

# How to Gather SQL Resource Consumption Metrics in Oracle

Karen Morton  
Sr. Technical Consultant





karen.morton@enkitec.com

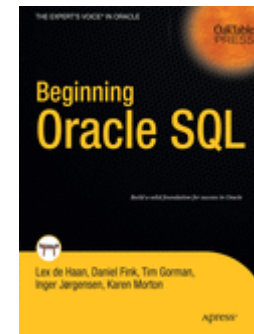
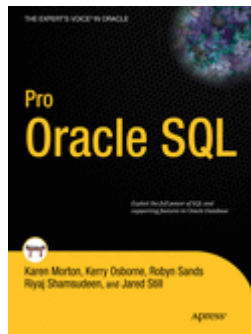


karenmorton.blogspot.com



karen\_morton

Oracle ACE 



If you can't measure it, you can't manage it.

—*David Garvin*

There are two types of  
performance problems in this  
world...

1

Response time  
problems.

# 2

Inefficiencies that aren't response  
time problems.

Yet.

Therefore...

You must be able to attack  
response time problems for  
specific tasks that the business  
cares about...



...even if they are inefficiencies  
that aren't yet noticeable as user  
response time problems.

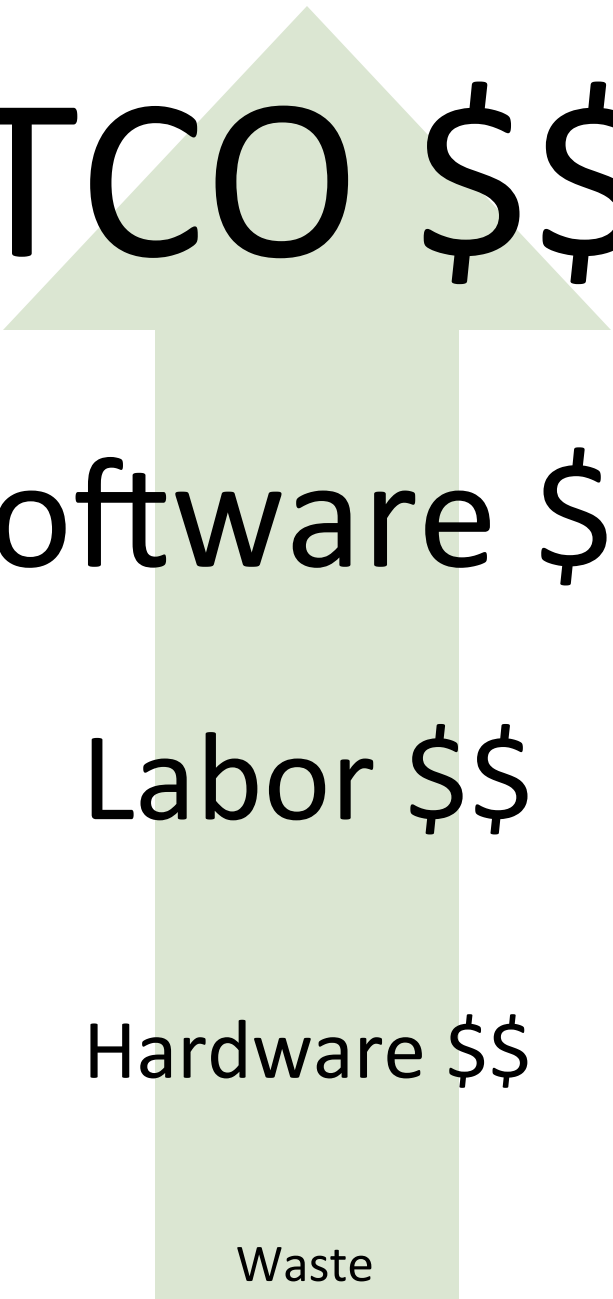
Why should you care?



Because

**waste**

costs you money



TCO \$\$\$

Software \$\$

Labor \$\$

Hardware \$\$

Waste

And waste...

...makes  
other work  
go slower...

...even your fast stuff







Why guess?  
When you can know.

# Resource Consumption Metrics



Time

&

Resources

Start simple  
with  
**AUTOTRACE**

```
SQL> SET TIMING ON
```

```
SQL> ALTER SESSION SET
```

```
STATISTICS_LEVEL = ALL;
```

*Can also use /\*+ gather\_plan\_statistics \*/ hint*

```
SQL> SET AUTOTRACE TRACEONLY
```

```

SQL>set autotrace traceonly
SQL>
SQL>select /* kmtst01g */
2
33     ...
34     from
35         WC_ACCT_HIER_DH T348271 /* Dim_WC_ACCT_HIER_DH */ ,
36         (SELECT DISTINCT FROM_CURCY_CD CURRENCY FROM W_EXCH_RATE_G) T347319,
37         W_MCAL_DAY_D T156337 /* Dim_W_MCAL_DAY_D_Fiscal_Day */ ,
38         W_HIERARCHY_D T148616 /* Dim_W_HIERARCHY_D_Segment2 */ ,
39         W_GL_SEGMENT_D T148908 /* Dim_W_GL_SEGMENT_D_Segment2 */ ,
40         W_HIERARCHY_D T148543 /* Dim_W_HIERARCHY_D_Segment3 */ ,
41         W_GL_SEGMENT_D T148937 /* Dim_W_GL_SEGMENT_D_Segment3 */ ,
42         W_HIERARCHY_D T125129 /* Dim_W_HIERARCHY_D_Segment1 */ ,
43         W_GL_SEGMENT_D T149255 /* Dim_W_GL_SEGMENT_D_Segment1 */ ,
44         W_INT_ORG_D T111515 /* Dim_W_INT_ORG_D_Company */ ,
45         W_LEDGER_D T146058 /* Dim_W_LEDGER_D */ ,
46         WC_GL_BALANCE_A_KM T348279 /* Fact_WC_GL_BALANCE_A */
47     where ( T156337.ROW_WID = T348279.BALANCE_DT_WID and T347319.CURRENCY = T348279.LOC_CURR_CODE
48     and T148616.HIER_CODE = T148908.SEGMENT_LOV_ID
49     and T148616.HIER20_CODE = T148908.SEGMENT_VAL_CODE and T148543.HIER_CODE = T148937.SEGMENT_LOV_ID
50     and T148543.HIER20_CODE = T148937.SEGMENT_VAL_CODE
51     and T148908.SEGMENT_LOV_ID = T348279.ACCOUNT_SEG2_ATTRIB
52     and T148908.SEGMENT_VAL_CODE = T348279.ACCOUNT_SEG2_CODE and T125129.HIER_CODE = T149255.SEGMENT_LOV_ID
53     and T125129.HIER20_CODE = T149255.SEGMENT_VAL_CODE
54     and T148937.SEGMENT_LOV_ID = T348279.ACCOUNT_SEG3_ATTRIB and T148937.SEGMENT_VAL_CODE = T348279.ACCOUNT_SEG3_CODE
55     and T111515.ROW_WID = T348279.COMPANY_ORG_WID and T146058.ROW_WID = T348279.LEDGER_WID
56     and T149255.SEGMENT_LOV_ID = T348279.ACCOUNT_SEG1_ATTRIB
57     and T149255.SEGMENT_VAL_CODE = T348279.ACCOUNT_SEG1_CODE and T111515.COMPANY_FLG = 'Y'
58     and T148908.SEGMENT_LOV_ID = 'Department~SHARE'
59     and T148937.SEGMENT_LOV_ID = 'Operating Unit~SHARE' and T149255.SEGMENT_LOV_ID = 'Account~SHARE'
60     and T156337.MCAL_PERIOD_NAME = 'March'
61     and T156337.MCAL_PER_NAME_YEAR = '2012' and T347319.CURRENCY = 'USD'
62     and T348271.ROW_WID = T348279.ACCOUNT_HIER_WID and T348279.LOC_CURR_CODE = 'USD'
63     and T111515.ORG_NUM is not null and T148543.HIER9_CODE is not null
64     and (T146058.LEDGER_NAME in ('Actuals', 'Local Ledger in Local Currency'))
65     and (T348271.LEVEL_NUM in (2, 3, 4, 5, 6, 7, 8, 9, 10, 12)) and T148616.HIER10_CODE is not null )
66     group by T149255.SEGMENT_VAL_CODE, T348271.LEVEL_NUM, T348271.SORT_NUM, T348271.ACCOUNT_CODE_DISP,
67     T348271.ACCOUNT_NAME_DISP, T348271.REPORT_SIGN,
68     case when T125129.HIER10_CODE = 'INTEREST_REVENUE10' then T125129.X_NODE_ID * 10 +
69     cast(T149255.SEGMENT_VAL_CODE as DOUBLE PRECISION )
70     else T125129.X_NODE_ID + cast(T149255.SEGMENT_VAL_CODE as DOUBLE PRECISION ) end
71     ) D1
72 order by c2, c3, c4, c5, c1, c6, c7
73 ;
307 rows selected.

```

Execution Plan

Plan hash value: 3249606325

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		4432K	2299M		980K (1)	00:00:39
1	WINDOW BUFFER		4432K	2299M		980K (1)	00:00:39
2	SORT GROUP BY		4432K	2299M	2473M	980K (1)	00:00:39
* 3	HASH JOIN		4432K	2299M		470K (1)	00:00:19
* 4	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	1973	94704		138 (0)	00:00:01
* 5	HASH JOIN		4370K	2067M		470K (1)	00:00:19
* 6	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	729	29889		138 (0)	00:00:01
* 7	HASH JOIN		4355K	1889M		470K (1)	00:00:19
* 8	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	303	13029		138 (0)	00:00:01
* 9	HASH JOIN		4350K	1709M		470K (1)	00:00:19
* 10	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	1651	44577		446 (1)	00:00:01
* 11	HASH JOIN		4352K	1597M		469K (1)	00:00:19
* 12	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	605	16335		446 (1)	00:00:01
* 13	HASH JOIN		4353K	1486M		469K (1)	00:00:19
* 14	TABLE ACCESS STORAGE FULL	W_INT_ORG_D	857	14569		30 (0)	00:00:01
* 15	HASH JOIN		4358K	1417M		469K (1)	00:00:19
* 16	TABLE ACCESS STORAGE FULL	WC_ACCT_HIER_DH	363	55176		7 (0)	00:00:01
* 17	HASH JOIN		6444K	1161M		469K (1)	00:00:19
* 18	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	202	5454		446 (1)	00:00:01
* 19	HASH JOIN		6458K	997M		468K (1)	00:00:19
* 20	TABLE ACCESS STORAGE FULL	W_LEDGER_D	11	275		6 (0)	00:00:01
* 21	MERGE JOIN		7046K	920M		468K (1)	00:00:19
* 22	HASH JOIN		7046K	893M		468K (1)	00:00:19
* 23	TABLE ACCESS BY INDEX ROWID	W_MCAL_DAY_D	31	682		37 (0)	00:00:01
* 24	INDEX SKIP SCAN	IDX\$\$_2FA30002	31			32 (0)	00:00:01
* 25	TABLE ACCESS STORAGE FULL	WC_GL_BALANCE_A_KM	7258K	768M		468K (1)	00:00:19
* 26	SORT JOIN		1	4		3 (67)	00:00:01
* 27	VIEW		1	4		2 (50)	00:00:01
* 28	HASH UNIQUE		1	4		2 (50)	00:00:01
* 29	BITMAP INDEX STORAGE FAST FULL SCAN	W_EXCH_RATE_G_M3	147	588		1 (0)	00:00:01

Elapsed time: 34.10 seconds



Predicate Information (identified by operation id):

```
-----  
3 - access("T125129"."HIER_CODE"="T149255"."SEGMENT_LOV_ID" AND "T125129"."HIER20_CODE"="T149255"."SEGMENT_VAL_CODE")  
4 - storage("T125129"."HIER_CODE"='Account~SHARE')  
   filter("T125129"."HIER_CODE"='Account~SHARE')  
5 - access("T148616"."HIER_CODE"="T148908"."SEGMENT_LOV_ID" AND "T148616"."HIER20_CODE"="T148908"."SEGMENT_VAL_CODE")  
6 - storage("T148616"."HIER_CODE"='Department~SHARE' AND "T148616"."HIER10_CODE" IS NOT NULL)  
   filter("T148616"."HIER_CODE"='Department~SHARE' AND "T148616"."HIER10_CODE" IS NOT NULL)  
7 - access("T148543"."HIER_CODE"="T148937"."SEGMENT_LOV_ID" AND "T148543"."HIER20_CODE"="T148937"."SEGMENT_VAL_CODE")  
8 - storage("T148543"."HIER_CODE"='Operating Unit~SHARE' AND "T148543"."HIER9_CODE" IS NOT NULL)  
   filter("T148543"."HIER_CODE"='Operating Unit~SHARE' AND "T148543"."HIER9_CODE" IS NOT NULL)  
9 - access("T149255"."SEGMENT_LOV_ID"="T348279"."ACCOUNT_SEG1_ATTRIB" AND  
   "T149255"."SEGMENT_VAL_CODE"="T348279"."ACCOUNT_SEG1_CODE")  
10 - storage("T149255"."SEGMENT_LOV_ID"='Account~SHARE')  
   filter("T149255"."SEGMENT_LOV_ID"='Account~SHARE')  
11 - access("T148908"."SEGMENT_LOV_ID"="T348279"."ACCOUNT_SEG2_ATTRIB" AND  
   "T148908"."SEGMENT_VAL_CODE"="T348279"."ACCOUNT_SEG2_CODE")  
12 - storage("T148908"."SEGMENT_LOV_ID"='Department~SHARE')  
   filter("T148908"."SEGMENT_LOV_ID"='Department~SHARE')  
13 - access("T111515"."ROW_WID"="T348279"."COMPANY_ORG_WID")  
14 - storage("T111515"."COMPANY_FLG"='Y' AND "T111515"."ORG_NUM" IS NOT NULL)  
   filter("T111515"."COMPANY_FLG"='Y' AND "T111515"."ORG_NUM" IS NOT NULL)  
15 - access("T348271"."ROW_WID"="T348279"."ACCOUNT_HIER_WID")  
16 - storage("T348271"."LEVEL_NUM"=2 OR "T348271"."LEVEL_NUM"=3 OR "T348271"."LEVEL_NUM"=4 OR "T348271"."LEVEL_NUM"=5  
   OR "T348271"."LEVEL_NUM"=6 OR "T348271"."LEVEL_NUM"=7 OR "T348271"."LEVEL_NUM"=8 OR "T348271"."LEVEL_NUM"=9 OR  
   "T348271"."LEVEL_NUM"=10 OR "T348271"."LEVEL_NUM"=12)  
   filter("T348271"."LEVEL_NUM"=2 OR "T348271"."LEVEL_NUM"=3 OR "T348271"."LEVEL_NUM"=4 OR "T348271"."LEVEL_NUM"=5  
   OR "T348271"."LEVEL_NUM"=6 OR "T348271"."LEVEL_NUM"=7 OR "T348271"."LEVEL_NUM"=8 OR "T348271"."LEVEL_NUM"=9 OR  
   "T348271"."LEVEL_NUM"=10 OR "T348271"."LEVEL_NUM"=12)  
17 - access("T148937"."SEGMENT_LOV_ID"="T348279"."ACCOUNT_SEG3_ATTRIB" AND  
   "T148937"."SEGMENT_VAL_CODE"="T348279"."ACCOUNT_SEG3_CODE")  
18 - storage("T148937"."SEGMENT_LOV_ID"='Operating Unit~SHARE')  
   filter("T148937"."SEGMENT_LOV_ID"='Operating Unit~SHARE')  
19 - access("T146058"."ROW_WID"="T348279"."LEDGER_WID")  
20 - storage("T146058"."LEDGER_NAME"='Actuals' OR "T146058"."LEDGER_NAME"='Local Ledger in Local Currency')  
   filter("T146058"."LEDGER_NAME"='Actuals' OR "T146058"."LEDGER_NAME"='Local Ledger in Local Currency')  
22 - access("T156337"."ROW_WID"="T348279"."BALANCE_DT_WID")  
24 - access("T156337"."MCAL_PERIOD_NAME"='March' AND "T156337"."MCAL_PER_NAME_YEAR"='2012')  
   filter("T156337"."MCAL_PER_NAME_YEAR"='2012' AND "T156337"."MCAL_PERIOD_NAME"='March')  
25 - storage("T348279"."LOC_CURR_CODE"='USD' AND "T348279"."ACCOUNT_SEG2_ATTRIB"='Department~SHARE' AND  
   "T348279"."ACCOUNT_SEG3_ATTRIB"='Operating Unit~SHARE' AND "T348279"."ACCOUNT_SEG1_ATTRIB"='Account~SHARE')  
   filter("T348279"."LOC_CURR_CODE"='USD' AND "T348279"."ACCOUNT_SEG2_ATTRIB"='Department~SHARE' AND  
   "T348279"."ACCOUNT_SEG3_ATTRIB"='Operating Unit~SHARE' AND "T348279"."ACCOUNT_SEG1_ATTRIB"='Account~SHARE')  
26 - access("T347319"."CURRENCY"="T348279"."LOC_CURR_CODE")  
   filter("T347319"."CURRENCY"="T348279"."LOC_CURR_CODE")  
29 - storage("FROM_CURCY_CD"='USD')  
   filter("FROM_CURCY_CD"='USD')
```

## Statistics

---

1	recursive calls
7	db block gets
1729710	consistent gets
1722997	physical reads
0	redo size
53471	bytes sent via SQL*Net to client
584	bytes received via SQL*Net from client
22	SQL*Net roundtrips to/from client
3	sorts (memory)
0	sorts (disk)
307	rows processed

Want more?

# Mine Shared Pool V\$ Views

# Find the sql\_id & child\_number

```
select /* recentsql */ sql_id, child_number, hash_value, address, executions, sql_text
from v$sql
where command_type in (2,3,6,7,189)
and UPPER(sql_text) like UPPER('%&1%')
and UPPER(sql_text) not like UPPER('%recentsql%')
```

SQL>@rs kmtst-auto

SQL_ID	CHILD_NUMBER	HASH_VALUE	ADDRESS	EXECUTIONS	SQL_TEXT
6x2sh0kvzac0t	0	3086299161	000000020FE75408	1	select /* kmtst-auto */ D1.c10 as c1, D1.c5 as c2, D1.c11 as c3, D1.c7 as c8, D1.c9 as c9, D1.c8 as c10, D1.c6 as c11, D1.c4 as c12, sum(D1.c3) over (partition by D1.c5) as c13, sum(T348279.ACTIVITY_LOC_AMT_YTD) as c1, + LEADING(T156337) */ sum(T348279.ACTIVITY_LOC_AMT_YTD) as c1, Y_LOC_AMT_YTD_YAGO) as c3, sum(T348279.ACTIVITY_LOC_AMT_YTD) as c6, sum(T348279.ACTIVITY_LOC_AMT_YAGO) as c9,

*Put a recognizable comment in your SQL to make it easy to find.*

Retrieve plan using  
DBMS\_XPLAN.display\_cursor  
( '&sql\_id', '&child\_no',  
'ALLSTATS LAST +COST +BYTES')

Other options available: +PEEKED\_BINDS +OUTLINE +ADVANCED, etc.

--Estimates--

-----Actuals-----

Plan hash value: 3249606325

Id	Operation	Name	Starts	E-Rows	E-Bytes	E-Temp	Cost (%CPU)	A-Rows	A-Time	Buffers	Reads	OMem	lMem	Used-Mem
0	SELECT STATEMENT		1				980K (100)	307	00:00:32.64	1729K	1722K			
1	WINDOW BUFFER		1	4433K	2300M		980K (1)	307	00:00:32.64	1729K	1722K	80896	80896	71680 (0)
2	SORT GROUP BY		1	4433K	2300M	2474M	980K (1)	307	00:00:32.64	1729K	1722K	106K	106K	96256 (0)
3	HASH JOIN		1	4433K	2300M		470K (1)	1303K	00:00:30.75	1729K	1722K	837K	837K	1336K (0)
4	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	1	1973	94704		138 (0)	1973	00:00:00.01	517	0			
5	HASH JOIN		1	4371K	2068M		470K (1)	1303K	00:00:30.05	1729K	1722K	940K	940K	1282K (0)
6	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	1	729	29889		138 (0)	729	00:00:00.01	517	0			
7	HASH JOIN		1	4356K	1890M		470K (1)	1303K	00:00:29.39	1728K	1722K	904K	904K	1283K (0)
8	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	1	303	13029		138 (0)	304	00:00:00.01	517	0			
9	HASH JOIN		1	4352K	1710M		470K (1)	1303K	00:00:28.67	1728K	1722K	981K	981K	1350K (0)
10	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	1	1652	44604		446 (1)	1438	00:00:00.01	1669	0			
11	HASH JOIN		1	4354K	1598M		469K (1)	1303K	00:00:27.99	1726K	1722K	955K	955K	1290K (0)
12	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	1	605	16335		446 (1)	601	00:00:00.01	1669	0			
13	HASH JOIN		1	4358K	1487M		469K (1)	1303K	00:00:27.27	1724K	1722K	1397K	1397K	1290K (0)
14	TABLE ACCESS STORAGE FULL	W_INT_ORG_D	1	857	14569		30 (0)	856	00:00:00.01	97	0			
15	HASH JOIN		1	4363K	1418M		469K (1)	1303K	00:00:26.69	1724K	1722K	735K	735K	1290K (0)
16	TABLE ACCESS STORAGE FULL	WC_ACCT_HIER_DH	1	363	55176		7 (0)	363	00:00:00.01	29	0			
17	HASH JOIN		1	6451K	1162M		469K (1)	1456K	00:00:25.93	1724K	1722K	927K	927K	1286K (0)
18	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	1	202	5454		446 (1)	189	00:00:00.01	1669	0			
19	HASH JOIN		1	6458K	997M		468K (1)	1456K	00:00:25.13	1723K	1722K	1348K	1348K	1270K (0)
20	TABLE ACCESS STORAGE FULL	W_LEDGER_D	1	11	275		6 (0)	11	00:00:00.01	10	0			
21	MERGE JOIN		1	7046K	920M		468K (1)	1541K	00:00:24.46	1723K	1722K			
22	HASH JOIN		1	7046K	893M		468K (1)	1541K	00:00:22.78	1723K	1722K	1236K	1236K	1247K (0)
23	TABLE ACCESS BY INDEX ROWID	W_MCAL_DAY_D	1	31	682		37 (0)	31	00:00:00.01	8	0			
24	INDEX SKIP SCAN	IDX\$\$_2FA30002	1	31			32 (0)	31	00:00:00.01	4	0			
25	TABLE ACCESS STORAGE FULL	WC_GL_BALANCE_A_KM	1	7258K	768M		468K (1)	42M	00:00:11.97	1723K	1722K			
26	SORT JOIN		1541K	1	4		3 (67)	1541K	00:00:00.87	3	0	2048	2048	2048 (0)
27	VIEW		1	1	4		2 (50)	1	00:00:00.01	3	0			
28	HASH UNIQUE		1	1	4		2 (50)	1	00:00:00.01	3	0	1453K	1453K	470K (0)
29	BITMAP INDEX STORAGE FAST FULL SCAN	W_EXCH_RATE_G_M3	1	147	588		1 (0)	1	00:00:00.01	3	0			

Want more?



# SQL Monitor Reports

Automatically generated when  
SQL consumes 5 seconds or  
higher of CPU or IO time

*No need for /\*+ gather\_plan\_statistics \*/ hint  
or STATISTICS\_LEVEL = ALL*

Can use

`/*+ monitor */`

`hint`

*Can turn off using `/*+ no_monitor */ hint`*

# Generate Report for any SQL\_ID

```
select dbms_sqltune.report_sql_monitor  
(report_level=>'ALL',  
type=>'TEXT',  
sql_id=>'6x2sh0kvzac0t') monitor_report  
from dual;
```

Types = (11.1) TEXT, HTML, XML, plus (11.2) EM, ACTIVE.

# Generate Report for current session

```
select dbms_sqltune.report_sql_monitor  
(report_level=>'ALL',  
type=>'TEXT',  
session_id=>sys_context('userenv','sid')  
) monitor_report  
from dual;
```

Global Information

I

```
-----  
Status           : DONE (ALL ROWS)  
Instance ID      : 1  
Session          : SYS (992:9971)  
SQL ID          : 6x2sh0kvzac0t  
SQL Execution ID : 16777216  
Execution Started : 05/05/2012 11:12:28  
First Refresh Time : 05/05/2012 11:12:31  
Last Refresh Time : 05/05/2012 11:13:02  
Duration         : 34s  
Module/Action    : sqlplus.exe/-  
Service         : BIUATAH  
Program         : sqlplus.exe  
Fetch Calls     : 22
```

Global Stats

Elapsed Time(s)	Cpu Time(s)	IO waits(s)	Application waits(s)	Other waits(s)	Fetch Calls	Buffer Gets	Read Reqs	Read Bytes	Cell Offload
33	32	0.80	0.00	0.14	22	2M	18384	13GB	63.37%

SQL Plan Monitoring Details (Plan Hash value=3249606325)

Id	Operation	Name	Rows (Estim)	Cost	Time Active(s)	Start Active	Execs	Rows (Actual)
0	SELECT STATEMENT				1	+34	1	307
1	WINDOW BUFFER		4M	981K	1	+34	1	307
2	SORT GROUP BY		4M	981K	32	+3	1	307
3	HASH JOIN		4M	471K	32	+3	1	1M
4	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	1973	138	1	+3	1	1973
5	HASH JOIN		4M	471K	32	+3	1	1M
6	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	729	138	1	+3	1	729
7	HASH JOIN		4M	470K	32	+3	1	1M
8	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	303	138	1	+3	1	304
9	HASH JOIN		4M	470K	32	+3	1	1M
10	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	1652	446	1	+3	1	1438
11	HASH JOIN		4M	470K	32	+3	1	1M
12	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	605	446	1	+3	1	601
13	HASH JOIN		4M	469K	32	+3	1	1M
14	TABLE ACCESS STORAGE FULL	W_INT_ORG_D	857	30	1	+3	1	856
15	HASH JOIN		4M	469K	32	+3	1	1M
16	TABLE ACCESS STORAGE FULL	WC_ACCT_HIER_DH	363	7	1	+3	1	363
17	HASH JOIN		6M	469K	32	+3	1	1M
18	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	202	446	1	+3	1	189
19	HASH JOIN		6M	469K	32	+3	1	1M
20	TABLE ACCESS STORAGE FULL	W_LEDGER_D	11	6	1	+3	1	11
21	MERGE JOIN		7M	469K	32	+3	1	2M
22	HASH JOIN		7M	469K	33	+2	1	2M
23	TABLE ACCESS BY INDEX ROWID	W_MCAL_DAY_D	31	37	1	+3	1	31
24	INDEX SKIP SCAN	IDX\$\$_2FA30002	31	32	1	+3	1	31
25	TABLE ACCESS STORAGE FULL	WC_GL_BALANCE_A_KM	7M	469K	32	+3	1	43M
26	SORT JOIN		1	3	32	+3	2M	2M
27	VIEW		1	2	1	+3	1	1
28	HASH UNIQUE		1	2	1	+3	1	1
29	BITMAP INDEX STORAGE FAST FULL SCAN	W_EXCH_RATE_G_M3	147	1	1	+3	1	1

SQL Plan Monitoring Details (Plan Hash value=3249606325)

Id	Operation	Name	Read Reqs	Read Bytes	Cell Offload	Mem (Max)	Activity (%)	Activity Detail (# samples)
0	SELECT STATEMENT					71680		
1	WINDOW BUFFER					96256	14.71	Cpu (5)
2	SORT GROUP BY					1M		
3	HASH JOIN					1M	2.94	Cpu (1)
4	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D				1M		
5	HASH JOIN					1M		
6	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D				1M		
7	HASH JOIN					1M		
8	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D				1M		
9	HASH JOIN					1M		
10	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D				1M		
11	HASH JOIN					1M		
12	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D				1M	2.94	Cpu (1)
13	HASH JOIN					1M		
14	TABLE ACCESS STORAGE FULL	W_INT_ORG_D				1M		
15	HASH JOIN					1M		
16	TABLE ACCESS STORAGE FULL	WC_ACCT_HIER_DH				1M		
17	HASH JOIN					1M		
18	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D				1M	2.94	Cpu (1)
19	HASH JOIN					1M	5.88	Cpu (2)
20	TABLE ACCESS STORAGE FULL	W_LEDGER_D				1M	47.06	Cpu (16)
21	MERGE JOIN							
22	HASH JOIN							
23	TABLE ACCESS BY INDEX ROWID	W_MCAL_DAY_D						
24	INDEX SKIP SCAN	IDX\$\$_2FA30002						
25	TABLE ACCESS STORAGE FULL	WC_GL_BALANCE_A_KM	17340	12GB	63.37%		17.65	Cpu (5) cell smart table scan (1)
26	SORT JOIN					2048		
27	VIEW					481K		
28	HASH UNIQUE							
29	BITMAP INDEX STORAGE FAST FULL SCAN	W_EXCH_RATE_G_M3						



# SQL Monitoring Report

## SQL Text

```
select /* kmstst-auto */ D1.c10 as c1, D1.c5 as c2, D1.c11 as c3, D1.c12 as c4, D1.c13 as c5, D1.c14 as c6, D1.c15 as c7, D1.c7 as c8, D1.c9 as c9, D1.c8 as c10, D1.c6 as c11, sum(D1.c2) over (partition by D1.c5) as c12, sum(D1.c4) over (partition by D1.c5) as c13, sum(D1.c3) over (partition by D1.c5) as c14, sum(D1.c1) over (partition by D1.c5) as c15 from (select /*+ LEADING(T156337) */ sum(T348279.ACTIVITY_LOC_AMT_YTD) as c1, sum(T348279.ACTIVITY_LOC_AMT) as c2, sum(T348279.ACTIVITY_LOC_AMT_YTD_YAGO) as c3, sum(T348279.ACTIVITY_LOC_AMT_YAGO) as c4, T348271.ACCOUNT_CODE_DISP as c5, sum(T348279.ACTIVITY_LOC_AMT_YTD) as c6, sum(T348279.ACTIVITY_LOC_AMT) as c7, sum(T348279.ACTIVITY_LOC_AMT_YTD_YAGO) as c8, sum(T348279.ACTIVITY_LOC_AMT_YAGO) as c9, T149255.SEGMENT_VAL_CODE as c10, T348271.ACCOUNT_NAME_DISP as c11, T348271.LEVEL_NUM as c12, T348271.REPORT_SIGN as c13, case when T125129.HIER10_CODE = 'INTEREST_REVENUE10' then T125129.X_NODE_ID * 10 + cast(T149255.SEGMENT_VAL_CODE as DOUBLE PRECISION) else T125129.X_NODE_ID + cast(T149255.SEGMENT_VAL_CODE as DOUBLE PRECISION) end as c14, T348271.SORT_NUM as c15 from WC_ACCT_HIER_DH T348271 /* Dim_WC_ACCT_HIER_DH */ , (SELECT DISTINCT FROM CURCY_CD CURRENCY FROM W_EXCH_RATE_G) T347319, W_MCAL_DAY_D T156337 /* Dim_W_MCAL_DAY_D Fiscal Day */ , W_HIERARCHY_D T148616 /* Dim_W_HIERARCHY_D Segment1 */ , W_GL_SEGMENT_D T148908 /* Dim_W_GL_SEGMENT_D Segment2 */ , W_INT_ORG_D T11515 /* Dim_W_INT_ORG_D Company */ , W_LEDGER_D T146058 /* Dim_W_LEDGER_D */ , WC_GL_BALANCE_A KM T348279 /* Fact_WC_GL_BALANCE_A */ where ( T156337.ROW_WID = T348279.BALANCE_DT_WID and T347319.CURRENCY = T348279.LOC_CURR_CODE and T148616.HIER_CODE = T148908.SEGMENT_LOV_ID and T148616.HIER20_CODE = T148908.SEGMENT_VAL_CODE and T148543.HIER_CODE = T148937.SEGMENT_LOV_ID and T148543.HIER20_CODE = T148937.SEGMENT_VAL_CODE and T148908.SEGMENT_LOV_ID = T348279.ACCOUNT_SEG2_ATTRIB and T148908.SEGMENT_VAL_CODE = T348279.ACCOUNT_SEG2_CODE and T125129.HIER_CODE = T149255.SEGMENT_LOV_ID and T125129.HIER20_CODE = T149255.SEGMENT_VAL_CODE and T148937.SEGMENT_LOV_ID = T348279.ACCOUNT_SEG3_ATTRIB and T148937.SEGMENT_VAL_CODE = T348279.ACCOUNT_SEG3_CODE and T11515.ROW_WID = T348279.COMPANY_ORG_WID and T146058.ROW_WID = T348279.LEDGER_WID and T149255.SEGMENT_LOV_ID = T348279.ACCOUNT_SEG1_ATTRIB and T149255.SEGMENT_VAL_CODE = T348279.ACCOUNT_SEG1_CODE and T11515.COMPANY_FLG = 'Y' and T148908.SEGMENT_LOV_ID = 'Department~SHARE' and T148937.SEGMENT_LOV_ID = 'Operating Unit~SHARE' and T149255.SEGMENT_LOV_ID = 'Account~SHARE' and T156337.MCAL_PERIOD_NAME = 'March' and T156337.MCAL_PER_NAME_YEAR = '2012' and T347319.CURRENCY = 'USD' and T348271.ROW_WID = T348279.ACCOUNT_HIER_WID and T348279.LOC_CURR_CODE = 'USD' and T11515.ORG_NUM is not null and T148543.HIER9_CODE is not null and (T146058.LEDGER_NAME in ('Actuals', 'Local Ledger in Local Currency')) and (T348271.LEVEL_NUM in (2, 3, 4, 5, 6, 7, 8, 9, 10, 12)) and T148616.HIER10_CODE is not null ) group by T149255.SEGMENT_VAL_CODE, T348271.LEVEL_NUM, T348271.SORT_NUM, T348271.ACCOUNT_CODE_DISP, T348271.ACCOUNT_NAME_DISP, T348271.REPORT_SIGN, case when T125129.HIER10_CODE = 'INTEREST_REVENUE10' then T125129.X_NODE_ID * 10 + cast(T149255.SEGMENT_VAL_CODE as DOUBLE PRECISION) else T125129.X_NODE_ID + cast(T149255.SEGMENT_VAL_CODE as DOUBLE PRECISION) end ) D1 order by c2, c3, c4, c5, c1, c6, c7
```

## Global Information: DONE (ALL ROWS)

Instance ID : 1  
 Session : SYS (992:9971)  
 SQL ID : 6x2sh0kvzac0t  
 SQL Execution ID : 16777216  
 Execution Started : 05/05/2012 11:12:28  
 First Refresh Time : 05/05/2012 11:12:31  
 Last Refresh Time : 05/05/2012 11:13:02  
 Duration : 34s  
 Module/Action : sqlplus.exe/-  
 Service : BIUATAH  
 Program : sqlplus.exe  
 Fetch Calls : 22

Buffer Gets	IO Requests	Cell Offload Efficiency	Database Time	Wait Activity
2M	18384	63.37%	33s	100%

## SQL Plan Monitoring Details (Plan Hash Value=3249606325)

Id	Operation	Name	Estimated Rows	Cost	Active Period (34s)	Execs	Rows	Memory (Max)	Temp (Max)	IO Requests	Cell Offload Efficiency	CPU Activity	Wait Activity
0	SELECT STATEMENT					1	307						
1	WINDOW BUFFER		4M	981K		1	307	70.0KB					
2	SORT GROUP BY		4M	981K		1	307	94.0KB				16%	
3	HASH JOIN		4M	471K		1	1M	1.3MB					
4	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	1973	138		1	1973						
5	HASH JOIN		4M	471K		1	1M	1.3MB				3.2%	
6	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	729	138		1	729						
7	HASH JOIN		4M	470K		1	1M	1.3MB					
8	TABLE ACCESS STORAGE FULL	W_HIERARCHY_D	303	138		1	304						
9	HASH JOIN		4M	470K		1	1M	1.3MB					
10	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	1652	446		1	1438						
11	HASH JOIN		4M	470K		1	1M	1.3MB					
12	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	605	446		1	601						
13	HASH JOIN		4M	469K		1	1M	1.3MB				3.2%	
14	TABLE ACCESS STORAGE FULL	W_INT_ORG_D	857	30		1	856						
15	HASH JOIN		4M	469K		1	1M	1.3MB					
16	TABLE ACCESS STORAGE FULL	WC_ACCT_HIER_DH	363	7		1	363						
17	HASH JOIN		6M	469K		1	1M	1.3MB					
18	TABLE ACCESS STORAGE FULL	W_GL_SEGMENT_D	202	446		1	189						
19	HASH JOIN		6M	469K		1	1M	1.2MB				3.2%	
20	TABLE ACCESS STORAGE FULL	W_LEDGER_D	11	6		1	11						
21	MERGE JOIN		7M	469K		1	2M					6.5%	
22	HASH JOIN		7M	469K		1	2M	1.2MB					51%
23	TABLE ACCESS BY INDEX ROWID	W_MCAL_DAY_D	31	37		1	31						
24	INDEX SKIP SCAN	IDXS\$_2FA30002	31	32		1	31						
25	TABLE ACCESS STORAGE FULL	WC_GL_BALANCE_A_KM	7M	469K		1	43M			17340 (94%)	63.37%	16%	100%
26	SORT JOIN		1	3		2M	2M	2.0KB					
27	VIEW		1	2		1	1						
28	HASH UNIQUE		1	2		1	1	470.0KB					
29	BITMAP INDEX STORAGE FAST FULL SCAN	W_EXCH_RATE_G_M3	147	1		1	1						

Want more?

Active Session History  
(ASH)  
&  
Automatic Workload Repository  
(AWR)

*Must have license for Diagnostics/Tuning Pack*

```
SQL> @ashrpti
```

```
SQL> @awrsqrpt
```

Want more?

Extended SQL Trace  
dbms\_monitor  
session\_trace\_enable

Remember:  
The more data you have,  
the easier problem diagnosis  
will be.

How you get the data doesn't  
really matter.

Just get it.



# Questions & Answers

