Oracle InMemory Application for reduced latency in maintenance processes

Oracle Open World, San Francisco, 2017
Agenda

1. Who are we
2. Who is my employer
3. What’s the customers business
4. Fundamentals
5. Problem
6. Solution
Who are we

- Thorsten Pensky
- Age 48
- First contact with Oracle in 1993 (Oracle V6, SQL*Forms 3)
  - Development of warehouse management systems and the connection to SPCs for control of flow of materials
  - Guidance systems for pallet transporter
  - Voice controlled picking of goods
- First DBA course for Oracle 9i
- Actually responsible for databases at Lufthansa
  - Provisioning/Customizing/Maintenance/Optimizing
  - Invoice
  - Human Resource Planing
- Hobbies: Music, travelling & scuba diving
Who are we

- Sonja Meyer
- Age formerly known ;-) 
- First contact with Oracle in 2009 (Oracle 10g, PL/SQL)
  - 2nd level support for Siebel and self-developed applications on ORACLE database
  - Development of warehouse management system and forecast
  - Business intelligence
- Started at ORACLE in 2009 in consulting expert service responsible for MAA and performance tuning
- Actually working for PreSales in Germany as an IT cloud architect Technology responsible for large accounts in cloud architecture, and project lead POCs (EXADATA and Database IN-Memory) and for crazy ideas
- Hobbies: Music, travelling, reading, beachvolleyball & running
Who is my employer

- Lufthansa Industry Solutions
  - Subsidiary of Lufthansa Group
  - IT-provider
- > 1300 employees
- 10 sites
- > 200 customers
  - MRO
  - Transport & logistics
  - Manufacturing
  - Automotive
  - Energy
  - Media
  - Healthcare

Oracle InMemory Application for reduced latency in maintenance processes
Who is my employer

- My business segment AB/M-E
- 19 colleagues
- Maintenance of
  - Oracle/MS SQL databases
  - Webmethods/servicebus
  - Openshift
- Administration/tuning/consulting
- Actually
  - 290 Oracle databases
  - 16 MS SQL databases
- New customer asking for service of 160 more Oracle databases

10/11/2017
Oracle InMemory Application for reduced latency in maintenance processes
What’s the customers business

- Primarily
  - Maintenance of 2000 aircrafts at more than 60 airports worldwide
  - Overhaul/repair of all components of 30 types of aircrafts
    - From coffee maker
    - To complete engines (40 types)
    - And landing gears
    - Operated 24/7/365

- Secondary
  - Support for private VIP, business and government aircrafts
    - Cabin upgrades
    - Maintenance
    - Painting
What’s the customers business – the special segment

- Every 6–10 years a D-check is performed
- Takes 4–6 weeks and 30,000–50,000 man-hours of labor
- Airplane is nearly completely disassembled
- Every part is examined if it needs repair
- Repair is done in own workshops or by subcontractors

- Tracking is very important
  - Which component is send for repair
  - Where is it actually (transport, repair, warehouse)
  - Will it be usable again and when
  - If not, which component can be used instead
What’s the customers business – the special segment
What’s the customers business – the special segment

- Tracking DB
  - Timestamps
  - Staging
  - Report Data (InMemory)

- Business-reports
- Aggregated reports

- Interfaces
Fundamentals

RAC OneNode Cluster

- RedHat Enterprise 6 x86-64
- 110 Oracle Databases (11.2 & 12.1)
  - 38 Production
  - 72 Test
- Oracle Grid Infrastructure
- ASM / ACFS
- RAC One-Node
- Partitioning Option
- In-Memory
**Fundamentals**

- Oracle 11.2.0.4
- SGA 24 GB
- PGA 5 GB
- AMM
- Total Size 1.2 TB
## Customer Status Report

<table>
<thead>
<tr>
<th>Customer</th>
<th>Supplier</th>
<th>PO No</th>
<th>PO Item</th>
<th>PO P/N</th>
<th>PO S/N</th>
<th>PO Qty</th>
<th>Status</th>
<th>PO Date</th>
<th>Rcvd Qty</th>
<th>Rcvd P/N</th>
<th>Rcvd S/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer A</td>
<td>Supplier C</td>
<td>203599</td>
<td>1</td>
<td>1703M30</td>
<td>DF5739</td>
<td>1</td>
<td>In progress</td>
<td>15 Aug 17</td>
<td>1</td>
<td>1703M30</td>
<td>DF5739</td>
</tr>
<tr>
<td>Customer A</td>
<td>Supplier C</td>
<td>203668</td>
<td>1</td>
<td>500730-1</td>
<td>1</td>
<td>shipping</td>
<td>12 Aug 17</td>
<td>1</td>
<td>500730-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer C</td>
<td>Supplier E</td>
<td>203212</td>
<td>1</td>
<td>31DA10</td>
<td>1</td>
<td>ready to ship</td>
<td>10 Aug 17</td>
<td>1</td>
<td>31DA10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer F</td>
<td>Supplier D</td>
<td>203781</td>
<td>1</td>
<td>9092M4</td>
<td>LJA443</td>
<td>2</td>
<td>In progress</td>
<td>18 Aug 17</td>
<td>2</td>
<td>9092M4</td>
<td>LJA443</td>
</tr>
<tr>
<td>Customer F</td>
<td>Supplier D</td>
<td>203781</td>
<td>2</td>
<td>1475M3</td>
<td>MDDA03</td>
<td>1</td>
<td>In progress</td>
<td>18 Aug 17</td>
<td>1</td>
<td>1475M3</td>
<td>MD23HC</td>
</tr>
<tr>
<td>Customer H</td>
<td>Supplier R</td>
<td>202355</td>
<td>1</td>
<td>6A76L4</td>
<td>1</td>
<td>closed</td>
<td>02 Aug 17</td>
<td>1</td>
<td>6A76L4</td>
<td>DF5739</td>
<td></td>
</tr>
<tr>
<td>Customer K</td>
<td>Supplier A</td>
<td>204002</td>
<td>1</td>
<td>508P27</td>
<td>AL2344</td>
<td>1</td>
<td>Repair</td>
<td>23 Aug 17</td>
<td>1</td>
<td>508P27</td>
<td>AL2344</td>
</tr>
<tr>
<td>Customer M</td>
<td>Supplier G</td>
<td>204137</td>
<td>1</td>
<td>45-008</td>
<td>LY222</td>
<td>1</td>
<td>Open</td>
<td>30 Aug 17</td>
<td>1</td>
<td>45-008</td>
<td>LY222</td>
</tr>
</tbody>
</table>
Underlying view selects
31 tables
5 views
3 materialized views

> 2000 lines of code
Problem

Runtime of views used in daily routine

<table>
<thead>
<tr>
<th>View-Content</th>
<th>Runtime 11g (Minutes)*</th>
<th>Runtime 12c (Minutes)**</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long takes repair of the engine</td>
<td>3:47</td>
<td>4:46</td>
</tr>
<tr>
<td>How long takes the transport of the engine</td>
<td>2:45</td>
<td>0:27</td>
</tr>
<tr>
<td>Timereport of component maintenance</td>
<td>0:06</td>
<td>0:10</td>
</tr>
<tr>
<td>Complete report of customer components</td>
<td>4:46</td>
<td>3:00</td>
</tr>
<tr>
<td>Customer sends back spare component</td>
<td>8:27</td>
<td>6:37</td>
</tr>
<tr>
<td>Average processing time over last 3 years</td>
<td>3:50</td>
<td>- ***</td>
</tr>
<tr>
<td>Performancereport</td>
<td>10:00</td>
<td>- ***</td>
</tr>
</tbody>
</table>

* 11.2.0.4  ** 12.1.0.2.2  *** lost in space

Tests done on a single node without other databases to avoid any interference
Solution

- Next step was a proof of concept: 31st March 2015–2nd April 2015
  - apply actual patch set #20329440
    (DATABASE PATCH FOR ENGINEERED SYSTEMS AND DB IN-MEMORY 12.1.0.2.5)
- establish huge pages on server
- enable inmemory in database
  - SGA_TARGET = 150G (formerly 24G)
  - INMEMORY_SIZE = 120G
  - MEMORY_TARGET = 0 (no AMM due to use of HugePages)
  - INMEMORY_MAX_POPULATE_SERVERS = 12 (#cores)
- lookup tables used in the reports (no IM advisor because we knew what we did)
- configured tables for inmemory

- Check changes in behaviour
### Solution

Examples of tables in memory (total 76 tables)

<table>
<thead>
<tr>
<th>SEGMENT_NAME</th>
<th>TYPE</th>
<th>IM SIZE</th>
<th>BYTES</th>
<th>COMP (%)</th>
<th>FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR_LTL_REQUEST_TIMETABLE_FINAL</td>
<td>TABLE</td>
<td>830.930.944</td>
<td>1.940.291.584</td>
<td>57,17</td>
<td>2,34</td>
</tr>
<tr>
<td>AR_CUSTOMER_PROCESS</td>
<td>TABLE</td>
<td>960.057.750</td>
<td>3.685.621.760</td>
<td>73,95</td>
<td>3,84</td>
</tr>
<tr>
<td>BR_CLARIF_RECV</td>
<td>TABLE</td>
<td>339.345.408</td>
<td>786.808.832</td>
<td>56,87</td>
<td>2,32</td>
</tr>
<tr>
<td>BR_BUSINESS_PROC_CLARIF_GROUP</td>
<td>TABLE</td>
<td>1.179.648</td>
<td>1.695.744</td>
<td>30,43</td>
<td>1,44</td>
</tr>
<tr>
<td>BR_ORDER_STATUS</td>
<td>TABLE</td>
<td>2.983.526.400</td>
<td>5.629.870.080</td>
<td>47,01</td>
<td>1,89</td>
</tr>
<tr>
<td>AR_ORD_PPS_MM</td>
<td>TABLE</td>
<td>181.141.504</td>
<td>226.967.552</td>
<td>20,19</td>
<td>1,25</td>
</tr>
<tr>
<td>AR_TCS_IN</td>
<td>TABLE</td>
<td>316.604.416</td>
<td>402.440.192</td>
<td>21,33</td>
<td>1,27</td>
</tr>
<tr>
<td>BR_LOCATION_BOOK</td>
<td>TABLE</td>
<td>1.128.595.456</td>
<td>1.540.235.264</td>
<td>26,73</td>
<td>1,36</td>
</tr>
</tbody>
</table>

MEMCOMPRESS FOR QUERY LOW PRIORITY HIGH

Factor varies from 1.08 up to 3.84
## Solution

**Attention**

<table>
<thead>
<tr>
<th>SEGMENT_NAME</th>
<th>TYPE</th>
<th>IM SIZE</th>
<th>BYTES</th>
<th>COMP (%)</th>
<th>FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR_ORDER_ATTACH</td>
<td>TABLE</td>
<td>1.179.648</td>
<td>106.496</td>
<td>-1107,69</td>
<td>-11,08</td>
</tr>
<tr>
<td>BR_ORDER_CLARIF_GROUP</td>
<td>TABLE</td>
<td>1.179.648</td>
<td>671.744</td>
<td>-175,61</td>
<td>-1,76</td>
</tr>
</tbody>
</table>

**Explanation:**

InMemory has a little bit of overhead for every table

--> Small tables don’t need to be inmemory
Solution

But it can also happen to larger tables

<table>
<thead>
<tr>
<th>SEGMENT_NAME</th>
<th>TYPE</th>
<th>IM_SIZE</th>
<th>BYTES</th>
<th>COMP (%)</th>
<th>FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR_BUSINESS_PROC_CLARIF</td>
<td>TABLE</td>
<td>182.190.080</td>
<td>176.832.512</td>
<td>103,03</td>
<td>-1,03</td>
</tr>
<tr>
<td>BR_MATERIAL&gt;Returns</td>
<td>TABLE</td>
<td>72.548.352</td>
<td>71.737.344</td>
<td>101,13</td>
<td>-1,01</td>
</tr>
</tbody>
</table>

Explanation:

No real explanation. After installation of two more bundle patches the issue vanished.

-> Keep your database up-to-date
## Solution

Runtime of views used in daily routine in minutes (12c only)

<table>
<thead>
<tr>
<th>View-Content</th>
<th>w/o IM</th>
<th>Serial with Indexes</th>
<th>Serial w/o Indexes</th>
<th>PQ4 with Indexes</th>
<th>PQ4 w/o Indexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long takes repair of an engine</td>
<td>4:46</td>
<td>00:54</td>
<td>00:57</td>
<td>00:04</td>
<td>00:38</td>
</tr>
<tr>
<td>How long takes the transport of an engine</td>
<td>0:27</td>
<td>00:18</td>
<td>00:18</td>
<td>08:17</td>
<td>09:12</td>
</tr>
<tr>
<td>Timereport of component maintenance</td>
<td>0:10</td>
<td>00:05</td>
<td>00:27</td>
<td>00:15</td>
<td>00:13</td>
</tr>
<tr>
<td>Complete report of customer components</td>
<td>3:00</td>
<td>04:30</td>
<td>- **</td>
<td>00:54</td>
<td>- **</td>
</tr>
<tr>
<td>Customer sends back spare component</td>
<td>6:37</td>
<td>00:19</td>
<td>00:06</td>
<td>00:20</td>
<td>00:03</td>
</tr>
<tr>
<td>Average processing time over last 3 years</td>
<td>- *</td>
<td>00:44</td>
<td>00:42</td>
<td>**00:12</td>
<td>00:12</td>
</tr>
<tr>
<td>Performance report</td>
<td>- *</td>
<td>00:06</td>
<td>00:21</td>
<td>00:15</td>
<td>00:03</td>
</tr>
</tbody>
</table>

* lost in space   ** cancelled due to runtime - see next slide

Tests done on a single node without other databases to avoid any interference
Solution

Runtime problem of view

Exceptional use of temp tablespace with “hash group by”

Fixed with Bundle Patch 6
Solution

Runtime of views used in daily routine in minutes (12c only) (PARALLEL_DEGREE_POLICY=AUTO)

<table>
<thead>
<tr>
<th>View-Content</th>
<th>w/o IM</th>
<th>Auto DOP with Index</th>
<th>Auto DOP w/o Index</th>
<th>Auto DOP with Index *</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long takes repair of the engine</td>
<td>4:46</td>
<td>00:12 (PQ8)</td>
<td>00:26 (PQ8)</td>
<td>00:02 (PQ8)</td>
</tr>
<tr>
<td>How long takes the transport of the engine</td>
<td>0:27</td>
<td>00:53 (S)</td>
<td>00:55 (S)</td>
<td>00:54 (S)</td>
</tr>
<tr>
<td>Timereport of component maintenance</td>
<td>0:10</td>
<td>00:08 (S)</td>
<td>00:27 (S)</td>
<td>00:04 (S)</td>
</tr>
<tr>
<td>Complete report of customer components</td>
<td>3:00</td>
<td>- **</td>
<td>- **</td>
<td>- **</td>
</tr>
<tr>
<td>Customer sends back spare component</td>
<td>6:37</td>
<td>00:30 (S)</td>
<td>00:06 (S)</td>
<td>00:12 (S)</td>
</tr>
<tr>
<td>Average processing time over last 3 years</td>
<td>-</td>
<td>00:42 (S)</td>
<td>00:41 (S)</td>
<td>00:46 (S)</td>
</tr>
<tr>
<td>Performancereport</td>
<td>-</td>
<td>00:25 (PQ8)</td>
<td><strong>00:01 (PQ8)</strong></td>
<td>00:19 (PQ8)</td>
</tr>
</tbody>
</table>

* new statistics for test reasons  ** remember previous slide, it’s the same trouble

Tests done on a single node without other databases to avoid any interference
Solution

Fastest runtimes of views in minutes

<table>
<thead>
<tr>
<th>View-Content</th>
<th>11g</th>
<th>Fastest time</th>
<th>Factor</th>
<th>Final (no PQ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long takes repair of the engine</td>
<td>3:47</td>
<td>0:02 Auto DOP with Ind (PQ8)</td>
<td>113</td>
<td>4.2</td>
</tr>
<tr>
<td>How long takes the transport of the engine</td>
<td>2:45</td>
<td>0:18 Serial with or w/o Indexes</td>
<td>9.2</td>
<td>9.2</td>
</tr>
<tr>
<td>Timereport of component maintenance</td>
<td>0:06</td>
<td>0:04 Auto DOP with Ind (PQ8)</td>
<td>1.5</td>
<td>1.2</td>
</tr>
<tr>
<td>Complete report of customer components</td>
<td>4:46</td>
<td>0:54 With Indexes (PQ4)**</td>
<td>5.3</td>
<td>1.1</td>
</tr>
<tr>
<td>Customer sends back spare component</td>
<td>8:27</td>
<td>0:03 Without Indexes (PQ4)</td>
<td>169</td>
<td>26.7</td>
</tr>
<tr>
<td>Average processing time over last 3 years</td>
<td>3:50</td>
<td>0:12 Serial with or w/o Indexes</td>
<td>19</td>
<td>5.5</td>
</tr>
<tr>
<td>Performancereport</td>
<td>10:00</td>
<td>0:01 Auto DOP w/o Ind (PQ8)</td>
<td>600</td>
<td>100</td>
</tr>
</tbody>
</table>

** on stardate -306752.4 we lost contact

Tests done on a single node without other databases to avoid any interference
Solution

Surprise looking into v$im_segments !!!

<table>
<thead>
<tr>
<th>SEGMENT_NAME</th>
<th>TYPE</th>
<th>IM_SIZE</th>
<th>BYTES</th>
<th>BYTES NOT POPULATED</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR_LTL_REQUEST_TIMETABLE_FINAL</td>
<td>TABLE</td>
<td>830.930.944</td>
<td>1.940.291.584</td>
<td>0</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>AR_CUSTOMER_PROCESS</td>
<td>TABLE</td>
<td>108.920.832</td>
<td>3.685.621.760</td>
<td>2.418.122.752</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>BR_LOCATION_BOOK</td>
<td>TABLE</td>
<td>1.128.595.456</td>
<td>1.540.235.264</td>
<td>0</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>BR_SPLITTING</td>
<td>TABLE</td>
<td>207.028.224</td>
<td>485.998.592</td>
<td>225.509.376</td>
<td>COMPLETED</td>
</tr>
</tbody>
</table>

Explanation:

The status shown in the view is partially wrong. This was found to be a bug, which was resolved in PBP 161018.

--> Keep your database up-to-date
Solution

- Conclusion
  
  - **Fact: In-memory solved our problems**

    - Due to shared infrastructure we started with no parallelism
    
    - Runtimes could be reduced by an average of factor 8 to 10
    - I/O was reduced which is a benefit for our shared servers
    - Diskspace was freed by deleting indexes

    - Today even dynamic SQL is allowed to users (self-service BI tools)
Solution

- Where are we today
  - new storage system is used (IBM XIV) (avg. 900MB/sec)
  - storage is connected to servers via 16 GB adapters
  - RAM on every node increased to 768 GB

- starting with customers production system we put all report tables inmemory (76 tables)
  - beginning with an inmemory_target of 80 GB today we use 110 GB
    due to increased amount of data and new tables (act. 88)

- more compression (QUERY HIGH) due to increased amount of data -> although no speed impact

- two more databases starting to use inmemory; three more databases to come
Any questions left?
Thanks for your attention