Scale-Up Your Use of the Advisor Framework

Roger Cornejo
Principal Oracle Performance Consultant
@OracleDBTuning

Oracle Open World - San Francisco – September 16 – 19 2019
RogerCornejo.com
Roger.D.Cornejo@gmail.com

❖ Computer Science degree
❖ ~34 yrs using Oracle (V4 - 1985)
   ❖ DevOps on large DB apps
   ❖ Enterprise Architect
   ❖ Software Project Manager
❖ Performance Tuning
   Past 10 years
   ❖ 300+ Production instances
   ❖ ~2500 applications

Events:
❖ Hotsos / RMOUG / IOUG / ECO / OOW19
Other Presentation Topics:

❖ Dynamic Oracle Performance Analytics Using Normalized Metrics

❖ Customized Database Tuning: Beyond the Standard AWR Tools
  ❖ OCOJ / HOTSOS 2017

❖ AWR Data Analysis for Database Problem Solving: Beyond Tuning
  ❖ ECO 2017  IOUG 2014

❖ Case Studies in Capacity Planning using AWR
  ❖ Various: RMOUG / ECO / OCOJ

❖ Scale-Up Your Use of the Advisor Framework  [for Performance Tuning]
  ❖ HOTSOS 2018  IOUG 2018 / 2019  ECO 2018

<Content genesis from extensive toolkit>
Scaling-Up Your Use of the Advisor Framework

❖ What is the Advisor Framework?
❖ Why scale-up your use of the Advisor Framework?
❖ Advisor tables
❖ What is stored in the advisor tables
❖ Querying the advisor framework
❖ Case Studies /
   Advisor framework in use
   in the tuning process
❖ Catalog of SQL Tools
❖ Questions
Licensing Restrictions

❖ Enterprise Edition

❖ Diagnostics + Tuning Pack

❖ Diagnostics is prerequisite of Tuning Pack
Scaling-Up Your Use of the Advisor Framework

❖ What is the Advisor Framework?
❖ Why scale-up your use of the Advisor Framework?
❖ Advisor tables
❖ What is stored in the advisor tables?
❖ Querying the advisor framework
❖ Case Studies
   Advisor framework in use
   in the tuning process
❖ Catalog of SQL Tools
❖ Questions
The Advisor Framework is …

❖ … Infrastructure to support Oracle’s database manageability
  Oracle: “… database, heal thy self …”

❖ Public interface:
  ❖ PL/SQL package DBMS_ADVISOR
  ❖ OEM (not covered in this presentation)

❖ Tuning related advisors are a subset of all advisors
  (some available since 9i)

❖ Focus of this topic:
  exploring the persisted information from
  the tuning related advisors
Reverse Engineering (RE) of the Advisor Framework

RE Definition for purposes intended
❖ What:
Understand design based on an in-depth examination of its structure and content.
❖ How:
Exposing content using simple SQL queries.
❖ Why:
Repurpose Advisor content for every day performance problem solving use-cases
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
❖ Why scale-up your use of the Advisor Framework?
❖ Advisor tables
❖ What is stored in the advisor tables?
❖ Querying the advisor framework
❖ Case Studies
  Advisor framework in use in the tuning process

❖ Catalog of SQL Tools
❖ Questions
Why scale-up your use of the Advisor Framework?

❖ Improved Tuning Capability:

1. Harvest information from periodic runs saves time: no need to re-run
   ❖ ADDM - **hourly** (each snapshot)
   ❖ Segment - daily
   ❖ SQL Tuning - daily

2. Quickly Assess run results from on-demand execution:
   ❖ SQL Access Advisor
   ❖ SQL Tuning Advisor

3. **Massively scale up** a DBA’s ability to **action** tuning opportunities

Custom SQL to Evaluate advice across many runs
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
✓ Why scale-up your use of the Advisor Framework?
❖ Advisor tables
❖ What is stored in the advisor tables?
❖ Querying the advisor framework
❖ Case Studies
❖ Advisor framework in use in the tuning process

❖ Scripts
❖ Questions
Views undergird the Advisor Framework

example:

```
select advisor_name from DBA_ADVISOR_DEFINITIONS
```

- ADDM
- SQL Tuning Advisor
- Segment Advisor
- SQL Access Advisor
- Undo Advisor
- SQL Repair Advisor
- SQL Performance Analyzer
- SQL Workload Manager
- Tune MView
- Compression Advisor
- SPM Evolve Advisor

See: `DBA_ADVISOR_DEF_PARAMETERS` for how the advisors are set-up
Advisor tables

```sql
select view_name from dba_views
where view_name like 'DBA_ADVISOR%'
; -- 41 Views
```

<table>
<thead>
<tr>
<th>VIEW_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA_ADVISOR_LOG</td>
</tr>
<tr>
<td>DBA_ADVISOR_JOURNAL</td>
</tr>
<tr>
<td>DBA_ADVISOR_TASKS</td>
</tr>
<tr>
<td>DBA_ADVISOR_EXECUTIONS</td>
</tr>
<tr>
<td>DBA_ADVISOR_FINDINGS</td>
</tr>
<tr>
<td>DBA_ADVISOR_RECOMMENDATIONS</td>
</tr>
<tr>
<td>DBA_ADVISOR_RATIONALE</td>
</tr>
<tr>
<td>DBA_ADVISOR_ACTIONS</td>
</tr>
<tr>
<td>DBA_ADVISOR_OBJECTS</td>
</tr>
<tr>
<td>DBA_ADVISOR_COMMANDS</td>
</tr>
<tr>
<td>DBA_ADVISOR_DEFINITIONS</td>
</tr>
<tr>
<td>DBA_ADVISOR_DEF_PARAMETERS</td>
</tr>
<tr>
<td>DBA_ADVISOR_DIR_DEFINITIONS</td>
</tr>
<tr>
<td>DBA_ADVISOR_DIR_INSTANCES</td>
</tr>
<tr>
<td>DBA_ADVISOR_DIR_TASK_INST</td>
</tr>
<tr>
<td>DBA_ADVISOR_EXECUTION_TYPES</td>
</tr>
<tr>
<td>DBA_ADVISOR_PARAMETERS</td>
</tr>
<tr>
<td>DBA_ADVISOR_PARAMETERS_PROJ</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_COLVOL</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_REC_SUM</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_TABLES</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_TABVOL</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_WK_MAP</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_WK_STMTS</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLA_WK_SUM</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLPLANS</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLSTATS</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_COLVOL</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_JOURNAL</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_PARAMETERS</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_STMTS</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_SUM</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_TABLES</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_TABVOL</td>
</tr>
<tr>
<td>DBA_ADVISOR_SQLW_TEMPLATES</td>
</tr>
<tr>
<td>DBA_ADVISOR_TEMPLATES</td>
</tr>
<tr>
<td>DBA_ADVISOR_USAGE</td>
</tr>
</tbody>
</table>
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
✓ Why scale-up your use of the Advisor Framework?
✓ Advisor tables
❖ What is stored in the advisor tables?
❖ Querying the advisor framework
❖ Case Studies
  Advisor framework in use in the tuning process
❖ Catalog of SQL Tools
❖ Questions
What is stored in the advisor tables?

❖ Findings - description about what was found (e.g. SQL Tuning Task, ADDM, …)  
  impact expressed in DB Time

❖ Recommendations - type and percent benefit of a Finding

❖ Actions – tuning actions recommended for a finding.
  Each action is specified by the COMMAND* and ATTR1 - ATTR6 col's
  * COMMAND defines how the attribute columns will be used.

❖ Rationale - type and benefit of a recommendation

❖ Objects – e.g. advisor objects has the SQL id's for Tuning Advisor

❖ Housekeeping:
  ❖ Tasks – advisor tasks for SQL Tuning, Segment Tuning
  ❖ Log - current: status, progress, error messages, execution times
  ❖ Journal – detailed progress of SQL Access advisor execution

DB Time = 
  CPU Time + Wait Time (non-idle) 
  “common currency” for 
  performance analysis

AAS = 
  Sum(DB Time) / Elapsed Time
DBA_ADVISOR - “E-R Diagram”

- TASKS
- FINDINGS
- ACTIONS
- RECOMMENDATIONS
- RATIONALE
- OBJECTS
### DBA_ADVISOR_TASKS

--- use this to get ADDM Report from another schema

```sql
SELECT DBMS_ADVISOR.GET_TASK_REPORT(task_name => :tname ,
owner_name => 'SYS' ) FROM DUAL;
```

ADDM Report for Task 'ADDM:1221021457_1_13991'

---

**Summary of Findings**

<table>
<thead>
<tr>
<th>Description</th>
<th>Active Sessions</th>
<th>Percent of Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  CPU Usage</td>
<td>.65</td>
<td>30.77</td>
</tr>
<tr>
<td>2  Top SQL Statements</td>
<td>.57</td>
<td>26.94</td>
</tr>
<tr>
<td>3  Undersized SGA</td>
<td>.41</td>
<td>19.48</td>
</tr>
<tr>
<td>4  Top Segments by &quot;User I/O&quot; and &quot;Cluster&quot;</td>
<td>.26</td>
<td>12.54</td>
</tr>
<tr>
<td>5  Hard Parse Due to Parse Errors</td>
<td>.12</td>
<td>5.73</td>
</tr>
<tr>
<td>6  Soft Parse</td>
<td>.08</td>
<td>4.02</td>
</tr>
<tr>
<td>7  PL/SQL Execution</td>
<td>.08</td>
<td>3.8</td>
</tr>
<tr>
<td>8  Shared Pool Latches</td>
<td>.08</td>
<td>3.67</td>
</tr>
<tr>
<td>9  Hard Parse Due to Invalidations</td>
<td>.04</td>
<td>2.09</td>
</tr>
</tbody>
</table>

---

**Impact** expressed in relation to **DB Time**

=> Be aware of overall AAS & CPU count when interpreting the findings

**DB Time** = **CPU Time + Wait Time (non-idle)**

**AAS** = **Sum(DB Time) / Elapsed Time**
ADDM: Taxonomy of Findings

- **Hardware Resource Issues**
  - CPU (capacity, top-sql, …)
  - IOs (capacity, top-sql, top-objects, undersized memory cache)
  - Cluster Interconnect
  - Memory (OS paging)

- **Software Resource Issues**
  - Application locks
  - Internal contention (e.g. access to db buffers)
  - Database Configuration

- **Application Issues**
  - Connection management
  - Cursor management (parsing, fetching, …)

Source: Oracle 10g The Self-Managing Database
DBA_ADVISOR  -  “E-R Diagram”
select * from ( 
select round((ratio_to_report(max(impact)) over () *100)) as pct_impact_overall,
           finding_name, type, min(impact) min_impact,
           max(impact) max_impact, impact_type, count(*) 
from DBA_ADVISOR_FINDINGS where impact_type is not null
       group by impact_type, finding_name, type) 
where pct_impact_overall >=5 order by
   pct_impact_overall desc ;
DBA_ADVISOR_FINDINGS – example: biggest “PROBLEM”

❖ high level findings can quickly highlight possible problems

USPRD025: SGA memory

UKPRD011: IO Subsystem
DBA_ADVISOR  -  “E-R Diagram”

TASKS

FINDINGS

OBJECTS

RECOMMENDATIONS

ACTIONS

RATIONALE

DBA_ADVISOR

"E-R Diagram"
### DBA_ADVISOR_RECOMMENDATIONS

```sql
select round((ratio_to_report(max(Benefit)) over () *100)) as overall_benefit_pct,
        type,
        min(benefit) min_benefit,
        max(Benefit) max_benefit,
        count(*) cnt
from dba_Advisor_recommendations
where type is not null group by type order by 1 desc;
```

<table>
<thead>
<tr>
<th>OVERALL BENEFIT PCT</th>
<th>TYPE</th>
<th>MIN_BENEFIT</th>
<th>MAX_BENEFIT</th>
<th>CNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Segment Tuning</td>
<td>39,665,809</td>
<td>20,763,426,690</td>
<td>1,331</td>
</tr>
<tr>
<td>22</td>
<td>Application Analysis</td>
<td>7,808,855</td>
<td>13,586,850,940</td>
<td>1,197</td>
</tr>
<tr>
<td>20</td>
<td>Host Configuration</td>
<td>168,781,614</td>
<td>11,950,233,345</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>SQL Tuning</td>
<td>36,964,444</td>
<td>10,120,889,078</td>
<td>2,556</td>
</tr>
<tr>
<td>7</td>
<td>DB Configuration</td>
<td>5,000,000</td>
<td>4,049,000,000</td>
<td>934</td>
</tr>
<tr>
<td>0</td>
<td>PARALLEL EXECUTION</td>
<td>5,043</td>
<td>9,947</td>
<td>35</td>
</tr>
<tr>
<td>0</td>
<td>STATISTICS</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>SQL PROFILE</td>
<td>100</td>
<td>9,999</td>
<td>43</td>
</tr>
<tr>
<td>0</td>
<td>INDEX</td>
<td>5,544</td>
<td>9,999</td>
<td>370</td>
</tr>
</tbody>
</table>
DBA_ADVISOR - “E-R Diagram”

- TASKS
- FINDINGS
- ACTIONS
- RECOMMENDATIONS
- RATIONALE
- OBJECTS

Diagram showing relationships between tasks, findings, recommendations, actions, rationale, and objects.
### DBA_ADVISOR_ACTIONS

```sql
select command, message, count(*)
from dba_advisor_Actions
group by command, message;
```

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>MESSAGE</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG OBJECT</td>
<td>Investigate the cause for high &quot;SQL*Net more data from client&quot; waits in Module &quot;osh@us1salx0024&quot;.</td>
<td>411</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Investigate the cause for high &quot;SQL*Net more data from client&quot; waits with P1 (&quot;driver id&quot;)</td>
<td>13</td>
</tr>
<tr>
<td>RUN SQL TUNING ADVISOR</td>
<td>Run SQL Tuning Advisor on the INSERT statement with SQL_ID &quot;722cx16t2m2tr&quot;.</td>
<td>14</td>
</tr>
<tr>
<td>RUN SQL TUNING ADVISOR</td>
<td>Run SQL Tuning Advisor on the UPDATE statement with SQL_ID &quot;3k77xwmkcnbna7&quot;.</td>
<td>27</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Investigate application logic involving I/O on INDEX &quot;BI_INTL_MEAP_ANALYSE.ERRL_E&quot;</td>
<td>35</td>
</tr>
<tr>
<td>RUN SQL TUNING ADVISOR</td>
<td>Run SQL Tuning Advisor on the UPDATE statement with SQL_ID &quot;5nwqynpwm8w5a&quot;.</td>
<td>35</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Investigate application logic to eliminate parse errors.</td>
<td>20</td>
</tr>
<tr>
<td>RUN SQL TUNING ADVISOR</td>
<td>Run SQL Tuning Advisor on the DELETE statement with SQL_ID &quot;3spmhfwkxx347&quot;.</td>
<td>1</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Investigate the SELECT statement with SQL_ID &quot;7rvddkacbuts&quot; for possible performance</td>
<td>1</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Look at the &quot;Top SQL Statements&quot; finding for SQL statements consuming significant I/O</td>
<td>10</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Look at the &quot;Top SQL Statements&quot; finding for SQL statements consuming significant time</td>
<td>1</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>Investigate the SELECT statement with SQL_ID &quot;04s9b5z0nmssx&quot; for possible performance</td>
<td>4</td>
</tr>
<tr>
<td>RUN SQL TUNING ADVISOR</td>
<td>Run SQL Tuning Advisor on the SELECT statement with SQL_ID &quot;fuq8my89cfqab&quot;.</td>
<td>21</td>
</tr>
</tbody>
</table>
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
✓ Why scale-up your use of the Advisor Framework?
✓ Advisor tables
✓ What is stored in the advisor tables?
❖ Querying the advisor framework
❖ Case Studies
   Advisor framework in use
   in the tuning process

❖ Catalog of SQL Tools
❖ Questions
Querying the advisor framework

“Mr. SQL” likes to “talk SQL” to his data

- DBA_ADVISOR - Alter Parameter.sql
- DBA_ADVISOR - Top SQL from ADDM Reports.sql
- DBA_ADVISOR - Accept SQL Profiles.sql
- DBA_ADVISOR – Indexes.sql
- DBA_ADVISOR – GatherStats.sql
- DBA_ADVISOR - Segment Advisor.sql
- DBA_ADVISOR - SQL Access Advisor - Reporting.sql
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
✓ Why scale-up your use of the Advisor Framework?
✓ Advisor tables
✓ What is stored in the advisor tables?
✓ Querying the advisor framework
❖ Case Studies
  Advisor framework in use in the tuning process

❖ Catalog of SQL Tools
❖ Questions
Customized Tuning Process

1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor

0. Collate accurate problem description

Root Cause Analysis
Use Case – “Worthy” SQL Access Advisor Rec’s

❖ **DBA_ADVISOR** - SQL Access Advisor - Reporting.sql

❖ Check to be sure SAA provides “worthy” rec’s

❖ **EXAMPLE:**

- **SQL Access Advisor Example.sql**
- Output: **SQL Access Advisor Example - Resulting Script.sql**

```
select * from dba_advisor_log where task_name = 'SQLA_USPRD069_DAY2017_09_14';
```

```
select * from dba_advisor_journal where task_name = 'SQLA_USPRD069_DAY2017_09_14'
order by journal_entry_seq desc;
```

Check to be sure SAA provides “worthy” rec’s
1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor
Use Case: Tune Top Flagged SQL

❖ DBA_ADVISOR - Top SQL from ADDM Reports.sql
❖ Action STA using the top 5 SQL from hourly ADDM runs

❖ Example: UKPRD666

DBA_ADVISOR - ADDM Report Details - UKPRD666.xls
Context within Customized Tuning Process

1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor
Use Case: Accept Profiles *enmasse*

- `DBA_ADVISOR - Accept SQL Profiles.sql`
- SQL Tuning Advisor (STA) runs are persisted
- Automatically extract lots of SQL Profiles
- Use when there are a lot of STA runs

Example: UKPRD666

`DBA_ADVISOR - Accept SQL Profiles - UKPRD666.xls`
Context within Customized Tuning Process

1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor
Use Case: Evaluate Index Recommendations

- **DBA_ADVISOR - Indexes.sql**
- **Aggregates index recommendations from multiple STA runs**
- **Provides**
  - Details on *benefit* and applicable SQL’s, ...
  - Create Index Commands, ...
- **Example: USPRD008**

<table>
<thead>
<tr>
<th>INDEX_TAB_COLS</th>
<th>MIN BENEFIT</th>
<th>MAX BENEFIT</th>
<th>CNT</th>
<th>SQL_LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD_FLUX8.FLUX_ACTION_RUN (&quot;ACTION_TYPE&quot;,&quot;ACTION_NAME&quot;)</td>
<td>53</td>
<td>90</td>
<td>5</td>
<td>1h2ukn4qv9qpa, 203ukc9zfqzv4, a1h14mu7wxbuz, bq5mubzd6azdt, gyj3m3g4fruwa</td>
</tr>
<tr>
<td>RD_FLUX8.FLUX_ACTION_RUN (&quot;ACTION_NAME&quot;,&quot;ACTION_NAME&quot;)</td>
<td>98</td>
<td>98</td>
<td>1</td>
<td>bq5mubzd6azdt</td>
</tr>
<tr>
<td>RD_FLUX8.FLUX_ACTION_RUN (&quot;ACTION_NAME&quot;)</td>
<td>77</td>
<td>77</td>
<td>1</td>
<td>bq5mubzd6azdt</td>
</tr>
<tr>
<td>RD_FLUX8.FLUX_AUDIT_TRAIL (CASE INSTR(&quot;NAMESPACE&quot;,&quot;$&quot;,1) WHEN 0 THEN &quot;NAMESPACE&quot; ELSE SUBSTR(&quot;NAMESPACE&quot;,CASE INSTR(&quot;NAMESPACE&quot;,&quot;$&quot;,-1,1) WHEN 0 THEN LENGTH(&quot;NAMESPACE&quot;)+1 ELSE INSTR(&quot;NAMESPACE&quot;,&quot;$&quot;,-1,1)+1 END,20)),&quot;EVENT&quot;,&quot;CREATION&quot;)</td>
<td>90</td>
<td>90</td>
<td>1</td>
<td>a1h14mu7wxbuz</td>
</tr>
<tr>
<td>RD_FLUX8.FLUX_AUDIT_TRAIL (&quot;EVENT&quot;,TO_NUMBER(SUBSTR(&quot;NAMESPACE&quot;,CASE INSTR(&quot;NAMESPACE&quot;,&quot;$&quot;,-1,1) WHEN 0 THEN LENGTH(&quot;NAMESPACE&quot;)+1 ELSE INSTR(&quot;NAMESPACE&quot;,&quot;$&quot;,-1,1)+1 END,20)),&quot;CREATION&quot;)</td>
<td>53</td>
<td>90</td>
<td>3</td>
<td>1h2ukn4qv9qpa, 203ukc9zfqzv4, a1h14mu7wxbuz, bq5mubzd6azdt</td>
</tr>
<tr>
<td>RD_FLUX8.FLUX_AUDIT_TRAIL (&quot;EVENT&quot;,&quot;CREATION&quot;,&quot;ACTION_NAME&quot;)</td>
<td>77</td>
<td>77</td>
<td>1</td>
<td>bq5mubzd6azdt</td>
</tr>
<tr>
<td>RD_FLUX8.FLUX_AUDIT_TRAIL (&quot;EVENT&quot;,&quot;CREATE&quot;,&quot;NAME&quot;)</td>
<td>67</td>
<td>78</td>
<td>2</td>
<td>203ukc9zfqzv4, gyj3m3g4fruwa, bq5mubzd6azdt, bq5mubzd6azdt</td>
</tr>
</tbody>
</table>
Use Case: Evaluate Parameter Recommendations

- **DBA_ADVISOR** - Alter Parameter.sql
- Leverages the fact that ADDM runs every hour
- **Example: USPRD025**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>ACTION</th>
<th>COUNT</th>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cursor_sharing</td>
<td>27</td>
<td></td>
<td>Alternatively, you may set the parameter &quot;cursor_sharing&quot; to &quot;force&quot;.</td>
</tr>
<tr>
<td>pga_aggregate_target</td>
<td>7</td>
<td></td>
<td>Increase the size of the PGA by setting the value of parameter &quot;pga_aggregate_target&quot; to 17416 M.</td>
</tr>
<tr>
<td>pga_aggregate_target</td>
<td>1</td>
<td></td>
<td>Increase the size of the PGA by setting the value of parameter &quot;pga_aggregate_target&quot; to 15481 M.</td>
</tr>
<tr>
<td>session_cached_cursors</td>
<td>137</td>
<td></td>
<td>Increase the size of the PGA by setting the value of parameter &quot;sessionCachedCursors&quot; to 11611 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>3</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 32768 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>3</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 30720 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>2</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 28672 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>10</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 26624 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>59</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 24576 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>63</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 22528 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>104</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 20480 M.</td>
</tr>
<tr>
<td>sga_target</td>
<td>299</td>
<td></td>
<td>Increase the size of the SGA by setting the parameter &quot;sga_target&quot; to 18432 M.</td>
</tr>
</tbody>
</table>
Context within Customized Tuning Process

1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor
Use Case: Segment Tuning *enmasse*

- **DBA_ADVISOR - Segment Advisor.sql**

- When RCA suggests need for Segment Tuning

- Extract Advice for specific or many runs

- Example: UKPRD011 with sysstat observation: high: “*table fetch continue row*”
  msg %chained rows%

<DBA_ADVISOR - Segment Advisor - UKPRD011.xls>
Context within Customized Tuning Process

1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor
Use cases not instrumented – remaining work

❖ Restructure SQL Finding:
  ❖ An expensive Cartesian product operation was found …
    ❖ add a join condition
  ❖ Predicate contains an expression on an indexed column
    ❖ Rewrite or create a function-based index on the expression.

❖ Alternative Plan Finding
  ❖ See “NUM_PLANS” column from
    DBA_ADVISOR - Top SQL from ADDM Reports.sql

❖ Integrating findings:
  ❖ Profiles
  ❖ Indexes
  ❖ Missing stats
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
✓ Why scale-up your use of the Advisor Framework?
✓ Advisor tables
✓ What is stored in the advisor tables?
✓ Querying the advisor framework
✓ Case Studies
  Advisor framework in use in the tuning process

❖ Catalog of SQL Tools
❖ Questions
Catalog of SQL Tools

❖ DBA_ADVISOR - Alter Parameter.sql
❖ DBA_ADVISOR - Top SQL from ADDM Reports.sql
❖ DBA_ADVISOR - Accept SQL Profiles.sql
❖ DBA_ADVISOR – Indexes.sql
❖ DBA_ADVISOR – GatherStats.sql
❖ DBA_ADVISOR - Segment Advisor.sql
❖ DBA_ADVISOR - SQL Access Advisor - Reporting.sql
❖ SQL_Details_rpt.sql
❖ SQLTUNE.sql
Scaling-Up Your Use of the Advisor Framework

✓ What is the Advisor Framework?
✓ Why scale-up your use of the Advisor Framework?
✓ Advisor tables
✓ What is stored in the advisor tables?
✓ Querying the advisor framework
✓ Case Studies
✓ Advisor framework in use
✓ in the tuning process

✓ Catalog of SQL Tools
❖ Questions
Scale-Up Your Use of the Advisor Framework

Roger Cornejo
Principal Oracle Performance Consultant
@OracleDBTuning

Dynamic Oracle Performance Analytics
Using Normalized Metrics to Improve Database Speed
— Roger Cornejo
Scale-Up Your Use of the Advisor Framework

Roger Cornejo
Principal Oracle Performance Consultant
@OracleDBTuning

Oracle Open World - San Francisco – September 16 – 19 2019

Appendix
### Tools and Scripts:

<table>
<thead>
<tr>
<th>Tool</th>
<th># of scripts</th>
<th>Lines of Code</th>
<th>Impact / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse Engineering Advisors</td>
<td>6</td>
<td>~1400</td>
<td>Faster ability to action tuning findings</td>
</tr>
<tr>
<td>Customized Tuning</td>
<td>10</td>
<td>~4000</td>
<td>Fills Standard Tools Gap</td>
</tr>
<tr>
<td>Beyond Tuning</td>
<td>6</td>
<td>~1000</td>
<td>Troubleshooting informed by AWR metrics</td>
</tr>
<tr>
<td>Health Checks</td>
<td>12</td>
<td>~5600</td>
<td>Various Metrics based health checks</td>
</tr>
<tr>
<td>Advanced Analytics</td>
<td>2</td>
<td>~2400</td>
<td>Laser focus on &gt; 11,000 Metrics</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>~14,400</td>
<td></td>
</tr>
</tbody>
</table>

My toolkit consists of dozens of SQL scripts; > 14,000 lines of code. Custom developed tools designed to expose root cause to specific perf. issues.
80% of DB performance issues due to application, 20% due to technology

<table>
<thead>
<tr>
<th>Application (~80%)</th>
<th>Technology (~20%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data volume</td>
<td>Data volume</td>
</tr>
<tr>
<td>SQL Tuning</td>
<td>SQL Tuning</td>
</tr>
<tr>
<td>Number of users</td>
<td>Number of users</td>
</tr>
<tr>
<td>Workload</td>
<td>Workload</td>
</tr>
<tr>
<td>DB Optimizer</td>
<td>DB Optimizer</td>
</tr>
<tr>
<td>Configuration</td>
<td>Configuration</td>
</tr>
<tr>
<td>Subsystems</td>
<td>Subsystems</td>
</tr>
<tr>
<td>I/O, CPU, Net.</td>
<td>I/O, CPU, Net.</td>
</tr>
</tbody>
</table>

Many metrics associated with application level concerns:
- Data changes
- Session level and SQL Stats
- Number of users
- Transaction rates
- Workloads

Many metrics associated with Technology:
- Technology stack
  - IO rates and throughput
  - CPU and Memory Usage
  - Network
- Oracle Subsystems
  - REDO / UNDO / TEMP

Categories of areas where problems come from

Performance issues are different every time & require specialized tools to analyze
Custom developed tools designed to expose root cause to specific perf. issues

Tool types I have developed identify specific categories of problems

<table>
<thead>
<tr>
<th>Tool</th>
<th>Application</th>
<th>Technology / Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Volume</td>
<td>SQL Tuning</td>
</tr>
<tr>
<td>Reverse Engineering Advisor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customized Tuning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beyond Tuning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced Analytics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Health Checks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RCA / Solutions to specific issues
Context within Customized Tuning Process

1. Identify Slow Session(s)
2. Examine Slow SQL Stats
3. Analyze SQL Details
4. SQL Tuning Advisor
5. Confirm RCA / Solution(s)
6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)
7. Implement / Monitor
Context within Customized Tuning Process

0. Collate accurate problem description

1. Identify Slow Session(s)

2. Examine Slow SQL Stats

3. Analyze SQL Details

4. SQL Tuning Advisor

5. Confirm RCA / Solution(s)

6. Convey:
   - Observations
   - Root Cause(s)
   - Solution(s)

7. Implement / Monitor

Root Cause Analysis