

## Oracle 11g Results Cache

**Dean Richards**  
**Senior DBA, Confio Software**



### Who Am I?

- Senior DBA for Confio Software
  - [DeanRichards@confio.com](mailto:DeanRichards@confio.com)
- 20+ Years in Oracle, SQL Server
- 5+ Years in Oracle Consulting - SPG
- Specialize in Performance Tuning
- Review Performance of 100's of Databases for Customers and Prospects



- **Server Results Cache**
  - SQL Query Results
  - Remote Result Cache
  - PL/SQL Function Results
- **Client Results Cache**
  - OCI Client Results

3

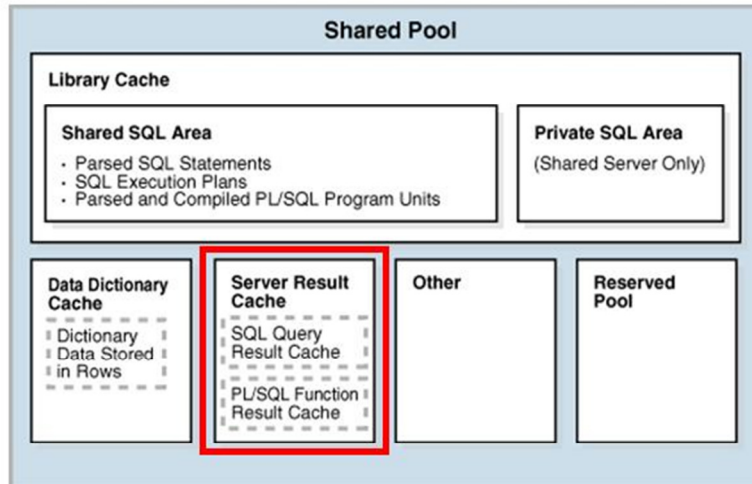


- **New in Oracle 11g, Improved in 11gR2**
- **Caches Results**
  - Buffer Cache caches blocks
- **Similar to Materialized View**
  - Without the headaches
  - Stale Data
  - Result Cache Automatically refreshes
- **Automatically marked stale if underlying data is changed**
- **Can Dramatically Increase Performance**

4



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



5



- **RESULT\_CACHE\_MODE**
  - MANUAL (default) – requires a query hint or table annotation
  - FORCE – every result set is cached. Not recommended because it can create 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)

Demo p1.sql

6

```
Command Prompt - sqlplus dean/dean

C:\Users\drichards\Documents\Ppresentations\Results Cache\Demo>sqlplus dean/dean

SQL*Plus: Release 11.1.0.7.0 - Production on Wed Sep 15 11:49:01 2010

Copyright (c) 1982, 2008, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @p1_
```

```
p1 - Notepad

File Edit Format View Help

SQL> show parameter result_cache

NAME                                TYPE                                VALUE
-----                                -                                -
client_result_cache_lag              big integer                         3000
client_result_cache_size             big integer                         0
result_cache_max_result              integer                             5
result_cache_max_size                big integer                         1792K
result_cache_mode                     string                             MANUAL
result_cache_remote_expiration        integer                             0
SQL> show parameter memory

NAME                                TYPE                                VALUE
-----                                -                                -
hi_shared_memory_address              integer                             0
memory_max_target                     big integer                         700M
memory_target                         big integer                         700M
shared_memory_address                 integer                             0
SQL> select 700*1024*.0025 result_cache_max_size_KB from dual;

RESULT_CACHE_MAX_SIZE_KB
-----
1792

SQL> select 700*1024*.0025*.05 result_cache_max_result_KB from dual;

RESULT_CACHE_MAX_RESULT_KB
-----
89.6

SQL> spool off
```





- Oracle Error Says AUTO Mode is Supported

```
SQL> alter system set result_cache_mode=bogus;  
alter system set result_cache_mode=bogus  
*  
ERROR at line 1:  
ORA-00096: invalid value BOGUS for parameter  
      result_cache_mode, must be  
from among FORCE, MANUAL, AUTO
```

- AUTO is not a valid setting according to Oracle Docs,  
maybe someday



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

Demo r1.sql

```
File Edit Format View Help
SQL> exec dbms_result_cache.flush
PL/SQL procedure successfully completed.
Elapsed: 00:00:00.00
SQL> set autotrace on
SQL> SELECT /*+ result_cache */ state, sum(order_total)
2 FROM order_history
3 WHERE order_date BETWEEN '1-JAN-09' AND '1-APR-09'
4 GROUP BY state
5 /
ST SUM(ORDER_TOTAL)
-----
AK      3061490.33
MN      3047050.77
NJ      3034419.28
OK       3048450.8
SD      3020848.36
WV      3025016.86
AL      3035956.74
CA      3032724.78
ND      3041623.26
WY      3057646.17
AR      3036405.77
```

```
File Edit Format View Help
51 rows selected.
Elapsed: 00:00:03.45
Execution Plan
-----
Plan hash value: 1508661739

-----
| Id | Operation          | Name                                | Rows | Bytes | Cost (%CPU)| Time |
-----
|  0 | SELECT STATEMENT   |                                     |    51 |   714 |    8718  (4)| 00:01:45 |
|  1 |  RESULT CACHE      | 3t3u7kj9w71v21dwar9vp1rwx         |      |      |             |      |
|  2 |    HASH GROUP BY   |                                     |     51 |   714 |    8718  (4)| 00:01:45 |
|*  3 |      FILTER        |                                     |      |      |             |      |
|*  4 |        TABLE ACCESS FULL | ORDER_HISTORY                     | 2802K |   37M |    8635  (3)| 00:01:44 |
-----

Predicate Information (identified by operation id):
-----
```

```
File Edit Format View Help
|*  4 |        TABLE ACCESS FULL | ORDER_HISTORY                     | 2802K |   37M |    8635  (3)| 00:01:44 |
-----

Predicate Information (identified by operation id):
-----

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

Result Cache Information (identified by operation id):
-----

1 - column-count=2; dependencies=(DEAN.ORDER_HISTORY); parameters=(nls); name="SELECT /*+ result_cache */ state, sum
(order_total)
FROM order_history
WHERE order_date BETWEEN '1-JAN-09' AND '1-APR-09'
GROUP "
```

```

File Edit Format View Help
1 - column-count=2; dependencies=(DEAN.ORDER_HISTORY); parameters=(nls); name="SELECT /*+ result_cache */ state, sum
(order_total)

FROM    order_history
WHERE   order_date BETWEEN '1-JAN-09' AND '1-APR-09'
GROUP  "

Statistics
-----
      0 recursive calls
      0 db block gets
    31138 consistent gets
    31133 physical reads
      0 redo size
    1815 bytes sent via SQL*Net to client
     553 bytes received via SQL*Net from client
      5 SQL*Net roundtrips to/from client
      0 sorts (memory)
      0 sorts (disk)
     51 rows processed

```

```

File Edit Format View Help

Statistics
-----
      0 recursive calls
      0 db block gets
    31138 consistent gets
    31133 physical reads
      0 redo size
    1815 bytes sent via SQL*Net to client
     553 bytes received via SQL*Net from client
      5 SQL*Net roundtrips to/from client
      0 sorts (memory)
      0 sorts (disk)
     51 rows processed

SQL> /
ST SUM(ORDER_TOTAL)
-----
AK      3061490.33
MN      3047050.77
NT      3034410.78

```

```

File Edit Format View Help

MS      303987.487
NV      303120.1389
TX      303052.
VT      3055542.77
AZ      3025923.57
KY      3046502.57
NC      3056994.51
OR      3020279.56
PA      3020279.56
51 rows 3025449.22

51 rows selected.
Elapsed: 00:00:00.18
Execution Plan
-----
Plan hash value: 1508661739

-----
| Id | Operation          | Name | Rows | Bytes | Cost (%CPU)| Time |
-----

```

File	Edit	Format	View	Help
Result Cache Information (identified by operation id):				
-----				
1 - column-count=2; dependencies=(DEAN.ORDER_HISTORY); parameters=(nls); name="SELECT /*+ result_cache */ state, sum(order_total)				
FROM order_history				
WHERE order_date BETWEEN '1-JAN-09' AND '1-APR-09'				
GROUP "				
Statistics				
-----				
0 recursive calls				
0 db block gets				
0 consistent gets				
0 physical reads				
0 redo size				
1815 bytes sent via SQL*Net to client				
553 bytes received via SQL*Net from client				



# Execution Plan



CONFIO  
SOFTWARE

Ignite IT Performance™

## Execution Plan

Plan hash value: 47235625

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		44	396	6854 (2)	00:01:23
1	RESULT CACHE	7zvt0xan8bw0pgry071f7mt85s				
2	HASH GROUP BY		44	396	6854 (2)	00:01:23
3	TABLE ACCESS FULL	T	1739K	14M	6802 (1)	00:01:22

## Result Cache Information (identified by operation id):

```
1 - column-count=2; dependencies=(PROD.ORDER_HISTORY); parameters=(nls);
name="SELECT /*+ result_cache */ state, sum(order_total) FROM order_history GROUP BY state"
```

```
SELECT ID, TYPE, CREATION_TIMESTAMP, BLOCK_COUNT, COLUMN_COUNT, PIN_COUNT, ROW_COUNT
FROM V$RESULT_CACHE_OBJECTS
WHERE CACHE_ID = '7zvt0xan8bw0pgry071f7mt85s';
```

ID	TYPE	CREATION_	BLOCK_COUNT	COLUMN_COUNT	PIN_COUNT	ROW_COUNT
2	Result	06-JAN-10	1	2	0	12



- Query Hint

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

- Session Mode

```
alter session set result_cache_mode = force
```

- Table Annotation Mode

```
alter table order_history result_cache (mode force)
```

- Database Setting

```
result_cache_mode = FORCE (not recommended)
```

10



- Access Large Amount of Data
- Return Few Rows
- Execute Somewhat Frequently
- Based on Slowly Changing Data
- Limited Number of Bind Values
  - Results are cached by Bind Value
- Notes
  - Also Works with Flashback Queries
  - RAC Instances have their own Result Cache

11





- Highly Selective Queries (PK Lookups)
- Returns a lot of Rows (Large Result Set)
  - Default is 5% of result\_cache\_max\_size
  - 90KB on my small laptop database
  - Relative to the number of rows read
- Execute Infrequently
  - Queries from monthly batch jobs are probably not good candidates
- Underlying tables change frequently

12



## Examples

- HR Data
  - Employees – unless you have a lot of hiring/firing
  - Paychecks – every 2-4 weeks
- Historical Info
  - Sales History
- Product Data
  - Unless you add products a lot
  - May not be large enough for some companies

13



- 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
  - SYS\_GUID
- Query must retrieve the most current committed state of the data
  - No Active Transaction Against Objects in Current Session

14



- Hope that result cache dependencies are by table partition
- Does not work that way
  - The query is dependent on the whole table
  - Hopefully it will someday

Demo part1.sql

15

```

SQL> set echo on;
SQL> CREATE TABLE date_part
2 ( x int,
3   y int,
4   z DATE
5 )
6 PARTITION BY RANGE (z)
7 (
8   PARTITION part_2010_q1 VALUES LESS THAN(to_date('01-apr-2010','dd-mon-yyyy')),
9   PARTITION part_2010_q2 VALUES LESS THAN(to_date('01-jul-2010','dd-mon-yyyy')),
10  PARTITION part_2010_q3 VALUES LESS THAN(to_date('01-oct-2010','dd-mon-yyyy')),
11  PARTITION part_2010_q4 VALUES LESS THAN(to_date('01-jan-2011','dd-mon-yyyy'))
12 )
13 /

```

Table created.

Elapsed: 00:00:00.02

```

SQL>
SQL> declare
2   imonth integer;
3   iday integer;
4   iyear integer := 2010;
5   begin
6     for i in 1 .. 16567 loop
7       imonth := trunc(dbms_random.value(1,12));
8       iday := trunc(dbms_random.value(1,28));
9       insert into date_part values (
10        i, trunc(dbms_random.value(0,50)),
11        to_date(imonth||'/'||iday||'/'||iyear, 'mm/dd/yy'));
12     end loop;
13   commit;
14 end;
15 /

```

PL/SQL procedure successfully completed.

Elapsed: 00:00:01.19

```
SQL> exec dbms_stats.gather_table_stats('DEAN','DATE_PART')
```

PL/SQL procedure successfully completed.

```

12   end loop;
13   commit;
14 end;
15 /

```

PL/SQL procedure successfully completed.

Elapsed: 00:00:01.19

```
SQL> exec dbms_stats.gather_table_stats('DEAN','DATE_PART')
```

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.17

```
SQL> set autotrace on
```

```
SQL> select /*+ result_cache */ count(1)
2 from date_part partition(part_2010_q2)
3 /
```

COUNT(1)

4455

Elapsed: 00:00:00.01

Execution Plan

Plan hash value: 4112570612

Id	Operation	Name	Rows	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT		1	5 (0)	00:00:01		

Id	Operation	Name	Rows	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT		1	5 (0)	00:00:01		
1	RESULT CACHE	bxqq71560zj9j6c5qcr2t02868					
2	SORT AGGREGATE		1				
3	PARTITION RANGE SINGLE		4455	5 (0)	00:00:01	2	2
4	TABLE ACCESS FULL	DATE_PART	4455	5 (0)	00:00:01	2	2

Result Cache Information (identified by operation id):

```

1 - column-count=1; dependencies=(DEAN,DATE_PART); attributes=(single-row); name="select /*+ result_cache */ count(1)
from date_part partition(part_2010_q2)"

```

Statistics

```

1 recursive calls
0 db block gets

```



- **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

Demo m1.sql

16

```
SQL> SET SERVEROUTPUT ON
SQL> EXECUTE DBMS_RESULT_CACHE.MEMORY_REPORT
Result Cache Memory Report

[Parameters]
Block Size           = 1K bytes
Maximum Cache Size   = 1792K bytes (1792 blocks)
Maximum Result Size  = 89K bytes (89 blocks)

[Memory]
Total Memory = 141256 bytes [0.048% of the Shared Pool]
... Fixed Memory = 5296 bytes [0.002% of the Shared Pool]
... Dynamic Memory = 135960 bytes [0.046% of the Shared Pool]
..... Overhead = 103192 bytes
..... Cache Memory = 32K bytes (32 blocks)
..... Unused Memory = 27 blocks
..... Used Memory = 5 blocks
..... Dependencies = 2 blocks (2 count)
..... Results = 3 blocks
..... SQL = 2 blocks (2 count)
..... Invalid = 1 blocks (1 count)

PL/SQL procedure successfully completed.
Elapsed: 00:00:00.07
SQL> spool off
```



- **V\$RESULT\_CACHE\_STATISTICS**
  - How well is the cache doing?
  - Monitor CREATES vs. FINDS
- **V\$RESULT\_CACHE\_MEMORY**
  - Memory components and statistics
- **V\$RESULT\_CACHE\_OBJECTS**
  - Objects that are in the cache along with attributes
- **V\$RESULT\_CACHE\_DEPENDENCY**
  - Dependencies of the results in cache

Demo s1.sql

17



- **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**

18





- Stores Results of Function by Parameter
- Automatically Refreshed Based on Object Usage
- Enabled Using "result\_cache" Option

```
create or replace function state_sales_totals (p_state in varchar2)
return number result_cache as
    l_order_total number;
begin
    select sum(order_total) into l_order_total from order_history
    where order_date BETWEEN '1-JAN-09' AND '1-JAN-10'
    and state = p_state;
    return l_order_total;
end;
```

Demo f1.sql

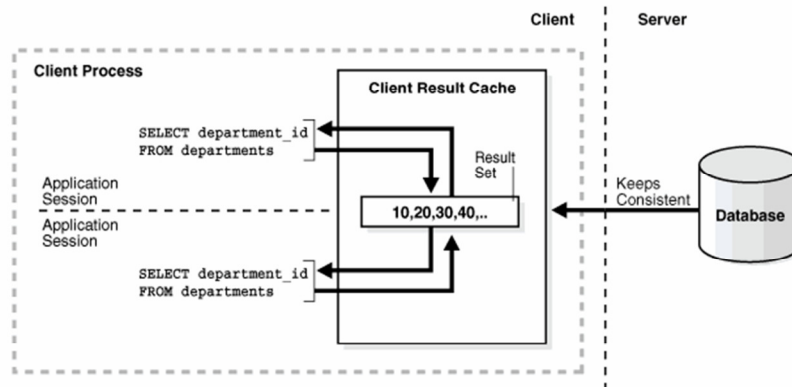
19



## Benefits and Restrictions

- Similar Benefits as SQL Query Results Cache
- 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

20



- 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

21



- **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

22



- **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

23



## CLIENT\_RESULT\_CACHE\_STATS\$

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

24



- **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
  - May have been predicated by low use case examples of Result Cache
- **Better Table Annotation Support**
  - PL/SQL required the RELIES\_ON clause which is deprecated in R2
  - OCI Client Result Cache now supports table annotation

25



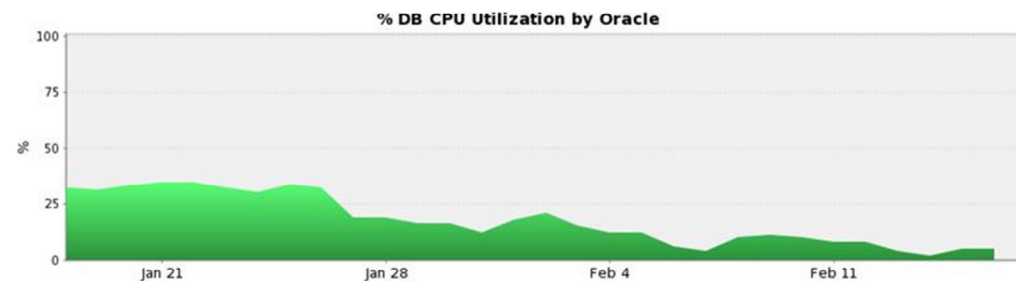
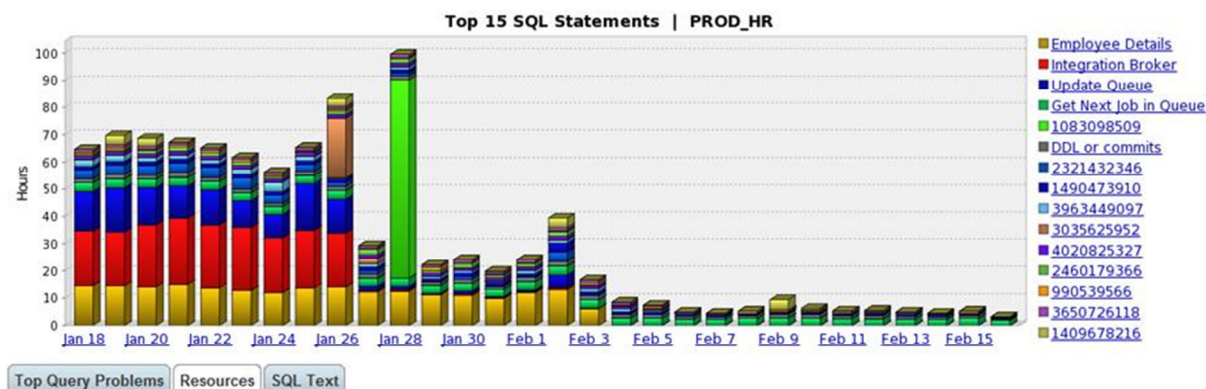
- **Two Types of Result Cache**
  - Server – SQL Queries, Functions
  - Client – OCI Applications
- **Can use Server Result Cache**
  - Manually via /\*+ result\_cache \*/ hint
  - Table annotation
  - Result\_cache\_mode=force – not recommended
  - Functions
- **Huge Performance Gains**
  - Make sure the underlying data is appropriate

26





27

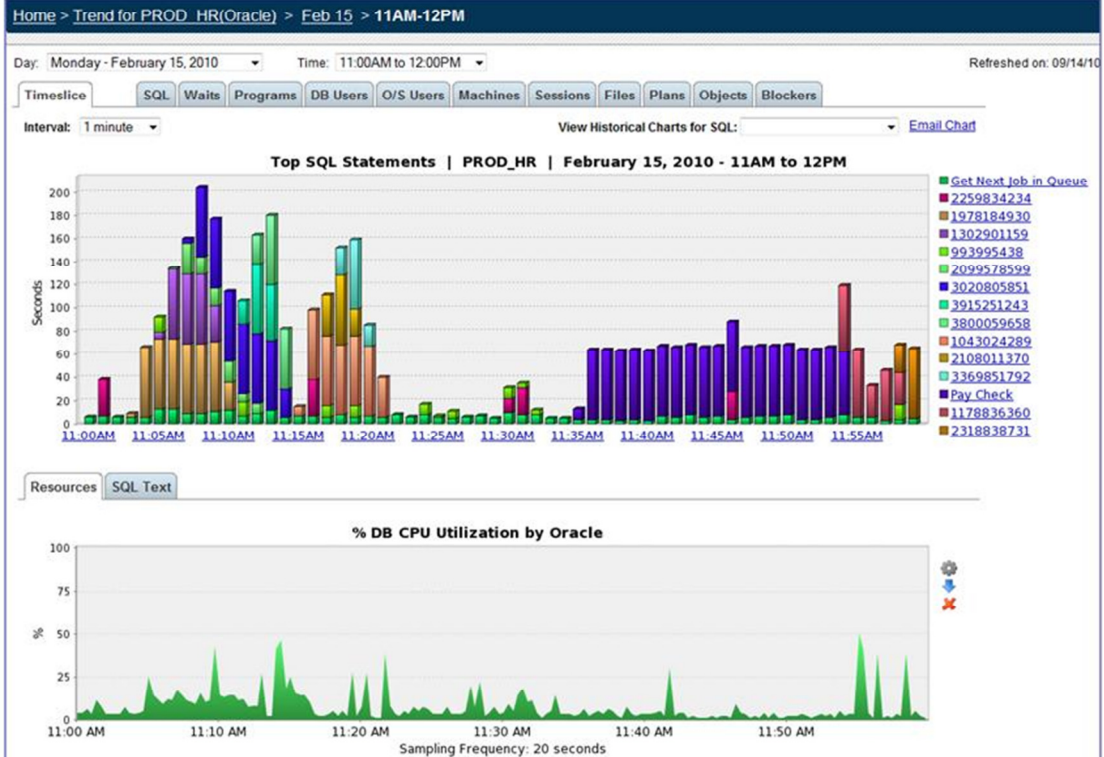


28





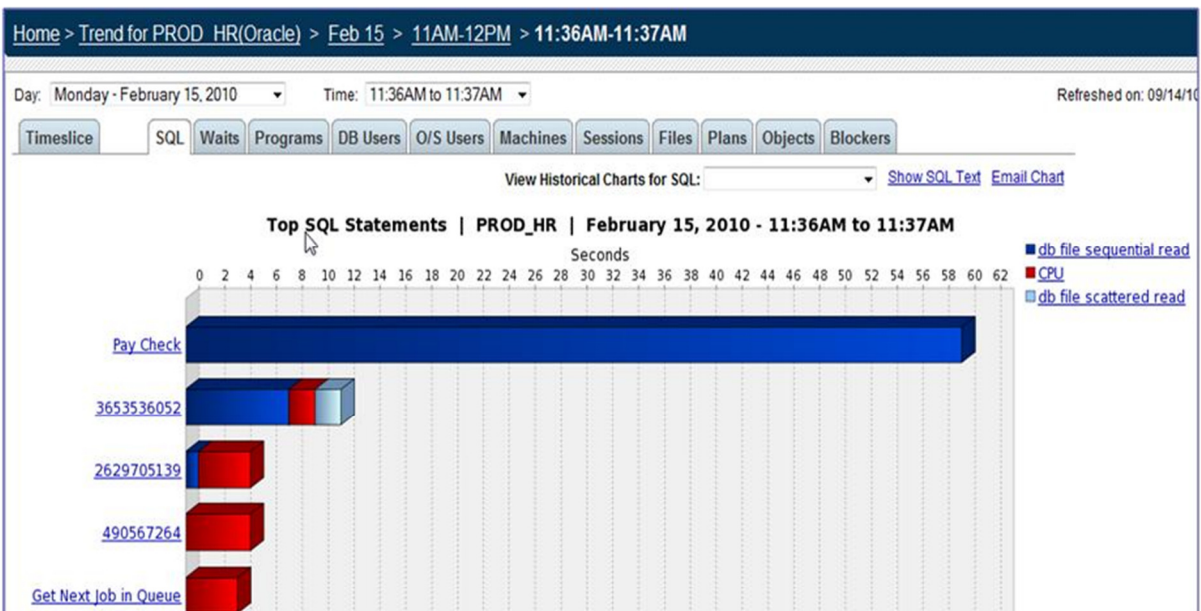
# Drill Down



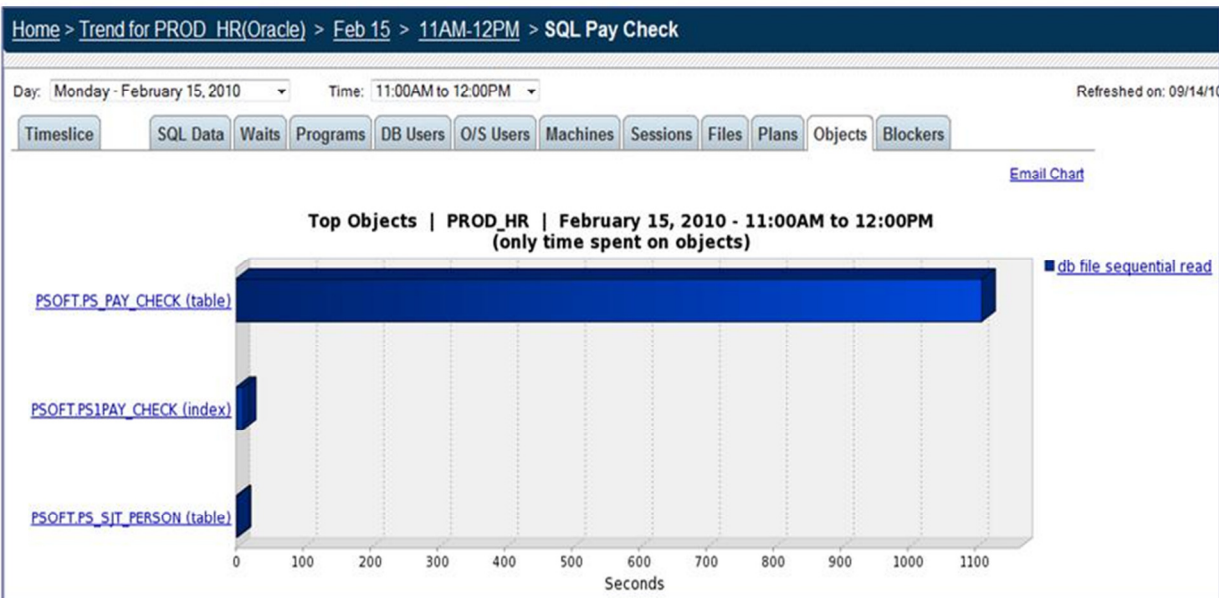
29



# What are you Waiting For



30



31



- Award Winning Performance Tools
- Ignite8 for Oracle, SQL Server, DB2, Sybase
- Uses Response Time Analysis (RTA)
  - Which SQL Statement Should I Tune
  - Why is it Slow – Wait Events, Plans, Metrics
- Based in Colorado, worldwide customers

Download free trial at  
[www.confio.com](http://www.confio.com)

32