1	7	N	0	VARCHAR2
	DELTODATE	1	1.00000	1	9	N	0	VARCHAR2
	REQUI D	1	1.00000	1	5	N	0	NUMBER
	REQUI D30	1	1.00000	1	31	N	0	VARCHAR2
	STEPI D	1	1.00000	1	3	N	0	NUMBER
	STEPI D10	1	1.00000	1	10	N	0	VARCHAR2
	TSTMP	1	1.00000	1	9	N	0	NUMBER
	TSTPNM	1	1.00000	1	9	N	0	VARCHAR2

→ CBO statistics for columns of involved tables

Output Sections of the SQL_ID Data Collector



TABLE STATISTICS HISTORY:

OWNER	TABLE_NAME	PARTITION_NAME	STATS_UPDATE_TIME
SAPR3	RSBKDATA		2010-03-21 06:05:57
			2010-03-14 06:07:44
			2010-03-07 06:13:49
			2010-02-28 06:08:36
			2010-02-21 06:07:35
SAPR3	RSBKDATAINFO		2010-03-21 06:05:57
			2010-03-14 06:07:45
			2010-03-07 06:13:50
			2010-02-28 06:08:36
			2010-02-21 06:07:35

- Last times of CBO statistics updates for involved tables
- Default retention time: 31 days
- Useful for checking if new CBO statistics impacted the CBO decisions

Output Sections of the SQL_ID Data Collector



COLLECTION OF SPECIAL STATISTICS:

OPERATION	START_TIME	END_TIME
gather_fixed_objects_stats	2010-01-16 17: 31: 10	2010-01-16 17: 37: 46
gather_dictionary_stats	2010-01-16 17: 25: 23	2010-01-16 17: 31: 10
gather_fixed_objects_stats	2010-01-08 10: 27: 53	2010-01-08 10: 37: 37
gather_dictionary_stats	2010-01-08 10: 22: 13	2010-01-08 10: 27: 52

- **Times of recent collections of special statistics**
- **Fixed objects statistics, dictionary statistics and system statistics are considered**
- **Useful for checking if CBO decision is impacted by new special statistics**

Output Sections of the SQL_ID Data Collector



TABLE MODIFICATIONS (DBA_TAB_MODIFICATIONS):

OWNER	TABLE_NAME	INSERTS	UPDATES	DELETES	TIMESTAMP	TRUNCATED	DROP_SEGMENTS
SAPR3	RSBKDATA	513198	0	513225	2010-03-24 07:29:45	NO	0
SAPR3	RSBKDATAINFO	2071	0	2071	2010-03-24 07:29:45	NO	0

- **Number of INSERT, UPDATE and DELETE operations since last CBO statistics creation**
- **Helpful to understand if a table is frequently changed and what has changed since last CBO statistics creation**

Output Sections of the SQL_ID Data Collector



FRAGMENTATION INFORMATION (BASED ON CBO STATISTICS):

SEGMENT_NAME	UNUSED_MB	QUALITY_%	SEG_GROSS_MB	SEG_NET_MB	DATA_GROSS_MB	DATA_NET_MB
RSBKDATA	378.66	0.00	432.00	378.60	376.74	0.00
RSBKDATA~0	12.36	0.00	14.00	12.36	0.01	0.00
RSBKDATAI NFO	2.20	0.00	2.52	2.20	0.14	0.00
RSBKDATAI NFO~0	0.57	0.01	0.64	0.57	0.01	0.00
RSBKDATAI NFO~001	0.01	0.12	0.02	0.01	0.01	0.00

**→ Unused space and storage quality analysis based on CBO statistics
(not always reliable)**

Output Sections of the SQL_ID Data Collector



SEGMENT ADVISOR INFORMATION (DBA_ADVISOR_TASKS, DBA_ADVISOR_FINDINGS, DBA_ADVISOR_OBJECTS):

SEGMENT_NAME MESSAGE

RSBKDATA Enable row movement of the table SAPR3.RSBKDATA and perform shrink, estimated savings is 409560085 bytes.

→ **Information provided by Oracle Segment Advisor**

→ **Only available if regularly scheduled or manually executed**

Output Sections of the SQL_ID Data Collector



SEGMENT INFORMATION (DBA_SEGMENTS, DBA_LOBS, DBA_TABLESPACES):

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE	EXTENT_MANAGEMENT	ASSM	BYTES	EXTENTS
SAPR3	RSBKDATA	TABLE	PSAPBTABD	LMTS (AUTOALLOCATE)	YES	452984832	125
SAPR3	RSBKDATA. CLUSTD	LOBSEGMENT	PSAPBTABD	LMTS (AUTOALLOCATE)	YES	1048576	16
SAPR3	RSBKDATA. CLUSTD	LOBINDEX	PSAPBTABD	LMTS (AUTOALLOCATE)	YES	65536	1
SAPR3	RSBKDATAINFO	TABLE	PSAPSTABD	LMTS (USER)	NO	2637824	2
SAPR3	RSBKDATAINFO~0	INDEX	PSAPSTABI	LMTS (USER)	NO	671744	2
SAPR3	RSBKDATAINFO~001	INDEX	PSAPSTABI	LMTS (USER)	NO	16384	1
SAPR3	RSBKDATA~0	INDEX	PSAPBTABI	LMTS (AUTOALLOCATE)	YES	14680064	29

→ Segment related information

→ Proper assignment of LOB segment and LOB index to table column

→ Real size of the segments

Output Sections of the SQL_ID Data Collector



LOB INFORMATION (DBA_LOBS):

OWNER	TABLE_NAME	COLUMN_NAME	LOB_NAME	PCTVERSION	CACHE	IN_ROW	LOGGING
SAPR3	RSBKDATA	CLUSTD	SYS_LOB0018270029C00007\$\$	10	YES	YES	YES

→ Overview of LOB segments related to accessed tables

→ Can be useful to check whether a LOB is cached

Output Sections of the SQL_ID Data Collector



BUFFER POOL OCCUPATION (V\$BH):

SEGMENT_NAME	SEGMENT_TYPE	POOL	BLKS	SIZE	POOL_GB	SEG_POOL_MB	POOL_%	SEG_DISK_MB	SEG_CACHED_%	DIRTY_%
RSBKDATA	TABLE	DEFAULT	8192		5.70	124.15	2.13	432.00	28.74	3.22
RSBKDATA~0	INDEX	DEFAULT	8192		5.70	11.34	0.19	14.00	81.03	5.10
RSBKDATAINFO~0	INDEX	DEFAULT	8192		5.70	0.04	0.00	0.64	6.10	0.00
RSBKDATAINFO	TABLE	DEFAULT	8192		5.70	0.02	0.00	2.52	0.93	33.33
SYS_LOB0018270029C00007\$\$	LOBSEGMENT	DEFAULT	8192		5.70	0.01	0.00	1.00	0.78	0.00
RSBKDATAINFO~001	INDEX	DEFAULT	8192		5.70	0.01	0.00	0.02	50.00	0.00

→ Buffer pool state of involved segments

→ Shows how well a segment is cached in buffer pool (SEG_CACHED_%)

Output Sections of the SQL_ID Data Collector



UNDO STATISTICS (V\$UNDOSTAT):

BEGIN_TIME	MAXQUERYLEN	TUNED_UNDORETENTION
2010-02-19 08:05:58	534	214027
2010-02-17 18:25:58	1498	150659
2010-02-17 18:15:58	898	150072
2010-02-17 18:05:58	295	149480
2010-02-17 03:45:58	1240	142387
2010-02-17 03:35:58	635	142755
2010-02-16 19:55:58	1631	176505
2010-02-16 17:55:58	1486	170041

→ Undo information for longer running SQL_IDs

Output Sections of the SQL_ID Data Collector



PARAMETER SETTINGS (V\$PARAMETER2):

PARAMETER_NAME	IS_DEFAULT	VALUE
db_cache_size	FALSE	16777216
db_file_multiblock_read_count	TRUE	128
event	FALSE	10027 trace name context forever, level 1
event	FALSE	10028 trace name context forever, level 1
event	FALSE	10142 trace name context forever, level 1
event	FALSE	10183 trace name context forever, level 1
event	FALSE	10191 trace name context forever, level 1
event	FALSE	10891 trace name context forever, level 1
event	FALSE	14532 trace name context forever, level 1
event	FALSE	38068 trace name context forever, level 100
event	FALSE	38085 trace name context forever, level 1
event	FALSE	38087 trace name context forever, level 1
event	FALSE	44951 trace name context forever, level 1024
optimizer_dynamic_sampling	FALSE	6
optimizer_features_enable	TRUE	10.2.0.4
optimizer_index_caching	TRUE	0
optimizer_index_cost_adj	FALSE	20
optimizer_mode	TRUE	ALL_ROWS
optimizer_secure_views_merging	TRUE	TRUE
pga_aggregate_target	FALSE	1048576000
sga_target	FALSE	8589934592
workarea_size_policy	TRUE	AUTO
_awr_flush_threshold_metrics	FALSE	TRUE
...		

→ Settings of parameters with potential performance impact

Output Sections of the SQL_ID Data Collector



PARAMETER CHANGES (DBA_HIST_PARAMETER):

END_INTERVAL_TIME	PARAMETER_NAME	VALUE	IS_DEFAULT	VALUE_BEFORE	WAS_DEFAULT
2010-02-13 17:41:21	_fix_control	6670551: ON, 5705630: ON, 5765456: 3, 6221403: ON, 6329318: ON, 6430500: ON, 6440977: ON, 6626018: ON, 6972291: ON, 7325597: ON, 6399597: ON, 7692248: ON, 5099019: ON, 7891471: ON	FALSE	6670551: ON, 5705630: ON, 5765456: 3, 6221403: ON, 6329318: ON, 6430500: ON, 6440977: ON, 6626018: ON, 6972291: ON, 7325597: ON, 6399597: ON, 7692248: ON, 5099019: ON	FALSE
2010-02-13 17:41:21	audit_trail	DB	FALSE	NONE	TRUE
2010-02-13 17:41:21	event	10027 trace name context forever, level 1, 10028 trace name context forever, level 1, 10183 trace name context forever, level 1, 10191 trace name context forever, level 1, 10629 trace name context forever, level 32, 38068 trace name context forever, level 100, 14532 trace name context forever, level 1, 10142 trace name context forever, level 1, 10411 trace name context forever, level 1, 38085 trace name context forever, level 1, 44951 trace name context forever, level 1	FALSE	10027 trace name context forever, level 1, 10028 trace name context forever, level 1, 10183 trace name context forever, level 1, 10191 trace name context forever, level 1, 10629 trace name context forever, level 32, 38068 trace name context forever, level 100, 14532 trace name context forever, level 1, 10891 trace name context forever, level 1, 10142 trace name context forever, level 1, 10411 trace name context forever, level 1, 38085 trace name context forever, level 1, 38087 trace name context forever, level 1	FALSE

→ Parameter change history

Output Sections of the SQL_ID Data Collector



BI ND VARIABLE CONTENTS (V\$SQL_BI ND_CAPTURE, DBA_HI ST_SQLBI ND):

SAMPLES: 155

CAPTURE_TIME	NAME	VALUE
CONSTANT	: A0	' 100'
	: A7	' 022'
	: A8	' '
2010-04-14 20: 28: 24	: A1	' 000000000010659721'
	: A2	' 0000000830'
	: A3	' 0000000831'
	: A4	' 0000000832'
	: A5	' 0000000870'
	: A6	' 0000000871'
2010-04-14 16: 51: 58	: A1	' 000000000014579619'
	: A2	' 0000000825'
	: A3	' 0000000826'
	: A4	' 0000000827'
	: A5	' 0000000828'
	: A6	' 0000000829'
...		

- Captured bind variable values
- Maximum of one value set per hour
- Variables with always the same captured value are displayed in the CONSTANT section

Agenda



1. **SQL Statement Tuning**
2. **The SQL_ID Data Collector**
3. **Output Sections of the SQL_ID Data Collector**
4. **Example**

Finally a real life example...

- **SQL statement with increased number of buffer gets during certain times:**

SQL Analysis Example

<https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/d03e9239-1646-2d10-4faa-be0b6e2bd8cc>

Thank you!



No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned and associated logos displayed are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Einige von der SAP AG und deren Vertriebspartnern vertriebene Softwareprodukte können Softwarekomponenten umfassen, die Eigentum anderer Softwarehersteller sind.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge und andere in diesem Dokument erwähnte SAP-Produkte und Services sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und in mehreren anderen Ländern weltweit. Alle anderen in diesem Dokument erwähnten Namen von Produkten und Services sowie die damit verbundenen Firmenlogos sind Marken der jeweiligen Unternehmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

Die in diesem Dokument enthaltenen Informationen sind Eigentum von SAP. Dieses Dokument ist eine Vorabversion und unterliegt nicht Ihrer Lizenzvereinbarung oder einer anderen Vereinbarung mit SAP. Dieses Dokument enthält nur vorgesehene Strategien, Entwicklungen und Funktionen des SAP®-Produkts und ist für SAP nicht bindend, einen bestimmten Geschäftsweg, eine Produktstrategie bzw. -entwicklung einzuschlagen. SAP übernimmt keine Verantwortung für Fehler oder Auslassungen in diesen Materialien. SAP garantiert nicht die Richtigkeit oder Vollständigkeit der Informationen, Texte, Grafiken, Links oder anderer in diesen Materialien enthaltenen Elemente. Diese Publikation wird ohne jegliche Gewähr, weder ausdrücklich noch stillschweigend, bereitgestellt. Dies gilt u. a., aber nicht ausschließlich, hinsichtlich der Gewährleistung der Marktgängigkeit und der Eignung für einen bestimmten Zweck sowie für die Gewährleistung der Nichtverletzung geltenden Rechts.

SAP übernimmt keine Haftung für Schäden jeglicher Art, einschließlich und ohne Einschränkung für direkte, spezielle, indirekte oder Folgeschäden im Zusammenhang mit der Verwendung dieser Unterlagen. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder die Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.

Alle Rechte vorbehalten.