



Ignite IT Performance™

Oracle 11g Results Cache

Janis Griffin
Senior DBA, Confio Software



Who Am I?

- Senior DBA for Confio Software
 - JanisGriffin@confio.com
- 20+ Years in Oracle, SQL Server
- 5+ Years in Oracle Consulting
- Specialize in Performance Tuning
- Review Performance of 100's of Databases for Customers and Prospects



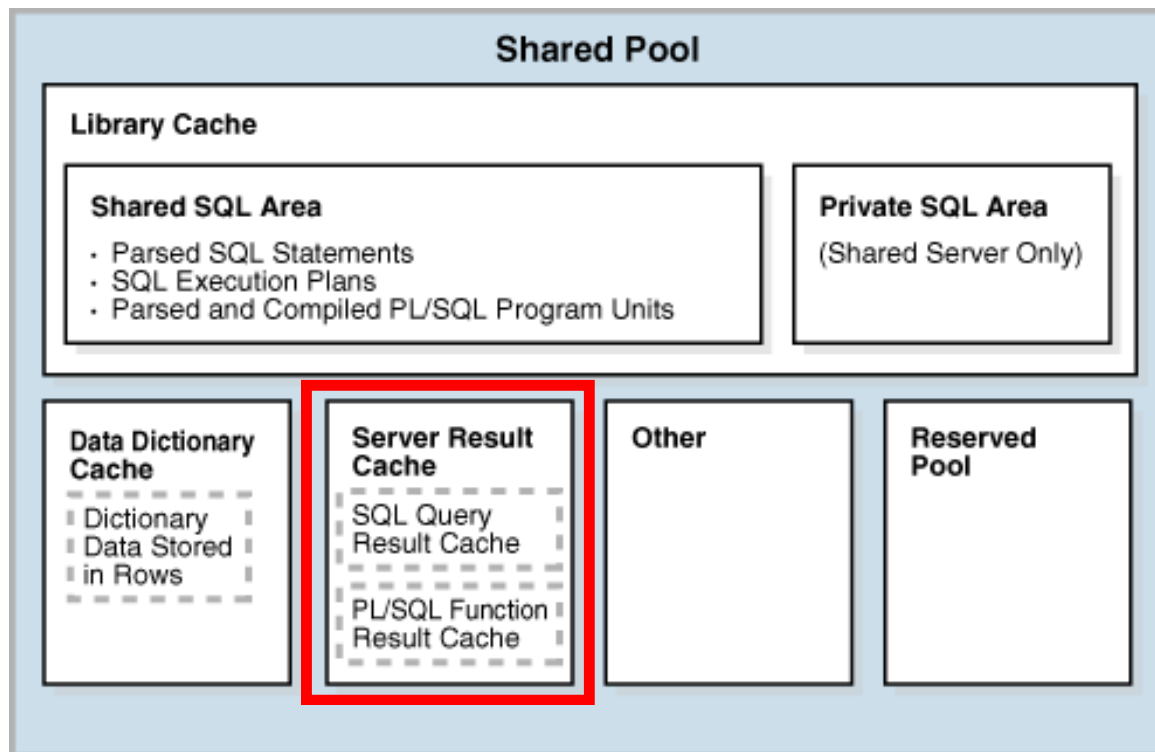
Oracle 11g Result Caches

- New in Oracle 11g, Improved in 11gR2
 - SQL Query Results (local and distributed)
 - PL/SQL Function Results
 - OCI Client Results
- Cached Data is shared across executions
- Automatically marked stale if underlying data is changed
- Can Dramatically Increase Performance



Server Results Cache

- Stores Results of Query or Function Call
- Uses a Slice of the Shared Pool
- Not Affected by Flushing Shared Pool





Queries that Benefit

- Access Large Amount of Data
- Return Few Rows
- Execute Somewhat Frequently
 - Product Lookups / Customer Info
- Based on Slowly Changing Data
 - Country / State / County / Zip Code
- Limited Number of Bind Values
 - Results are cached by Bind Value



- **RESULT_CACHE_MODE**
 - MANUAL (default) – requires a query hint or table annotation
 - FORCE – every result set is cached. Not recommended because it can create significant performance and latching overhead
 - AUTO??? – more about this option
- **RESULT_CACHE_MAX_SIZE**
 - Amount of memory allocated to server result cache
 - 0 (Disabled), 0.25% (memory_target), 0.5% (sga_target) and ~1% (shared_pool_size)
- **RESULT_CACHE_MAX_RESULT**
 - Amount of memory for a single result set
 - 5% (Default)



Using Result Cache

- **Database Setting**

`result_cache_mode = FORCE (not recommended)`

- **Query Hint**

```
select /*+ result_cache */ rep_name, sum(order_total)
from orders
group by rep_name
```

- **Table Annotation Mode**

`alter table order_history result_cache (mode force)`

- **Session Mode**

`alter session set result_cache_mode = force`



- Oracle Error Says AUTO Mode is Supported

```
SQL> alter system set result_cache_mode=incorrect;  
alter system set result_cache_mode=incorrect  
*
```

```
ERROR at line 1:
```

```
ORA-00096: invalid value INCORRECT for parameter  
  result_cache_mode, must be  
from among FORCE, MANUAL, AUTO
```

- Many notes in blogs, etc that say AUTO mode is not supported and probably never will



Example 1 – SQL

Yearly Sales by Sales Representative

```
alter system set result_cache_max_size = 2m;  
alter system set result_cache_max_result = 10;  
show parameter result
```

NAME	TYPE	VALUE
-----	-----	-----
result_cache_max_result	integer	10
result_cache_max_size	big integer	2M
result_cache_mode	string	MANUAL

```
set AUTOTRACE TRACEONLY  
SELECT /*+ result_cache */ sales_rep_id,  
       sum(order_total)  
FROM order_history  
WHERE order_date BETWEEN '1-JAN-09' AND '1-JAN-10'  
GROUP BY sales_rep_id;
```



Example 1 - Execution Plan

Execution Plan

Plan hash value: 1508661739

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2053K	76M	4260 (5)	00:00:52
1	RESULT CACHE	414f6qx2bjqd1fz2abd1u2v6c0				
2	HASH GROUP BY		2053K	76M	4260 (5)	00:00:52
* 3	FILTER					
* 4	TABLE ACCESS FULL	ORDER_HISTORY	2053K	76M	4172 (3)	00:00:51

Predicate Information (identified by operation id):

- 3 - filter('1-JAN-09'<='1-JAN-10')
- 4 - filter("ORDER_DATE">='1-JAN-09' AND "ORDER_DATE"<='1-JAN-10')

Result Cache Information (identified by operation id):

1 - column-count=2; dependencies=(TR.ORDER_HISTORY);

**name="SELECT /*+ result_cache */ sales_rep_id, sum(order_total)
FROM order_history
WHERE order_date BETWEEN '1-JAN-09' AND '1-JAN-10'"**

Note

- dynamic sampling used for this statement



Example 1 - Statistics

First Execution - Statistics

```
375 recursive calls
  0 db block gets
20467 consistent gets
15083 physical reads
  0 redo size
 678 bytes sent via SQL*Net to client
 416 bytes received via SQL*Net from client
   2 SQL*Net roundtrips to/from client
   5 sorts (memory)
   0 sorts (disk)
  10 rows processed
```

Elapsed: 00:00:22.14

Second Execution - Statistics

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

Elapsed: 00:00:00.01



- **V\$RESULT_CACHE_STATISTICS**
 - How well is the cache doing?
 - Monitor CREATES vs. FINDS
- **V\$RESULT_CACHE_MEMORY**
 - Memory components and statistics
 - Possible latching issue in 11.1 when querying
- **V\$RESULT_CACHE_OBJECTS**
 - Objects that are in the cache along with attributes
- **V\$RESULT_CACHE_DEPENDENCY**
 - Dependencies of the results in cache



Example 1 - After First 2 Executions

ID	NAME	VALUE
1	Block Size (Bytes)	1024
2	Block Count Maximum	2048
3	Block Count Current	32
4	Result Size Maximum (Blocks)	204
5	Create Count Success	1
6	Create Count Failure	0
7	Find Count	1
8	Invalidation Count	0
9	Delete Count Invalid	0
10	Delete Count Valid	0



Example 1 – After 32 Executions, 1 Update, 1 Insert

ID	NAME	VALUE
1	Block Size (Bytes)	1024
2	Block Count Maximum	2048
3	Block Count Current	32
4	Result Size Maximum (Blocks)	204
5	Create Count Success	3
6	Create Count Failure	0
7	Find Count	29
8	Invalidation Count	2
9	Delete Count Invalid	0
10	Delete Count Valid	0



Example 1 – After 32 Executions, 1 Update, 1 Insert

```
SELECT ID, TYPE, CREATION_TIMESTAMP,  
       BLOCK_COUNT, COLUMN_COUNT,  
       PIN_COUNT, ROW_COUNT  
FROM   V$RESULT_CACHE_OBJECTS  
WHERE  CACHE_ID = '414f6qx2bjqd1fz2abd1u2v6c0'  
ORDER BY 1
```

ID	TYPE	CREATION_TIMESTAMP	BLOCK_COUNT	COLUMN_COUNT	PIN_COUNT	ROW_COUNT
1	Result	28-feb-10 15:19	1	2	0	10
2	Result	28-feb-10 15:21	1	2	0	10
3	Result	28-feb-10 15:23	1	2	0	10



Example 1 – Dependencies

```
SELECT * FROM V$RESULT_CACHE_DEPENDENCY;
```

RESULT_ID	DEPEND_ID	OBJECT_NO
-----	-----	-----
1	0	309235

```
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE  
FROM dba_objects  
WHERE object_id = 309235;
```

OWNER	OBJECT_NAME	OBJECT_TYPE
-----	-----	-----
TR	ORDER_HISTORY	TABLE



Example 1 – Performance?

CONFIO
SOFTWARE

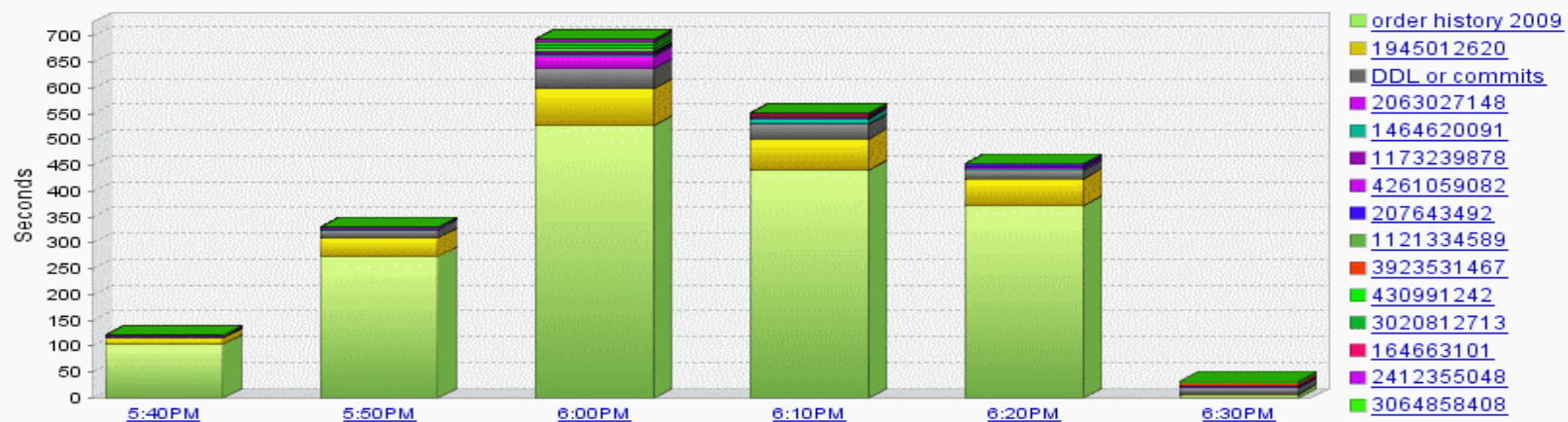
Ignite IT Performance™

Home > Current for CECE_JGRIFFIN(Oracle)

Queries

Highest Total Execution Time | Long Running (Average) | Most Executed

Queries causing the most user wait time in the last hour (sum of all execution times)



4PM to 8PM

SQL Name	order history 2009
Wait Time	28:42 (mm:ss)
Total Wait Time for Time Period	44:34 (mm:ss)
% of Total Wait Time	64%
Average (seconds)	0.71660424
Executions	2,403

SQL Text

```
SELECT SALES_REP_ID, SUM(ORDER_TOTAL) FROM ORDER_HISTORY  
WHERE ORDER_DATE BETWEEN '1-JAN-09' AND '1-JAN-10'  
AND SALES_REP_ID = :B1 GROUP BY SALES_REP_ID
```



Example 1 – Performance?

Home > Trend for CECE_JGRIFFIN(Oracle) > Feb 28 > 4PM-8PM > SQL order history 2009

The time period is now being monitored. New data will display and SQL statistics will increment at the end of each 10 minutes.

Day: Sunday - February 28, 2010

Time: 4:00PM to 8:00PM

Refreshed on: 02/28/10 06:31:43 PM

Timeslice

SQL Data

Waits

Programs

DB Users

O/S Users

Machines

Sessions

Files

Plans

Objects

Blockers

SQL: order history 2009 (747978284) [Name SQL](#)

[View Historical Charts](#)

Statistics

Executions	2,403	Rows Processed	2,402
Parses	26	Disk Reads	35,855,796
Sorts	0	Buffer Gets	36,289,016

Note: SQL Statistics reflect changes in statistical values over the sampled time, and may be 0 or blank if the monitored database instance has not updated its published statistics.

SQL Text

Live Plan

Go

```
SELECT SALES_REP_ID,  
       SUM(ORDER_TOTAL)  
FROM ORDER_HISTORY  
WHERE ORDER_DATE BETWEEN '1-JAN-09' AND '1-JAN-10'  
AND SALES_REP_ID = :B1  
GROUP BY SALES_REP_ID
```



Example 1 – Performance?

Home > Trend for CECE_JGRIFFIN(Oracle) > Feb 28 > 4PM-8PM > SQL order history 2009

The time period is now being monitored. New data will display and SQL statistics will increment at the end of each 10 minutes.

Day:

Time:

Refreshed on: 02/28/10 06:32:18 PM

Timeslice

SQL Data

Waits

Programs

DB Users

O/S Users

Machines

Sessions

Files

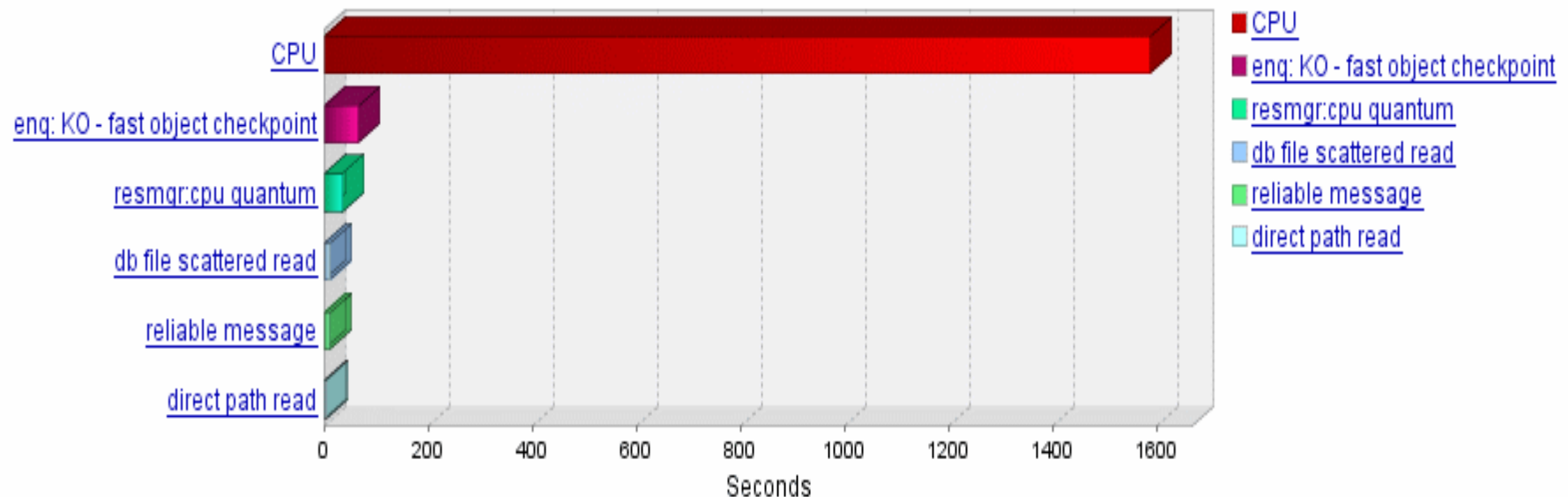
Plans

Objects

Blockers

[Email Chart](#)

Top Waits | CECE_JGRIFFIN | February 28, 2010 - 4:00PM to 8:00PM





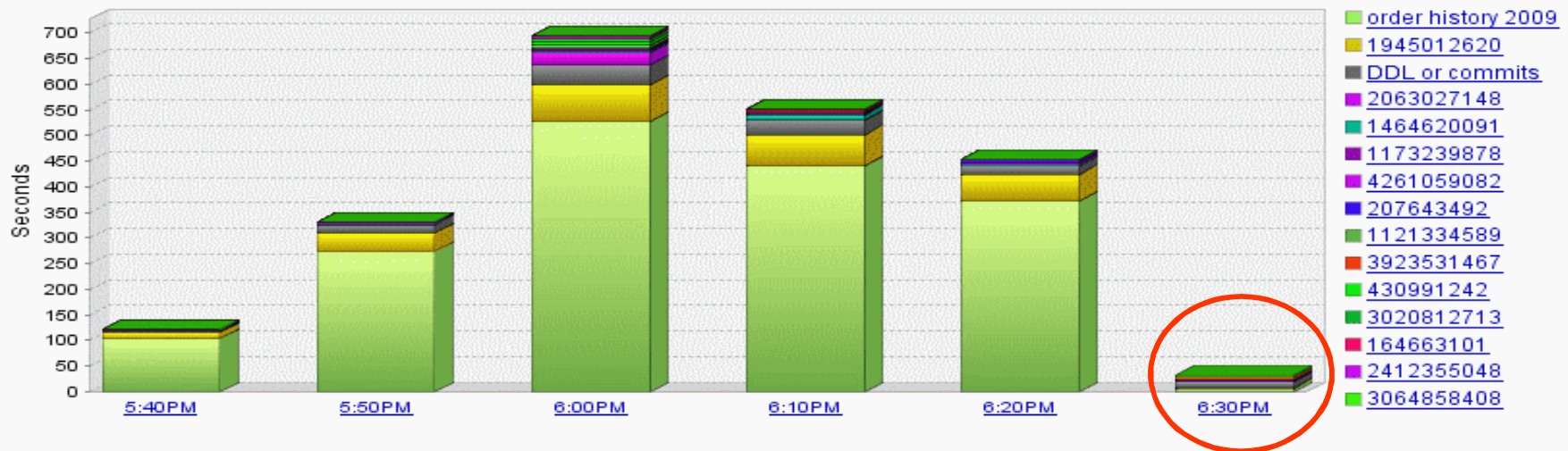
Performance With Result Cache

Home > Current for CECE_JGRIFFIN(Oracle)

Queries

Highest Total Execution Time | Long Running (Average) | Most Executed

Queries causing the most user wait time in the last hour (sum of all execution times)



Home > Current for CECE_JGRIFFIN(Oracle) > Feb 28 >

Day: Sunday - February 28, 2010 Time: 6:33PM to 6:34PM

Timeslice | SQL | Waits | Programs | DB Users | O/S Users

Top SQL Statements | CECE

[order history 2009](#)

SQL Name: order history 2009
Wait CPU
Wait Time 7 seconds
Total Wait Time for SQL 7 seconds
% of Total Wait Time 100%

SQL Text
SELECT SALES_REP_ID, SUM(ORDER_TOTAL) FROM ORDER_HISTORY
WHERE ORDER_DATE BETWEEN '1-JAN-09' AND '1-JAN-10'
AND SALES_REP_ID = :B1 GROUP BY SALES_REP_ID

Click the bar or axis label to drill



Performance With Result Cache

Home > Current for CECE_JGRIFFIN(Oracle) > Feb 28 > 4PM-8PM > 6:30PM-6:40PM > SQL order history 2009

Day: Sunday - February 28, 2010

Time: 6:30PM to 6:40PM

Refreshed on: 02/28/10 07:22:36 PM

Timeslice SQL Data Waits Programs DB Users O/S Users Machines Sessions Files Plans Objects Blockers

SQL: order history 2009 (747978284) Name SQL

[View Historical Charts](#)

Statistics

Executions	7,200	Rows Processed	7,200
Parses	72	Disk Reads	150,830
Sorts	0	Buffer Gets	151,070

Note: SQL Statistics reflect changes in statistical values over the sampled time, and may be 0 or blank if the monitored database instance has not updated its published statistics.

SQL Text

Live Plan Go

```
SELECT SALES_REP_ID,  
       SUM(ORDER_TOTAL)  
FROM ORDER_HISTORY  
WHERE ORDER_DATE BETWEEN '1-JAN-09' AND '1-JAN-10'  
AND SALES_REP_ID = :B1  
GROUP BY SALES_REP_ID
```



- **RESULT_CACHE_REMOTE_EXPIRATION**
 - Expiration time (minutes) for results that depend on remote database objects
 - 0 (Default, Disabled)
- **DML on Remote Database does not Invalidate the local results cache**
- **Must be Careful of Stale Results**



DBMS_RESULT_CACHE

- **BYPASS**
 - Disables result cache database-wide
- **FLUSH**
 - Flushes all objects from result cache
 - Note: flushing shared pool has no affect
- **MEMORY_REPORT**
 - Nice report that shows usage of result cache
- **STATUS**
 - ENABLED or NOT ENABLED
- **INVALIDATE**
 - Invalidate contents of the result cache
- **INVALIDATE_OBJECT**
 - Invalidates contents that rely on object passed in



MEMORY_REPORT

SET SERVEROUTPUT ON

EXECUTE DBMS_RESULT_CACHE.MEMORY_REPORT

Result Cache Memory Report

[Parameters]

Maximum Cache Size = 950272 bytes (928 blocks)

Maximum Result Size = 47104 bytes (46 blocks)

[Memory]

Total Memory = 46340 bytes [0.048% of the Shared Pool]

... Fixed Memory = 10696 bytes [0.011% of the Shared Pool]

... State Object Pool = 2852 bytes [0.003% of the Shared Pool]

... Cache Memory = 32792 bytes (32 blocks) [0.034% of the Shared Pool]

..... Unused Memory = 30 blocks

..... Used Memory = 2 blocks

..... Dependencies = 1 blocks

..... Results = 1 blocks

..... SQL = 1 blocks



- Will Not Work With
 - Temporary tables
 - SYS or SYSTEM tables
 - Sequences (NEXTVAL or CURRVAL)
 - Date/Time Functions – SYSDATE, CURRENT_DATE, SYS_TIMESTAMP, CURRENT_TIMESTAMP, etc
 - USERENV / SYS_CONTEXT (with non-constant variables)
 - SYS_GUID
- Query must retrieve the most current committed state of the data
 - No Active Transaction Against Objects in Current Session



Example 2 – PL/SQL Function

- Stores Results of Function by Parameter
- Automatically Refreshed Based on Object Usage
- Enabled Using “result_cache” Option

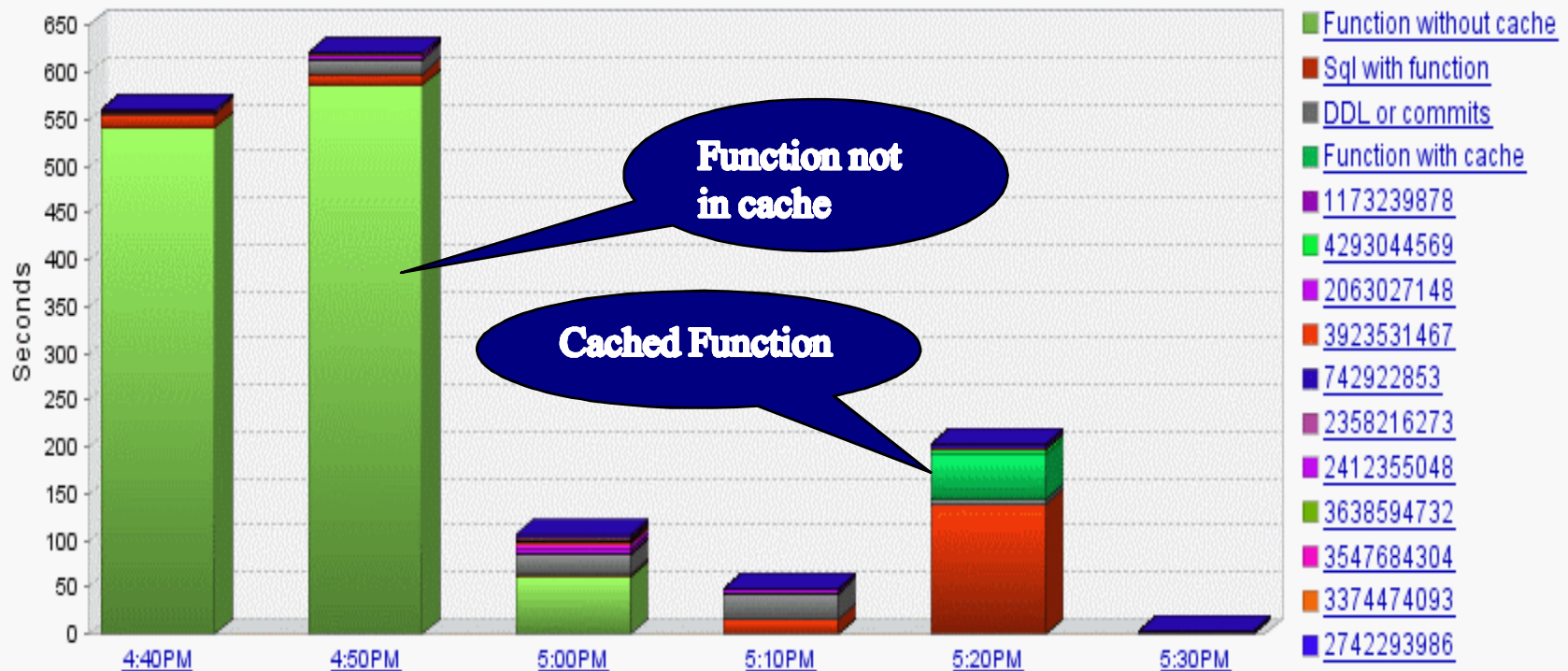
```
create or replace function rep_sales_totals (p_rep in number)
Return number result_cache as
  l_order_total number;
Begin
  select sum(order_total) into l_order_total from order_history
  where to_number(order_date,'YYYYMM') between 200901 and
    200903
  and sales_rep_id = p_rep;
  return l_order_total;
End;
```



PL/SQL Function Result Cache

Highest Total Execution Time | Long Running (Average) | Most Executed

Queries causing the most user wait time in the last hour (sum of all execution times)



```
SELECT sales_rep_id,  
       rep_sales_totals(sales_rep_id)  
FROM orders  
GROUP BY sales_rep_id
```



PL/SQL Function Result Cache

SQL: Sql with function (3784385302) [Name SQL](#)

Statistics			
Executions	23	Rows Processed	221
Parses	23	Disk Reads	3,690,674
Sorts	0	Buffer Gets	3,713,706

SQL: Function without cache (856166700) [Name SQL](#)

Statistics			
Executions	234	Rows Processed	226
Parses	2	Disk Reads	3,408,498
Sorts	0	Buffer Gets	3,431,382

Using Result Cache:

```
-----+-----+-----
ID NAME                                VALUE
-----+-----+-----
 1 Block Size (Bytes)                   1024
 2 Block Count Maximum                   2112
 3 Block Count Current                    32
 4 Result Size Maximum (Blocks)          211
 5 Create Count Success                   10
 6 Create Count Failure                   0
 7 Find Count                             1300
 8 Invalidation Count                     0
 9 Delete Count Invalid                   0
10 Delete Count Valid                     0
```

SQL: Sql with function (3784385302) [Name SQL](#)

Statistics			
Executions	131	Rows Processed	1,310
Parses	131	Disk Reads	2,126,066
Sorts	0	Buffer Gets	2,131,000

SQL: Function with cache (2494512257) [Name SQL](#)

Statistics			
Executions	10	Rows Processed	10
Parses	1	Disk Reads	150,830
Sorts	0	Buffer Gets	302,948

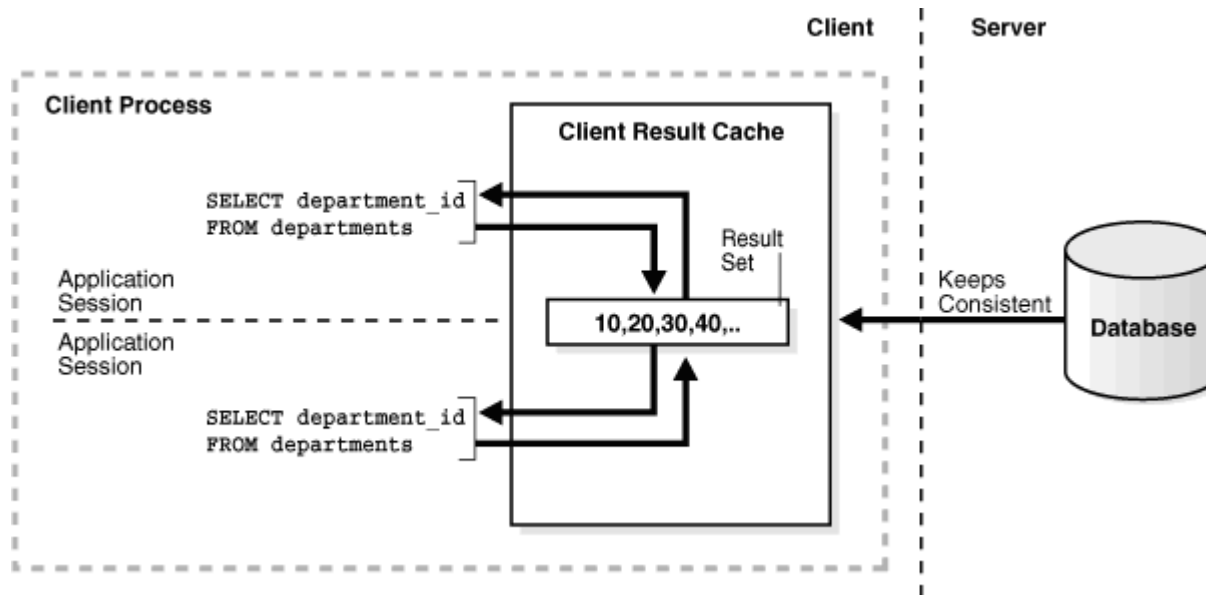


Benefits and Restrictions

- Similar Benefits as SQL Query Results Cache
- Works for Recursive Function Calls
- Restrictions
 - No invoker's rights or anonymous block
 - No pipelined table function
 - Does not reference dictionary tables, temporary segments, sequences or non-deterministic SQL functions
 - Has no OUT or IN OUT parameters
 - No IN parameters of type BLOB, CLOB, NCLOB, REF CURSOR, Collection, Object, Record
 - The Return Type is not a BLOB, NCLOB, REF CURSOR, Object, Record or collection using one of these



OCI Client Cache



- Must use an OCI driver that Supports Results Cache
- Must use 11g client and 11g server
- Shared by All Sessions in Client Process
- Subqueries and Query Blocks are not Cached
- Database will Invalidate Client Result Cache
- Independent of Server Result Cache



- **CLIENT_RESULT_CACHE_SIZE**
 - Maximum size of client result cache
 - 0 – 32767 (Disabled)
- **CLIENT_RESULT_CACHE_LAG**
 - 3000 ms (Default)
 - Forces next statement execution to check for validations
- **Optional Client Parameter File (SQLNET.ORA)
Overrides Database Parameters**
 - OCI_RESULT_CACHE_MAX_SIZE
 - OCI_RESULT_CACHE_MAX_RSET_SIZE (bytes)
 - OCI_RESULT_CACHE_MAX_RSET_ROWS



- **CLIENT_RESULT_CACHE_STATS\$**
 - One row for every client using Result Cache
 - Cache Settings and Statistics
- **DBA_TABLES, ALL_TABLES, USER_TABLES**
 - Column to show if FORCE has been used



NAME	VALUE	CACHE_ID
Block Size	256	124
Block Count Max	256	124
Block Count Current	128	124
Hash Bucket Count	1024	124
Create Count Success	10	124
Create Count Failure	0	124
Find Count	12	124
Invalidation Count	8	124
Delete Count Invalid	0	124
Delete Count Valid	0	124

```
SELECT * FROM GV$SESSION_CONNECT_INFO WHERE CLIENT_REGID = <cache_id>;
```

- Look for high values of Find Count
- Look for low values
 - Create Count Failure
 - Delete Count Valid



11g R1 vs. R2

- R1 Memory Grows to Maximum Size but does not Automatically Free Memory
 - DBMS_RESULT_CACHE.FLUSH
- Latching Issues in R1
 - R1 Result Cache controlled by one latch
 - R2 controlled by many latches
- Better Table Annotation Support
 - PL/SQL required the RELIES_ON clause which is deprecated in R2
 - OCI Client Result Cache now supports table annotation



- Developer of Wait-Based Performance Tools
- Igniter Suite – Web Based and Agentless
 - Ignite for Oracle, SQL Server, DB2, Sybase
- Helps Identify Queries that may benefit from using Results Cache
- Based in Colorado, worldwide customers
- Free trial at www.confio.com



College Alumni Information & Addresses

```
SELECT p.CdbID, p.intSchoolID_fk, blnDeceased, blnLost, intViewFieldPref,
       strPrefix, strFirstName, strMiddleName, strLastName, strSuffix, strMaidenName,
       strNickName, n.strComments, blnHasMedia, g.strName, strCity, strLocation,
       strEmployerName, strWorkTitle, intNameFormatShort, intNameFormatNormal,
       t.intNameFormatLong, m.blnSuspended
FROM   tblCdbPeople p left outer join tblCdbSearchAddress a on
       a.CdbID = p.CdbID left outer join tblCdbNonMember n on
       n.CdbID = p.CdbID left outer join tblCdbMember m on
       m.CdbID = p.CdbID left outer join tblCdbMemberType t on
       t.intSchoolID_fk = p.intSchoolID_fk and
       t.strTypeCode = m.strTypeCode_fk left outer join tblLinkCdbPeopleToGroup lpg on
       lpg.CdbID = p.CdbID and lpg.blnPrimary = 1 left outer join school_classInfo g on
       g.aut_ID = lpg.intGroupID_fk left outer join tbl_college c on
       c.aut_collegeID = intCollegeID_fk left outer join tblCdbResume r on
       r.CdbID = p.CdbID
WHERE  p.intSchoolID_fk = @intSchoolID and
       ( t.blnHideMembers is null or t.blnHideMembers = 0) and
       ( blnOnlyLost = 0 or blnLost = 1) and
       ( cdTypeID is null or ( cdTypeID is not null and
         exists (select * from types where ID = t.autTypeID)))
```



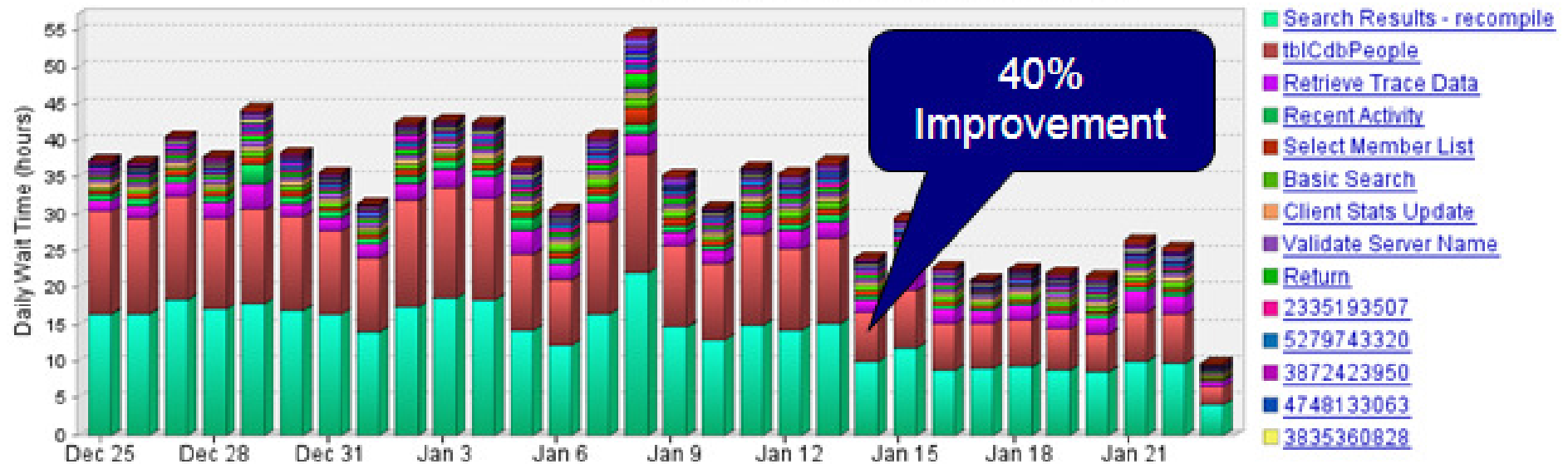
Example Of Performance

Day:

[Show Tips](#)

[SQL](#) [Waits](#) [Programs](#) [Databases](#) [Machines](#) [DB Users](#)

**Top 20 SQL Statements by Total Daily Wait | SQLPROD02
December 25, 2008 to January 23, 2009**



Change View: Total Wait Average Wait Typical Day

[Email Chart](#)

View Historical Charts for SQL:

[Show Full SQL Text](#)



CONFIO
SOFTWARE

Ignite IT Performance™



CONFIO
SOFTWARE

Ignite IT Performance™