RDP302

Understanding SAP HANA Database Performance

Lars Breddemann, SCE EMEA
Richard Bremer, SCE EMEA
Charu Agarwal, SCE APJ
TechEd 2013
Disclaimer

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.
You and SAP HANA

- SAP HANA is, amongst other things, first and foremost a database management system.

- Or short: **it’s the DB system for your application.**

- As that it takes care about storing and retrieving your data safely and efficiently.

- That means:
  You don’t have to care about that – SAP HANA offers this as a service to you, the application developer.

- Focus on your application code and let SAP HANA figure out how to deal with your data.
Performance dilemma

But how to care about performance if the DB is handling all the data handling for me?

• Use funny SQL coding tricks? \( (\text{WHERE } 1=1 \text{ AND } 1 \text{ IS NOT NULL...}) \)

• Put optimizer hints into the SQL? \( (\text{/*+ INDEXACCESS ... */}) \)

• Set DB parameters? \( (\text{force\_fast\_query\_processing = true}) \)

STOP! This means YOU.

• This kills your productivity
• This makes your system hard to maintain and operate
• This prevents your SQL code to benefit from the improvements in SAP HANA revisions
Performance dilemma – EXIT
(because system performance is still your responsibility)

• Don’t treat SAP HANA as a black box!

• Understand how it works and use SAP HANA based on that!

• Because: “Garbage in, garbage out” is still true in a client/server situation.

• So, let’s see what SAP HANA does with your queries and how you can find out about it…
The rise and fall of your SQL commands
or the lifetime of a session in SAP HANA
Lifetime of a SAP HANA session

- Applications create sessions in the SAP HANA server
- Sessions could be seen as a conversation between client and server
- They provide context to the execution of SQL commands
- Sessions are facilitated over one or more physical connections (network connections)

- **SYSTEM** variables are set at logon/automatically and cannot be changed during the session e.g. **session language** is set via **LOCALÉ** (ISO lang code)/**LOCALÉ_SAP** (SAP internal language encoding)
- **USER** variables can be freely defined (up to 50 per session) and accessed from outside the session (**m_session_context** view, SAP HANA studio)
- **APPLICATION** variables used by convention to allow easier system monitoring
  - **APPLICATIONUSER** should show the „real“ user (SAP NetWeaver puts in ABAP user)
  - **APPLICATIONSOURCE** shows source location of database call (automatically in SAP NetWeaver)
  - **APPLICATIONVERSION** should provide versioning information of the DB client
### Session context variables

```sql
SELECT * FROM m_session_context WHERE connection_id = current_connection
```

<table>
<thead>
<tr>
<th>HOST</th>
<th>PORT</th>
<th>CONNECTION_ID</th>
<th>KEY</th>
<th>VALUE</th>
<th>SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>PROTOCOL_VERSION 4.1 (1, 4)</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>2</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>LOCALE de</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>3</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>LOCALE_SAP D</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>4</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>APPLICATIONVERSI... 1.0.64.201309030448</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>5</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>APPLICATION HDBStudio</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>6</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>APPLICATIONSOURCE... run(SQLExecuteFormEditor.java:847);</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>7</td>
<td>Id9506</td>
<td>30.003</td>
<td>220.329</td>
<td>APPLICATIONUSER 1028297</td>
<td>SYSTEM</td>
</tr>
</tbody>
</table>
Session context variables – used for monitoring

- **APPLICATION**-variables provide additional meta data for monitoring and answers questions like:
  - Which ABAP module runs a statement?
  - Which SAP HANA Studio versions are used?

- Which user is running the session?
- DB user ≠ OS user
- DB user ≠ SAP NetWeaver user

- **APPLICATIONUSER** provides this information
Lifetime of SAP HANA session

- Applications create sessions in the SAP HANA server
- Sessions could be seen as a conversation between client and server
- They provide context, a semantic frame, to the execution of SQL commands
- Sessions are facilitated over one or more connections

- Connections are physical network connections
- Can be cancelled/killed by administrator
- Enable multi-node access for a single DB session
- Can be automatically cancelled based on timeout mechanisms

indexserver.ini
[session]

idle_connection_timeout_application_statisticsserver
idle_connection_timeout_application_hdbstudio
idle_connection_timeout
connection_history_lifetime
### Connection information – SQL

```sql
SELECT connection_id, transaction_id, client_host, client_pid, connection_status, user_name, start_time, 
    round((seconds_between (start_time, current_timestamp)/60/60, 2) as connection_time_h, own 
FROM connections 
WHERE port = 30000 AND connection_type not like '%History%' 
ORDER BY connection_status desc, start_time desc
```

<table>
<thead>
<tr>
<th>CONNECTION_ID</th>
<th>TRANSACTION_ID</th>
<th>CLIENT_HOST</th>
<th>CLIENT_PID</th>
<th>CONNECTION_STATUS</th>
<th>USER_NAME</th>
<th>START_TIME</th>
<th>CONNECTION_TIME_H</th>
<th>OWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>219.237</td>
<td>251</td>
<td>VIERN02324022A.sap.corp</td>
<td>3.328</td>
<td>RUNNING</td>
<td>1028297</td>
<td>09.09.2013 14:14:15 910266</td>
<td>1.43</td>
</tr>
<tr>
<td>2</td>
<td>219.300</td>
<td>256</td>
<td>WDFN00287523A.sap.corp</td>
<td>16.376</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>08.09.2013 14:33:59 673523</td>
<td>1.1</td>
</tr>
<tr>
<td>7</td>
<td>219.221</td>
<td>241</td>
<td>WDFN00287523A.sap.corp</td>
<td>16.376</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 14:10:06 447062</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>219.204</td>
<td>254</td>
<td>WDFN00287523A.sap.corp</td>
<td>16.376</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 14:07:53 840567</td>
<td>1.54</td>
</tr>
<tr>
<td>12</td>
<td>219.197</td>
<td>257</td>
<td>WDFN00287523A.sap.corp</td>
<td>16.376</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 14:07:27 987721</td>
<td>1.54</td>
</tr>
<tr>
<td>14</td>
<td>219.062</td>
<td>249</td>
<td>WDFN30002408A.dhcp...</td>
<td>9.172</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 13:36:52 686686</td>
<td>2.03</td>
</tr>
<tr>
<td>15</td>
<td>219.060</td>
<td>248</td>
<td>WDFN30002408A.dhcp...</td>
<td>9.172</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 13:36:52 433505</td>
<td>2.05</td>
</tr>
<tr>
<td>16</td>
<td>219.030</td>
<td>246</td>
<td>WDFN30002408A.dhcp...</td>
<td>9.172</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 13:36:44 51392</td>
<td>2.06</td>
</tr>
<tr>
<td>17</td>
<td>218.988</td>
<td>247</td>
<td>WDFN30002408A.dhcp...</td>
<td>9.172</td>
<td>IDLE</td>
<td>SYSTEM</td>
<td>09.09.2013 13:26:48 799067</td>
<td>2.22</td>
</tr>
</tbody>
</table>
### Connection information – DBACockpit

<table>
<thead>
<tr>
<th>Host</th>
<th>Port Number</th>
<th>Connection ID</th>
<th>Transaction ID</th>
<th>Start Time</th>
<th>Idle Time</th>
<th>Connection Status</th>
<th>Client Host</th>
<th>Client IP</th>
<th>Client FID</th>
<th>User Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.6</td>
<td>30.0.04</td>
<td>500.027</td>
<td>13</td>
<td>09.09.2013 17:08:15</td>
<td>0</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>3.536</td>
<td>SAPWRN</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.05</td>
<td>308.312</td>
<td>73</td>
<td>09.09.2013 18:00:41</td>
<td>90.079</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>9.136</td>
<td>SAPWRN</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.05</td>
<td>308.296</td>
<td>12</td>
<td>09.09.2013 17:58:21</td>
<td>244.962</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>9.136</td>
<td>SAPWRN</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.05</td>
<td>300.191</td>
<td>7</td>
<td>08.09.2013 21:22:41</td>
<td>0</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>9.136</td>
<td>SAPWRN</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.507</td>
<td>225</td>
<td>09.09.2013 18:00:06</td>
<td>129.754</td>
<td>IDLE</td>
<td>VMW00317</td>
<td>10.66.146.84</td>
<td>5.720</td>
<td>SYSTEM</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.504</td>
<td>216</td>
<td>09.09.2013 18:00:36</td>
<td>130.105</td>
<td>IDLE</td>
<td>VMW00317</td>
<td>10.66.146.84</td>
<td>4.460</td>
<td>SYSTEM</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.486</td>
<td>155</td>
<td>09.09.2013 17:58:08</td>
<td>68.550</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>3.536</td>
<td>SAPWRN</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.483</td>
<td>18</td>
<td>09.09.2013 17:58:21</td>
<td>145.962</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>9.136</td>
<td>SAPWRN</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.466</td>
<td>234</td>
<td>09.09.2013 17:50:36</td>
<td>699.745</td>
<td>IDLE</td>
<td>dswdfwp00399</td>
<td>10.68.84.140</td>
<td>1.644</td>
<td>SYSTEM</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.389</td>
<td>231</td>
<td>09.09.2013 17:44:32</td>
<td>921.070</td>
<td>IDLE</td>
<td>WDRN0027220878A.sap.corp</td>
<td>10.87.86.234</td>
<td>9.832</td>
<td>SYSTEM</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.377</td>
<td>24</td>
<td>09.09.2013 17:43:46</td>
<td>1.042.470</td>
<td>IDLE</td>
<td>WDRN0027220878A.sap.corp</td>
<td>10.87.86.234</td>
<td>9.832</td>
<td>SYSTEM</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.329</td>
<td>228</td>
<td>09.09.2013 17:21:53</td>
<td>2.391.539</td>
<td>IDLE</td>
<td>VEN062394682A.sap.corp</td>
<td>147.204.206.133</td>
<td>3.328</td>
<td>T028297</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.316</td>
<td>23</td>
<td>09.09.2013 17:20:22</td>
<td>2.452.038</td>
<td>IDLE</td>
<td>VEN062394682A.sap.corp</td>
<td>147.204.206.133</td>
<td>3.328</td>
<td>T028297</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.308</td>
<td>179</td>
<td>09.09.2013 17:19:40</td>
<td>2.496.476</td>
<td>IDLE</td>
<td>VEN062394682A.sap.corp</td>
<td>147.204.206.133</td>
<td>3.328</td>
<td>T028297</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.306</td>
<td>177</td>
<td>09.09.2013 17:19:38</td>
<td>2.426.900</td>
<td>IDLE</td>
<td>VEN062394682A.sap.corp</td>
<td>147.204.206.133</td>
<td>3.328</td>
<td>T028297</td>
<td>Rem</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.209</td>
<td>228</td>
<td>09.09.2013 17:04:29</td>
<td>0</td>
<td>IDLE</td>
<td>_SYS_AFL</td>
<td>_SY_S_AFL</td>
<td>Loc</td>
<td>_SYS_AFL</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.208</td>
<td>237</td>
<td>09.09.2013 17:04:29</td>
<td>0</td>
<td>IDLE</td>
<td>_SYS_AFL</td>
<td>_SY_S_AFL</td>
<td>Loc</td>
<td>_SYS_AFL</td>
<td>Loc</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>30.0.03</td>
<td>220.194</td>
<td>235</td>
<td>09.09.2013 17:00:54</td>
<td>0</td>
<td>IDLE</td>
<td>_SYS_AFL</td>
<td>_SY_S_AFL</td>
<td>Loc</td>
<td>_SYS_AFL</td>
<td>Loc</td>
</tr>
</tbody>
</table>
Connection information – SAP HANA Studio

<table>
<thead>
<tr>
<th>IDLE_TIME</th>
<th>CONNECTION_STATUS</th>
<th>CONNECTION_UNICODE</th>
<th>ID</th>
<th>CLIENT_HOST</th>
<th>CONNECTED_HOST</th>
<th>CLIENT_JD</th>
<th>CLIENT_PID</th>
<th>USER_NAME</th>
<th>CONNECTION_TYPE</th>
<th>OWN</th>
<th>IS_HISTORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.625.053</td>
<td>IDLE</td>
<td>VIEN20239482A.sap...</td>
<td>12</td>
<td>WDFN00287523A.sap...</td>
<td>10.19.27.176</td>
<td>16.376</td>
<td>SYSTEM</td>
<td>Remote</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>8.523.268</td>
<td>IDLE</td>
<td>VIEN20239482A.sap...</td>
<td></td>
<td>WDFN00287523A.sap...</td>
<td>10.19.27.176</td>
<td>16.376</td>
<td>SYSTEM</td>
<td>Remote</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>2.548.088</td>
<td>IDLE</td>
<td>VIEN20239482A.sap...</td>
<td>12</td>
<td>WDFN00287523A.sap...</td>
<td>10.19.27.176</td>
<td>16.376</td>
<td>SYSTEM</td>
<td>Remote</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>2.565.005</td>
<td>IDLE</td>
<td>VIEN20239482A.sap...</td>
<td></td>
<td>WDFN00287523A.sap...</td>
<td>10.19.27.176</td>
<td>16.376</td>
<td>SYSTEM</td>
<td>Remote</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>186.283</td>
<td>IDLE</td>
<td>YMWS217</td>
<td></td>
<td>10.55.146.84</td>
<td>5.720</td>
<td>SYSTEM</td>
<td>Remote</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>
Lifetime of a SAP HANA session

- With a session/connection ready, the application can send command text to SAP HANA.
- SQL commands really are DB program source code that needs to be parsed, validated, optimized and compiled before it can be executed.
- Similar to activation of ABAP reports
- This can take a lot of time and resources and need to be executed for every SQL command – so caching the compiled program is key for performance!
- Think of parameterized statements!

- Shared SQL plan cache keeps compiled and optimized statements in memory and reuses these for exact matching SQL commands
- Also provides execution statistics for performance analysis

indexserver.ini

[sql]
plan_cache_size

The "magic" happens here…
- Query transformation
- Query optimization and plan generation (1. Part!)
Shared SQL plan cache – SQL monitoring

- Most flexible analysis approach
- System views M_SQL_PLAN_CACHE and M_SQL_PLAN_CACHE_RESET provide lots of runtime statistics on statement level

```
select plan_id, statement_string, execution_count, avg_execution_time, accessed_objects, parameter_count, reference_count, plan_memory_size, avg_preparation_time, total_execution_fetch_time, preparation_count
from "M_SQL_PLAN_CACHE" pc
join m_services s on s.host = pc.host
and s.port = pc.port
where s.service_name = 'indexserver'
and pc.is_internal = 'FALSE'
and pc.user_name = '1028297'
```
Shared SQL plan cache – DBACockpit monitoring

- Easy filtering and sorting of the data
- Straight forward workflow to find e.g. the long runner SQL statements
- Build-in navigation to EXPLAIN PLAN, SQL Editor and PlanViz-File creation
Shared SQL plan cache – SAP HANA Studio monitoring

- Easy filtering and sorting of the data
- Straight forward workflow to find e.g. the long runner SQL statements
- Build-in navigation to Plan Visualization
Query plan information available after preparation

- Explain plan and its graphical representation represent the plan created and optimized by the SQL optimizer.
- It's based on runtime information about the data (e.g. number of records, data distribution etc.) as well as table statistics (if available).
- Upon actual execution, additional optimizations are applied by the execution engines (e.g. column pruning in the calculation engine, join ordering in the join engine, removal of superfluous joins, etc.)
- That means, the explain plan can not tell us, where time is spent during execution!
As a result of the parsing process the application gets a **handle** for the parsed statement.

With this handle the application can access the parsed statement directly and thereby save time (no text parsing required anymore to find the statement in the shared SQL plan cache).

SAP NetWeavers’ database interface (DBIF) keeps track of up to 1000 handles per work process.

ABAP developers automatically benefit from this feature.
• SAP HANA uses **multiple** execution engines
• All of them run upon the basic data structures: column store tables or row store tables

• Joins of column store tables ➔ **Join Engine** (JE-POPs)
• OLAP query support ➔ **OLAP Engine** (OE, BW-POPs)
• Complex calculations ➔ **Calculation engine** (CE-POPs)
• Rest is handled by SQL engine

• calling/opening/running a cursor ➔ provide parameter values and execute a prepared query

• “Prepare once - execute often” only possible with parameterized statements (or by asking the same question over and over again)!
• Massive time savings possible as no parsing required for similar queries with different filter condition values.
Parameterized Statements…?! 

```sql
select id, name from employees
where id = 1;
```

```sql
select id, name from employees
where id = 2;
```

- 2 similar, but different statements
- 2 x parsing, optimizing
- 2 x storing and keeping the execution plan in memory
- Can be executed only for the hard coded ID number.

```sql
select id, name from employees
where id = ?;
```

- Just 1 statement with a parameter
- 1 x parsing, optimizing
- 1 x storing and keeping the execution plan in memory
- Can be re-used indefinitely

Execute [F8]
How to know what is really executed?

- The execution engines perform additional optimizations when the parameter values are known.

- This can change the actual execution tremendously.

- Plan Visualization → Execute shows actual operation order, intermediate result set size, parallelism, biggest runtime contributors, etc.
Lifetime of SAP HANA session

- Once the result set is computed it’s kept at the SAP HANA server until the client closes the cursor.
  ➔ Memory allocation!
  ➔ Record versions need to be preserved if exclusive locks have been set.

- The client can decide not to fetch all records, e.g., to display only the first 10 result rows.
  ➔ better use `SELECT TOP 10... /LIMIT 10` for this!

- Closing the `cursor` removes the result set from the server.
- Closing the `statement handle` only decreases the reference counter of the statement in the shared SQL cache (doesn’t free anything immediately, but allows removal from cache when the statement is not referenced anymore by any session).
- Closing the `connection` ends the session and implicitly will release all locks again.
How does that look in the real world?

```java
Connection connection = myHANAconnection.getConnection();
ClientInfo myClientInfos = new ClientInfo(connection);

PreparedStatement pstmt = connection.prepareStatement
    ("select * from EFASHION.SHOP_FACTS where shop_facts_id = ?");

for (int i = 1; i <= LOOPCNT; i++) {
    pstmt.setInt(1, i);
    ResultSet resultSet = pstmt.executeQuery();
    while (resultSet.isAfterLast() == false) {
        resultSet.next();
    }
    resultSet.close();
}

pstmt.close();
connection.close();
System.exit(0);
```

vanilla JAVA code

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Object name</th>
<th>Operation</th>
<th>Curs</th>
<th>Array</th>
<th>Recs.</th>
<th>RC</th>
<th>Conn</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAPLP0_PACKAGE_SERVICES</td>
<td>TDEV</td>
<td>PREPARE</td>
<td>486</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>R/3</td>
<td>SELECT WHERE &quot;DEVCLASS&quot; = ?</td>
</tr>
<tr>
<td>SAPLP0_PACKAGE_SERVICES</td>
<td>TDEV</td>
<td>OPEN</td>
<td>486</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>R/3</td>
<td>SELECT WHERE &quot;DEVCLASS&quot; = 'SM&amp;P'</td>
</tr>
<tr>
<td>SAPLP0_PACKAGE_SERVICES</td>
<td>TDEV</td>
<td>FETCH</td>
<td>486</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>R/3</td>
<td>SELECT WHERE &quot;DEVCLASS&quot; = ?</td>
</tr>
</tbody>
</table>

SAP NetWeaver SQL trace (ST05)
Performance Analysis Tools

What tools do you use when...
Trace options in SAP HANA Studio

Default settings for traces

Actual system wide trace settings

Actual user specific trace settings

Setup for NetWeaver E2E-traces via SAP PASSPORT

System wide or user specific SQL trace

SAP internal performance trace & profiler

Expensive statement trace

SAP internal sampling profiler
Trace options in SAP HANA Studio

- SQLTRACE (PYTHON TRACE)
- PERFORMANCE TRACE + FUNCTION PROFILER
- KERNEL PROFILER
- OPTIMIZER TRACES
- LOAD GRAPH
- STATISTICS SERVER
- EXPENSIVE STATEMENTS TRACE
- SQL PLAN CACHE
- EXPLAIN PLAN
- PLAN VISUALIZATION
SQLTRACE (PYTHON TRACE)

• Incomprehensible w/o SAP HANA development knowledge

• Is like “taking a sledgehammer to crack a nut”

• Doesn't provide what happens during the execution (rather input → output trace)
PERFORMANCE TRACE + FUNCTION PROFILER

- TREX heritage
- unsupported for non-SAP-Development usage
- requires HANA server access + X11-Window client
- requires special privileges
- doesn't cover row store activity
- tool is barely documented and complex
- function profiler is useless w/o source code
KERNEL PROFILER

- stack code sampling

- profiler tells you how often a specific part of the SAP HANA code was executed and how long that took (approx.)

- again, useless w/o source code
OPTIMIZER TRACES

- sqlopttime (runtime information for the query optimization phase)
- sqloptStep (single query optimization/transformation steps)
- trex_qo (TREX query optimizer → column store plans)
Trace options in SAP HANA Studio

- SQLTRACE (PYTHON TRACE)
- PERFORMANCE TRACE + FUNCTION PROFILER
- KERNEL PROFILER
- OPTIMIZER TRACES
- LOAD GRAPH
- STATISTICS SERVER
- EXPENSIVE STATEMENTS TRACE
- SQL PLAN CACHE
- EXPLAIN PLAN
- PLAN VISUALIZATION

pretty useless for mere mortals like us…
LOAD GRAPH

- Nameserver data (`nameserver_history.trc`)
- TREX heritage
- Highly aggregated data - cannot (easily) be mapped to a specific user action
- Yet, give a "feeling" of how workload "looks like"
- Could be used to give a rough baseline
- No function to compare to time frames side-by-side or by overlay
- MIN/MAX/AVG numbers are highly volatile and fully depend on the chosen time frame
LOAD GRAPH – Screenshot

CPU spikes ca. every 15 min.?
Spot the pattern?

<table>
<thead>
<tr>
<th>Host</th>
<th>Time Begin</th>
<th>Time End</th>
<th>KPI</th>
<th>Y-Scale</th>
<th>Unit</th>
<th>Max</th>
<th>Average</th>
<th>Sum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmB012</td>
<td>21.03.2013 03:01</td>
<td>01.04.2013</td>
<td>CPU</td>
<td>%</td>
<td>93</td>
<td>1</td>
<td>-</td>
<td>CPU usage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Memory Size</td>
<td>MB</td>
<td>64.564</td>
<td>64.564</td>
<td>-</td>
<td>Physical</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Disk Used</td>
<td>GB</td>
<td>1.105</td>
<td>1.101</td>
<td>-</td>
<td>Disk Use</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Disk Size</td>
<td>GB</td>
<td>1.258</td>
<td>1.258</td>
<td>-</td>
<td>Disk Size</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Network In</td>
<td>MB/...</td>
<td>146</td>
<td>0,3</td>
<td>26.856</td>
<td>Bytes r</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Network Out</td>
<td>MB/...</td>
<td>18</td>
<td>0</td>
<td>799</td>
<td>Bytes w</td>
<td></td>
</tr>
</tbody>
</table>
STATISTICS SERVER

- _SYS_STATISTICS schema
- Lots of data, gathered by automatic snapshots
- Are purged automatically
- Is used as the data source for solution manager (performance warehouse)
- Provide alerts for SAP HANA
STATISTICS SERVER
EXPENSIVE STATEMENTS TRACE

• Provides information about runtimes of single statement executions
• Captures the parameter values for parameterized statements
• Can be easily configured to focus on the real long running statements
• What long running statements are, clearly depends on your use case.
• Sometimes this means milliseconds, sometimes it's minutes
EXPENSIVE STATEMENTS TRACE – Setup Screen
EXPENSIVE STATEMENTS TRACE – Navigation

![Diagram of the EXPENSIVE STATEMENTS TRACE interface with highlighted actions: double click and right click.]
SQL PLAN CACHE

• Contains two kind of information:
  1. parsed statements
  2. call and runtime statistics for these statements

• Can answer questions like:
  • "How much time was spent on actually executing the statement?"
  • "Can my statement be reused or does it need to be optimized over and over again?"

• Depending on the use case, the default cache size can be much too small
  • We're gaining experience with that right now
Trace options in SAP HANA Studio

- SQLTRACE (PYTHON TRACE)
- PERFORMANCE TRACE + FUNCTION PROFILER
- KERNEL PROFILER
- OPTIMIZER TRACES
- LOAD GRAPH
- STATISTICS SERVER
- EXPENSIVE STATEMENTS TRACE
- SQL PLAN CACHE
- EXPLAIN PLAN
- PLAN VISUALIZATION

System wide monitoring – the BIG picture
EXPLAIN PLAN

• Can't look "into" modeled views!

• Raw formatting

• Focusses on "costs" and estimated cardinalities (no. of rows)

• First information on which **engine** is used
EXPLAIN PLAN – Demo 1

SELECT TOP 20
"ARTICLE_LABEL_LOCALE",
    sum("MARGIN") AS "MARGIN",
    sum("AMOUNT_SOLD") AS "AMOUNT_SOLD",
    sum("QUANTITY_SOLD") AS "QUANTITY_SOLD"
FROM "_SYS_BIC"."i028297.demo/AN_SALES"
WHERE "FAMILY_NAME" IN ('City Trousers', 'City Skirts')
GROUP BY
    "ARTICLE_LABEL_LOCALE"
ORDER BY "AMOUNT_SOLD" desc;

<table>
<thead>
<tr>
<th>ARTICLE_LABEL_LOCALE</th>
<th>MARGIN</th>
<th>AMOUNT_SOLD</th>
<th>QTY_SOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side Slit Long Skirt</td>
<td>64.437,700000000004</td>
<td>130.569,09999999992</td>
<td>598</td>
</tr>
<tr>
<td>Lycra Trousers</td>
<td>45.848,600000000006</td>
<td>105.164,09999999992</td>
<td>600</td>
</tr>
<tr>
<td>Corduroy Shorts</td>
<td>26.852,000000000015</td>
<td>76.395,49999999994</td>
<td>429</td>
</tr>
</tbody>
</table>
...
EXPLAIN PLAN – Demo 2

```
SELECT TOP 20
    "ARTICLE_LABEL_LOCALE",
    SUM("MARGIN") AS "MARGIN",
    SUM("AMOUNT_SOLD") AS "AMOUNT_SOLD",
    SUM("QUANTITY_SOLD") AS "QUANTITY_SOLD"
FROM "SYS_DIC".t028297.demo/AN_SALES"
WHERE "FAMILY_NAME" IN ('City Trousers', 'City Skirts')
GROUP BY
    "ARTICLE_LABEL_LOCALE"
ORDER BY "AMOUNT_SOLD" desc;
```

right click

<table>
<thead>
<tr>
<th>Command</th>
<th>Keyboard Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo Typing</td>
<td>Ctrl-Z</td>
</tr>
<tr>
<td>Revert File</td>
<td>Ctrl-S</td>
</tr>
<tr>
<td>Save</td>
<td></td>
</tr>
<tr>
<td>Save As...</td>
<td></td>
</tr>
<tr>
<td>Show In</td>
<td>Alt+Shift+W</td>
</tr>
<tr>
<td>Cut</td>
<td>Ctrl-X</td>
</tr>
<tr>
<td>Copy</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Paste</td>
<td>Ctrl-V</td>
</tr>
<tr>
<td>Open File...</td>
<td></td>
</tr>
<tr>
<td>Preferences...</td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>F8</td>
</tr>
<tr>
<td>Prepare Statement</td>
<td>Ctrl+Shift+P</td>
</tr>
<tr>
<td>Explain Plan</td>
<td>Ctrl+Shift+Z</td>
</tr>
<tr>
<td>Visualize Plan</td>
<td>Ctrl+Shift+V</td>
</tr>
<tr>
<td>Choose Connection</td>
<td></td>
</tr>
<tr>
<td>Clear</td>
<td></td>
</tr>
<tr>
<td>Close All Results</td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td></td>
</tr>
<tr>
<td>Show Definition</td>
<td></td>
</tr>
</tbody>
</table>
Which Engines are used to process the query?
### EXPLAIN PLAN – Demo 4

<table>
<thead>
<tr>
<th>OPERATOR_NAME</th>
<th>OPERATOR_DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN SEARCH</td>
<td>i028297.demo/AN_SALES ARTICLE_LABEL_LOCALE, SUM(i028297.demo/AN_SALES.MARGIN), SUM(i028297.demo/AN_SALES.AMOUNT_SOLD), SUM(i028297.demo/AN_SALES.QUANTITY_SOLD)</td>
</tr>
<tr>
<td>LIMIT</td>
<td>NUM RECORDS: 20</td>
</tr>
<tr>
<td>ORDER BY</td>
<td>SUM(i028297.demo/AN_SALES.AMOUNT_SOLD) DESC</td>
</tr>
<tr>
<td>AGGREGATION</td>
<td>GROUPING: i028297.demo/AN_SALES ARTICLE_LABEL_LOCALE, SUM(i028297.demo/AN_SALES.MARGIN), SUM(i028297.demo/AN_SALES.AMOUNT_SOLD), SUM(i028297.demo/AN_SALES.QUANTITY_SOLD)</td>
</tr>
<tr>
<td>COLUMN VIEW</td>
<td>FILTER CONDITION: i028297.demo/AN_SALES.FAMILY_NAME = 'City Trousers' OR i028297.demo/AN_SALES.FAMILY_NAME = 'City Skirts'</td>
</tr>
</tbody>
</table>
EXPLAIN PLAN – Demo 5

<table>
<thead>
<tr>
<th>OPERATOR_NAME</th>
<th>EXEC_ENGINE</th>
<th>SCHEMA_NAME</th>
<th>TABLE_NAME</th>
<th>TABLE_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN SEARCH</td>
<td></td>
<td></td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

| LIMIT               | COLUMN      | NULL        | NULL       | NULL       |
| ORDER BY            | COLUMN      | NULL        | NULL       | NULL       |
| AGGREGATION         | COLUMN      | NULL        | NULL       | NULL       |

| COLUMN VIEW         | COLUMN      | NULL        | NULL       | NULL       |

| COLUMN              | _SYS_BIC    | i028297.demo/AN_SALES | OLAP VIEW   |
PLAN VISUALIZATION

- Relatively easy to use
- Graphical tool, build into SAP HANA studio
- Provides estimation and actual runtime statistics for memory, CPU time, parallelism and total runtime.
- Can be access via SQL Editor, SQL statement cache view, Expensive statement view
- Can be saved (XML file) and reviewed later on
- THE TOOL OF CHOICE!
Plan Visualization (PlanViz)
PlanViz
PlanViz – display timelines (pre-SPS 06)
PlanViz – display timelines
PlanViz – display timelines (SPS 06)

• New PlanViz Perspective
Trace options in SAP HANA Studio

- SQLTRACE (PYTHON TRACE)
- PERFORMANCE TRACE + FUNCTION PROFILER
- KERNEL PROFILER
- OPTIMIZER TRACES
- LOAD GRAPH
- STATISTICS SERVER
- EXPENSIVE STATEMENTS TRACE
- SQL PLAN CACHE
- EXPLAIN PLAN
- PLAN VISUALIZATION

Single statement analysis
Summary

• Use SAP HANA database as a service, but not as a black box!
• In case of performance problems, think about where the problem could occur.
• Use the proper tools to measure if the problem really is where you think it is.
Further Information

SAP Public Web

SCN:
SAP In-Memory computing: http://scn.sap.com/community/hana-in-memory
SAP HANA Developer center: http://scn.sap.com/community/developer-center/hana

Further important information sources:
Public HANA community: http://www.saphana.com/
Public documentation: http://help.sap.com/hana_platform

SAP Education and Certification Opportunities

SAP HANA Training Curriculum

Watch SAP TechEd Online
www.sapteched.com/online
SAP TechEd Virtual Hands-on Workshops and SAP TechEd Online
Continue your SAP TechEd education after the event!

SAP TechEd Virtual Hands-on Workshops
• Access hands-on workshops post-event
• Available January – March 2014
• Complementary with your SAP TechEd registration

http://saptechedhandson.sap.com/

SAP TechEd Online
• Access replays of keynotes, Demo Jam, SAP TechEd LIVE interviews, select lecture sessions, and more!
• View content only available online

http://sapteched.com/online
Feedback

Please complete your session evaluation for RDP302.

Thanks for attending this SAP TechEd session.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG.
The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies (“SAP Group”) for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.