Database Time-Based Performance Tuning
From Theory to Practice

Graham Wood
Architect
Real-World Performance, Server Technologies
October 1, 2014
Real-World Performance Root Causes

The database is not being used as it was designed to be used

The application architecture/code design is sub-optimal

There is a sub optimal algorithm in the database
Program Agenda

1. DB Time and Average Active Sessions
2. The DB Time Method
3. Tools and What’s New
Program Agenda

1. DB Time and Average Active Sessions
2. The DB Time Method
3. Tools and What’s New
Why Do We Care About Time?

• Human time is critical to the enterprise

• Systems performance affects business goals
  – Human time + technology resource time

• “Time is money”

• Performance improvement means doing things faster

Performance is always and only about time
Database Time (DB Time)

- Total time in database calls by foreground sessions
- Includes CPU time, IO time and non-idle wait time
- DB Time <> response time
- Common currency for Oracle performance analysis

*Database time is total time spent by user processes either actively working or actively waiting in a database call.*
A Single Session

Single session with Database Black Box server

- Browse Books
- Read Reviews For One Book
- Add to Cart
- Checkout

= time spent in database

TIME
Fundamental Concepts

**Database Time (DB Time) =**
Total time session spent in all database calls

**Active Session =**
Session currently spending time in a database call

**Average Activity of the Session (% Activity) =**
The ratio of time active to total wall clock time
Session Details: 1869 (AFOHERG)

Collected From Target  Oct 30, 2007 9:43:37 AM CDT

Detail for Selected 5 Minute Interval

Start Time:  Oct 30, 2007 9:17:05 AM

Activity (%)

SQL ID  SQL Command  Plan Hash Value  Module  Action  Client ID
100.00  gkmd7wuxz1na0  SELECT  84730335  oraclealan@ap103fam (TNS V1-V3) AFOHERG
Multiple Sessions

DB Time = Sum of DB Time Over All Sessions

Avg. Active Sessions = Sum of Avg. Activity Over All Sessions

At time $t$ we have 2 active sessions

$\text{time spent in database}$
Visualizing DB Time

Avg. Active Sessions =

Total Database Time

Wall Clock (Elapsed) Time

Active Sessions over time

User 1
User 2
User 3
User n

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.
EM Performance Page

- Active Sessions by wait class over time
- Colored area = amount of DB time
- “Click on the big stuff”
Average active sessions

= \text{DB time} / \text{elapsed time}

• Full-time equivalent sessions
  – Not whole sessions
  – How many full-time virtual sessions to do the work?

• Comparable
  – Across systems
  – Across time periods
System Variables and DB Time

- More users
  - => More calls
    • => DB time increases

- Larger transactions
  - => Longer calls
    • => DB time increases

- IO performance degrades
  - => IO time increases
    • => DB time increases

- Application performance degrades
  - => Wait time increases
    • => DB time increases

*DB time increases as system load increases.*
Host Performance and DB Time

• Host is CPU-bound
  – => foregrounds accumulate active run-queue time
    • => wait event times are artificially inflated
      – => DB time increases

_Tune for CPU before waits when CPU constrained_
CPU Run-queue and DB Time

DB time is inflated when host is CPU-bound
Click on an area of a graph or legend to get more detail.

**CPU or I/O problem?**
Instrumentation: Where to find DB Time?

• **V$SYS_TIME_MODEL, V$SESS_TIME_MODEL**
  – STAT_NAME = ‘DB time’

• **V$SYSMETRIC_HISTORY**
  – “Database Time Per Second”, “CPU Usage Per Sec”
  – 11g new metric “Average Active Sessions”

• **V$SQL**
  – ELAPSED_TIME and CPU_TIME
  – Wait class times: APPLICATION, CONCURRENCY, CLUSTER, USER_IO

• **V$ACTIVE_SESSION_HISTORY**
Active Session History (ASH)

• All ‘Active’ sessions captured every second
  – Foregrounds and backgrounds are sampled
  – Active foregrounds contribute to DB Time
  – Many dimensions captured

• In-memory: V$ACTIVE_SESSION_HISTORY
  – Sampling interval = 1 second

• On-disk: DBA_HIST_ACTIVE_SESS_HISTORY
  – Sampling interval = 10 second

• ASH is a system-wide record of database activity
ASH Math

\[ \text{COUNT}(*) = \text{DB Time} \]

\[ \text{GROUP BY ?} \]
ASH Math: COUNT(*)=DB Time

• ASH is a big fact table
  – Each row represents 1-second of active session time

• V$ACTIVE_SESSION_HISTORY
  – COUNT(*) = DB time in seconds

• DBA_HIST_ACTIVE_SESS_HISTORY
  – COUNT(*) * 10 = DB time in seconds
Example: DB Time by SQL ID

```sql
select sql_id,
    count(*) DBTime,
    round(count(*)*100/sum(count(*))
        over (), 2) pctload
from v$active_session_history
where sample_time > sysdate - 1/24/60
    and session_type <> 'BACKGROUND'
group by sql_id
order by count(*) desc;
```
Example: DB Time by SQL ID

```sql
select sql_id,
       count(*) DBTime,
       round(count(*)*100/sum(count(*))
       over (), 2) pctload
from v$active_session_history
where sample_time > sysdate - 1/24/60
  and session_type <> 'BACKGROUND'
group by sql_id
order by count(*) desc;
```

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>DBTIME</th>
<th>PCTLOAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>6bmxrabnwswxd</td>
<td>60</td>
<td>63.83</td>
</tr>
<tr>
<td>azzsynmz43nrr</td>
<td>8</td>
<td>8.51</td>
</tr>
<tr>
<td>28pb73sbwhmm8</td>
<td>5</td>
<td>5.32</td>
</tr>
<tr>
<td>58psyvgau23s2</td>
<td>3</td>
<td>3.19</td>
</tr>
<tr>
<td>amrq8hk767tuz</td>
<td>2</td>
<td>2.13</td>
</tr>
<tr>
<td>2r5qhb3fb63vm</td>
<td>1</td>
<td>1.06</td>
</tr>
<tr>
<td>f3919usqp5wj2</td>
<td>1</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Example: DB Time by SQL ID
Avg Active Sessions and DB Time

Active sessions

ASH sample count is value of active sessions function at sample times

DB time is area under curve

$\Delta t = 1$ sec

DB Time

t0
t1
Top Activity

Drag the shaded box to change the time period for the detail section below.

Average Active Sessions

Instance Disk I/O
Where is DB Time used?

- ADDM

- EM Performance page and drill downs

- ASH report

- AWR and AWR Warehouse

- SQL Monitor and Perfhub
Program Agenda

1. DB Time and Average Active Sessions
2. The DB Time Method
3. What’s New
The DB Time Method: Short Course

or

just ask ADDM
The DB Time Method

1. Identify performance issue
2. Scope the issue
3. Set goals
4. Data capture (NO OP)
5. Investigate DB time distribution
   - Identify the largest potential for improvement
6. Modify system to tune for largest gain
7. Evaluate against goals
   - Repeat from step 4 if goals not met
Investigate DB Time Distribution

• Identify uneven distributions of DB time (skew)
  – Largest potential improvement within scope
  – If CPU bound tune for CPU first!

• System scope:
  – Resource limits – is problem outside the DB?

• Application scope:
  – Service, module, action
  – Resource contention (e.g. latches)
  – SQLID, rowsource

• Session scope
  – Long running SQL
  – Resource contention (e.g. enqueues)
Drag the shaded box to change the time period for the detail section below.

**Detail for Selected 5 Minute Interval**

**Start Time:** Oct 31, 2007 3:29:12 PM CDT

### Top SQL

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>SQL Type</th>
<th>Activity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>305y6g3wwbh</td>
<td>SELECT</td>
<td>1.57</td>
</tr>
<tr>
<td>86k3y24v910b</td>
<td>SELECT</td>
<td>6.72</td>
</tr>
<tr>
<td>22f3yatcc8bd</td>
<td>SELECT</td>
<td>5.66</td>
</tr>
<tr>
<td>93sgq7vng35x</td>
<td>SELECT</td>
<td>5.36</td>
</tr>
<tr>
<td>4.95</td>
<td>SELECT</td>
<td>4.95</td>
</tr>
<tr>
<td>4.95</td>
<td>SELECT</td>
<td>4.95</td>
</tr>
<tr>
<td>3.86</td>
<td>SELECT</td>
<td>3.86</td>
</tr>
</tbody>
</table>

### Top Sessions

<table>
<thead>
<tr>
<th>Activity (%)</th>
<th>Session ID</th>
<th>User Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.54</td>
<td>1996</td>
<td><a href="mailto:perl@atgebs.us.oracle.com">perl@atgebs.us.oracle.com</a></td>
<td>(TNS V1-V3)</td>
</tr>
<tr>
<td>9.32</td>
<td>1728</td>
<td><a href="mailto:oracledb92@iasbde.us.oracle.com">oracledb92@iasbde.us.oracle.com</a></td>
<td>(TNS V1-V3)</td>
</tr>
<tr>
<td>6.38</td>
<td>2251</td>
<td>ARUDAS</td>
<td>JDBC Thin Client</td>
</tr>
<tr>
<td>5.72</td>
<td>1682</td>
<td>BUGPATCH</td>
<td></td>
</tr>
<tr>
<td>5.66</td>
<td>1570</td>
<td>MFGOPSMT</td>
<td></td>
</tr>
<tr>
<td>5.36</td>
<td>2047</td>
<td>MFGOPSMT</td>
<td></td>
</tr>
<tr>
<td>4.62</td>
<td>1895</td>
<td>TOGEORGE</td>
<td></td>
</tr>
<tr>
<td>3.96</td>
<td>1936</td>
<td>JSARICOS</td>
<td>OMS</td>
</tr>
</tbody>
</table>

*DB time*
Modify System

- Start with the largest DB time issues first
  - Address root causes, not symptoms
- Match solution scope to problem scope
  - Scope mismatch causes many problems
- Proceed iteratively, one fix at a time
  - Concurrent fixes should be orthogonal
- Measure and validate results at each successive step
- Stop when goals are met
The DB Time Method: Advantages

• Tunes the one thing that affects users: Time
• Data capture scoping not necessary
  – ‘Always on’ data collection
  – No requirement to reproduce problem
• DB time based diagnosis removes value judgments
  – Scientific method, not sorcerer’s magic
• Works for concurrency problems such as locking

Performance improvement means doing the same work in less DB Time
Program Agenda

1. DB Time and Average Active Sessions
2. The DB Time Method
3. Tools and What’s New
AWR Reporting
AWR, ASH, Compare period etc

**WORKLOAD REPOSITORY report for**

<table>
<thead>
<tr>
<th>DB Name</th>
<th>DB Id</th>
<th>Instance</th>
<th>Inst num</th>
<th>Startup Time</th>
<th>Release</th>
<th>RAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHOPPING</td>
<td>1722515804</td>
<td>SHOPPING02</td>
<td>2</td>
<td>07-Oct-13 16:10</td>
<td>11.2.0.2.0</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host Name</th>
<th>Platform</th>
<th>CPUs</th>
<th>Cores</th>
<th>Sockets</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nq11db-na.bigetail.com</td>
<td>Linux x86 64-bit</td>
<td>64</td>
<td>32</td>
<td>4</td>
<td>504.03</td>
</tr>
</tbody>
</table>

**Report Summary**

**Cache Sizes**

<table>
<thead>
<tr>
<th>Buffer Cache</th>
<th>Begin</th>
<th>End</th>
<th>Std Block Size</th>
<th>16K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30.720M</td>
<td>30.720M</td>
<td></td>
<td>16K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shared Pool Size</th>
<th>Begin</th>
<th>End</th>
<th>Log Buffer</th>
<th>14.772K</th>
</tr>
</thead>
<tbody>
<tr>
<td>16,384M</td>
<td>16,384M</td>
<td>16,384M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Load Profile**

<table>
<thead>
<tr>
<th>DB Time(s)</th>
<th>Per Second</th>
<th>Per Transaction</th>
<th>Per Exec</th>
<th>Per Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>283.7</td>
<td>0.2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>38.0</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Redo Size</th>
<th>Logical reads</th>
<th>Block changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>9,929,605.5</td>
<td>802,177.1</td>
<td>44,854.5</td>
</tr>
<tr>
<td>7,222.2</td>
<td>589.4</td>
<td>33.0</td>
</tr>
</tbody>
</table>
AWR Reporting
AWR, ASH, Compare period etc

WORKLOAD REPOSITORY report for

<table>
<thead>
<tr>
<th>DB Name</th>
<th>DB Id</th>
<th>Instance</th>
<th>Inst num</th>
<th>Startup Time</th>
<th>Release</th>
<th>RAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>POCDB</td>
<td>2676171438</td>
<td>pocdb2</td>
<td>2</td>
<td>28-Aug-13 18:08</td>
<td>12.1.0.1</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host Name</th>
<th>Platform</th>
<th>CPUs</th>
<th>Cores</th>
<th>Sockets</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>db1/db2/db3</td>
<td>Linux x86 64-bit</td>
<td>52</td>
<td>16</td>
<td>2</td>
<td>252.41</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Snap Id</th>
<th>Snap Time</th>
<th>Sessions</th>
<th>Cursors/Session</th>
<th>Instances</th>
<th>Pluggable Databases Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin Snap</td>
<td>12-Sep-13 15:30:16</td>
<td>469</td>
<td>4.0</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>End Snap</td>
<td>12-Sep-13 16:00:27</td>
<td>458</td>
<td>4.2</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Elapsed</td>
<td>30.18 (mins)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB Time</td>
<td>431.57 (mins)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Report Summary

Top ADDM Findings by Average Active Sessions

<table>
<thead>
<tr>
<th>Finding Name</th>
<th>Avg active sessions of the task</th>
<th>Percent active sessions of finding</th>
<th>Task Name</th>
<th>Begin Snap Time</th>
<th>End Snap Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top SQL Statements</td>
<td>14.30</td>
<td>42.31</td>
<td>ADDM:2676171438_2_3287</td>
<td>12-Sep-13 15:30</td>
<td>12-Sep-13 16:00</td>
</tr>
<tr>
<td>Shared Pool Latches</td>
<td>14.30</td>
<td>17.18</td>
<td>ADDM:2676171438_2_3287</td>
<td>12-Sep-13 15:30</td>
<td>12-Sep-13 16:00</td>
</tr>
<tr>
<td>PLUSQL Execution</td>
<td>14.30</td>
<td>14.36</td>
<td>ADDM:2676171438_2_3287</td>
<td>12-Sep-13 15:30</td>
<td>12-Sep-13 16:00</td>
</tr>
<tr>
<td>Java Execution</td>
<td>14.30</td>
<td>7.00</td>
<td>ADDM:2676171438_2_3287</td>
<td>12-Sep-13 15:30</td>
<td>12-Sep-13 16:00</td>
</tr>
<tr>
<td>Under sized SQO</td>
<td>14.30</td>
<td>2.63</td>
<td>ADDM:2676171438_2_3287</td>
<td>12-Sep-13 15:30</td>
<td>12-Sep-13 16:00</td>
</tr>
</tbody>
</table>

Load Profile

<table>
<thead>
<tr>
<th></th>
<th>Per Second</th>
<th>Per Transaction</th>
<th>Per Exec</th>
<th>Per Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Time(s)</td>
<td>14.3</td>
<td>0.4</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>DB CPU(s)</td>
<td>9.7</td>
<td>0.3</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
ADDM

• Embedded expert system using the DB time method
  – Identifies root causes behind the symptoms

• Variably scoped:
  – Host to instance to SQL and even database block
  – Scoped to database for RAC (new in 11g)
  – Real-time ADDM with 12c

• Findings prioritized by impact on DB time
  – Finding history allows flexible time scoping
  – Directives can filter findings

• Recommendations by benefit (reduction) to DB time
Automatic Database Diagnostic Monitor (ADDM)

Database Activity

The icon selected below the graph identifies the ADDM analysis period. Click on a different icon to select a different analysis period.

TIP For an explanation of the icons and symbols used in this page, see the Icon Key

ADDM Performance Analysis

Task Name: ADDM:3132078998_1_1976

Task Owner: SYS
Average Active Sessions: 10.2
Period Start Time: Apr 4, 2008 1:00:31 AM PDT
Period Duration (minutes): 60
Instance: emtarget_emtarget1

Finding: Top SQL by DB Time

<table>
<thead>
<tr>
<th>Impact (%)</th>
<th>Occurrences (latest 24 hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>47.9</td>
<td>24 of 24</td>
</tr>
<tr>
<td>39.5</td>
<td>0 of 24</td>
</tr>
<tr>
<td>36.3</td>
<td>1 of 24</td>
</tr>
<tr>
<td>12.9</td>
<td>23 of 24</td>
</tr>
<tr>
<td>7.6</td>
<td>1 of 24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance Finding Details: Top SQL by DB Time

Finding Impact (Active Sessions) Impact (%) 52.8
Period Start Time Apr 4, 2008 12:00:04 PM PDT
Period Duration (minutes) 60.2
Final Finding SQL statements consuming significant database time were found.

Recommendations

Schedule SQL Tuning Advisor

Select All | Select None | Show All Details | Hide All Details

Select Details Category

<table>
<thead>
<tr>
<th>Hide</th>
<th>SQL Tuning</th>
</tr>
</thead>
</table>

Action Investigate the SQL statement with SQL_ID "6en44vwsmyknr" for possible performance improvements. View Tuning History

SQL Text:
```
select /* serial_guys *// p_brand, p_type, p_size, ...
```
SQL ID: 6en44vwsmyknr

Rationale SQL statement with SQL_ID "6en44vwsmyknr" was executed 4 times and had an average elapsed time of 1031 seconds.

<table>
<thead>
<tr>
<th>Hide</th>
<th>SQL Tuning</th>
</tr>
</thead>
</table>

Action Run SQL Tuning Advisor on the SQL statement with SQL_ID "4scj37y190kp". View Tuning History

SQL Text:
```
select /* big_guys *// /* NO_GBY_PUSHDOWN *// s_name, s_address ...
```
SQL ID: 4scj37y190kp

Benefit (%): 13.3

Action Run SQL Tuning Advisor on the SQL statement with SQL_ID "3sk746v545rp". View Tuning History

SQL Text:
```
select /* big_guys *// /* NO_GBY_PUSHDOWN *// s_name, s_address ...
```
SQL ID: 3sk746v545rp

Benefit (%): 10.2

Findings Path

Expand All | Collapse All
EM Real-time Interface

• Transient (sub-hour) or immediate time scope
  – Requires interactivity of UI

• ‘Click on the big stuff’
  – Data visualizations display skew directly

• Takes some expertise to separate symptoms from root causes
<table>
<thead>
<tr>
<th>Operation</th>
<th>Object</th>
<th>Object Type</th>
<th>Order</th>
<th>Rows</th>
<th>Size (KB)</th>
<th>Cost</th>
<th>Time (sec)</th>
<th>CPU Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td>71,662</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILTER</td>
<td></td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>BG.RPTHEAD</td>
<td>TABLE</td>
<td>9</td>
<td>1</td>
<td>0.172</td>
<td>71,662</td>
<td>557,628,710,561</td>
<td>74,441,376</td>
</tr>
<tr>
<td>BITMAP CONVERSION TO ROWIDS</td>
<td></td>
<td></td>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BITMAP AND</td>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BITMAP CONVERSION FROM ROWIDS</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORT ORDER BY</td>
<td>INDEX RANGE SCAN</td>
<td>INDEX</td>
<td>2</td>
<td>1</td>
<td>1,074</td>
<td>9</td>
<td></td>
<td>74,441,376</td>
</tr>
<tr>
<td>INDEX RANGE SCAN</td>
<td>BG.I_RPTDATE</td>
<td>INDEX</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>311,071,176</td>
</tr>
<tr>
<td>BITMAP CONVERSION FROM ROWIDS</td>
<td></td>
<td></td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>22,364</td>
</tr>
<tr>
<td>INDEX RANGE SCAN</td>
<td>BG.BG_ACCESS_UNQ</td>
<td>INDEX</td>
<td>4</td>
<td>1</td>
<td>0.016</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AWR Warehouse

- Warehouse dashboard tracking ETL jobs
- All AWR features available on long term AWR data
  - Performance page
  - AWR report
  - ASH analytics
  - Compare Period ADDM
  - Compare Period Report
- Integrated seamlessly into EM UI
- Zero runtime overhead on source databases
<table>
<thead>
<tr>
<th>Status</th>
<th>Duration</th>
<th>SQL ID</th>
<th>User</th>
<th>Parallel</th>
<th>Database Time</th>
<th>IO Requests</th>
<th>Start</th>
<th>Ended</th>
<th>SQL Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUSION</td>
<td>6h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>11:57 AM</td>
<td>10:05 AM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>4.8h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>6h</td>
<td>23h</td>
<td>23k</td>
<td>10:05 AM</td>
<td>T00X</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>7.6h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>7h</td>
<td>23h</td>
<td>23k</td>
<td>9:37 AM</td>
<td>9:55 AM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>8.2h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>8h</td>
<td>23h</td>
<td>23k</td>
<td>7:59 AM</td>
<td>8:17 AM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>SOLREP</td>
<td>8.1h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>8h</td>
<td>23h</td>
<td>23k</td>
<td>5:40 AM</td>
<td>5:58 AM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>4h16h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>4h16h</td>
<td>5</td>
<td>14</td>
<td>Sat Nov 13, 2010</td>
<td>12:00 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>52h8h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>4h16h</td>
<td>5</td>
<td>14</td>
<td>Sat Nov 13, 2010</td>
<td>12:00 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>7:45h</td>
<td>82623v26v0507B</td>
<td>FUSION</td>
<td>4h16h</td>
<td>5</td>
<td>14</td>
<td>Sat Nov 13, 2010</td>
<td>12:00 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>5.6h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>0.0h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>30.8h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>13.3h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>4h16h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>2h6.7h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>1h6.3h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>1h6.3h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>1h6.3h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>1h6.3h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
<tr>
<td>FUSION</td>
<td>1h6.3h</td>
<td>614120b600000w</td>
<td>FUSION</td>
<td>4h</td>
<td>23h</td>
<td>23k</td>
<td>4:13 PM</td>
<td>4:30 PM</td>
<td>WITH SAWTHID AS T0E24734.2042675484 AS s1, T00X</td>
</tr>
</tbody>
</table>

Mon Nov 15, 2010 3:54 PM
DB Time
• Instrumentation and reporting across many different scopes
• Allows engineering approach to performance diagnosis
• Built into new features of Oracle Database such as DBIM
ORACLE IS THE INFORMATION COMPANY