Multitenant – Is it worth it?
About me

Oracle DBA since 2000 and Version 7.3.4
Certified Professional 10g, 11g
RAC / Cluster Certified Expert 10g, 11g, 12c
Oracle ACE
@DBAMarco
dbamarco.wordpress.com
### Facts and Figures
**Robotron Datenbank-Software GmbH**

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year of formation</td>
<td>1990</td>
</tr>
<tr>
<td>Legal form</td>
<td>GmbH (Limited liability company, 8 associates)</td>
</tr>
<tr>
<td>Number of employees</td>
<td>442 (Status 11/2017)</td>
</tr>
<tr>
<td>Capital stock</td>
<td>2.4 million EUR</td>
</tr>
<tr>
<td>Turnover 2017</td>
<td>43.8 million EUR</td>
</tr>
<tr>
<td>Technology-Partner</td>
<td><img src="image" alt="Oracle Platinum Partner" /> <img src="image" alt="Oracle Approved Education Center" /> <img src="image" alt="Gold Microsoft Partner" /></td>
</tr>
<tr>
<td>ISO 9001 certified</td>
<td><img src="image" alt="DeuZert" /></td>
</tr>
</tbody>
</table>
The Robotron Company Group

Czech Rep.: Robotron Database Solutions s.r.o.

Switzerland: Robotron Schweiz GmbH

Austria: Robotron Austria GmbH

Russia: Robotron Rus GmbH

Germany: SASKIA® Informations-Systeme GmbH
The Range of Services of Robotron with branch-specific Expertise

- Methodical and technological responsibility
- Comprehensive expertise of industry-specific business processes
Agenda

- Legacy Operation
- Schema Level Consolidation
- Multitenant Consolidation
- Conclusion
Legacy Operation
Initial Situation

- What we had
  - 1 production database
  - 2-3 test databases - for end user tests
  - 2-4 integration databases - for testing different software releases
  - 4-6 education databases - for educating employees
  - That’s 9-14 databases to manage
- That’s all the same application
Operational Tasks

- What are the typical tasks
  - Database driven
    - Install quarterly patches
    - Upgrades
  - Application driven
    - Install new software releases (schema updates) - all levels
    - Reset schema/content to a specific point - integration / education
    - Update database with production data - test / integration / education
Legacy Operation – Time Consumption

- Install Bundle Patch
  - Approx. 2h / database system
    - Shutdown application servers
    - Apply patch
    - Startup application servers
    - Quick functional test
  - 18-24h of work in total!
Legacy Operation – Update Schema / Baseline

- Apply Application Patch
  - 4h / database system

Sequence:
1. Backup DB
2. Install Update
3. OK?
   - Yes: Done
   - No: Restore DB
Legacy Operation – Refresh Schema

- Refresh Schema from Prod/Baseline
  - 4-8h / database system
Legacy Operation - Caveats

- Hardware Resource Usage
  - CPU is not an issue, what’s there can be used
  - Storage overhead for system files (System tablespaces, Redo, ...)
  - RAM usage, difficult to optimally utilize
    - Number of DBs unknown at time of initial setup
    - in/decrease SGA requires DB restart
Legacy Operation – RAM utilization

Available RAM (64GB)

SGA: 24GB
Legacy Operation – RAM utilization

Available RAM (64GB)

SGA: 24GB

16GB
Available RAM (64GB)

- SGA: 24GB
- 16GB
- 12GB
Legacy Operation – RAM utilization

Available RAM (64GB)

SGA: 24GB

16GB

12GB

8GB
Legacy Operation – Summary

- Lot of manual tasks for several databases
  - Install database patches
  - Install application patches
  - Refresh database from prod/baseline
  - Monitoring
  - Backup

- Complex tasks, error prone
  - Install application patches
  - Refresh database from prod/baseline
Schema Consolidation
First Consolidation Ideas

- Application allows to configure the schema name 😊 kind of unique and very fortunate 😊
  - Consolidate DBs for one purpose into one DB with multiple schemas

- Pro
  - Lesser databases
    - 1 Production – 1 schema
    - 1 Test – 2-3 schemas
    - 1 Integration – 2-4 schemas
    - 1 Education – 4-6 schemas
  - Lesser time for database patching etc.
  - Better RAM utilization
Changes in tasks

- Backup/Restore/Duplicate
  → replaced by DataPump
Schema Consolidation - Changes

- Changes in tasks
  - Backup/Restore/Duplicate
    → replaced by DataPump

- Refresh Application
  - Drop Database → Drop Schema
  - ... → Export Schema
  - Duplicate Database → Import Schema

- Time to complete is longer
  - approx. 8h / schema
Schema Consolidation - Changes

- Changes in tasks
  - Backup/Restore/Duplicate → replaced by DataPump

- Refresh Application
  - Drop Database → Drop Schema
  - ... → Export Schema
  - Duplicate Database → Import Schema

- Time to complete is longer
  - approx. 8h / schema

- Patch Application
  - Backup Database → Export Schema
  - Patch Application

- Time to complete is similar (good case)
  - approx. 4h / schema
Multitenant Consolidation
Why Multitenant?

- Migration to 12.1 due to Oracle Support Policy
- Single Tenant was deprecated in 12.1
- Need to build up knowledge for new technology
- Need to migrate from single- to multitenant sometime anyway
- So why not start with multitenant right away?
Mulitenant Consolidation

- Consolidate Schemas for one purpose into one CDB with multiple PDBs

- Pro
  - At the first glance equal to schema consolidation:
  - Lesser databases
    - 1 Production – 1 PDB
    - 1 Test – 2-3 PDBs
    - 1 Integration – 2-4 PDBs
    - 1 Education – 4-6 PDBs
  - Lesser time for database patching etc.
  - Better RAM utilization
At the second glance, all tasks become much easier
  - It all comes down to

  create pluggable database <PDB> ...
Mulitenant Consolidation - Benefits

- At the second glance, all tasks become much easier
  - It all comes down to
    
    create pluggable database <PDB> ...

- Additional Pros
  - Much less error prone
  - Much less time consuming
  - (nearly) all tasks can be accomplished with 12.1 mutlitenant features
- Refresh from Baseline
  - 1h / application

- as of 12.1 master PDB must be read-only
  - Not feasible for refresh from Prod
  - Limitation has been lifted in 12.2
1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

1. Preparation (PDB)
   - None
Refresh from Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Drop existing Schema
   drop user <schema> cascade;

1. Preparation (PDB)
   - None

2. Drop existing PDB
   drop pluggable database <PDB>;}
Refresh from Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Drop existing Schema
   drop user <schema> cascade;

3. Export Baseline
   expdp parfile=base_exp.par
   <one-time parameters>

1. Preparation (PDB)
   - None

2. Drop existing PDB
   drop pluggable database <PDB>;

3. Have a cup of coffee
Refresh from Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Drop existing Schema
   drop user <schema> cascade;

3. Export Baseline
   expdp parfile=base_exp.par
   <one-time parameters>

4. Import Baseline as new schema
   impdp parfile=base_imp.par
   <one-time parameters>

1. Preparation (PDB)
   - None

2. Drop existing PDB
   drop pluggable database <PDB>;

3. Have a cup of coffee

4. Clone Baseline PDB as new PDB
   create pluggable database <PDB>
   from <Base-PDB>;
   alter pluggable database <PDB> open;
Update Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

1. Preparation (PDB)
   - None
Update Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Export Baseline
   expdp parfile=base_exp.par
   <one-time parameters>

1. Preparation (PDB)
   - None

2. Clone Baseline PDB as new PDB
   create pluggable database <PDB>
   from <Base-PDB>;
Update Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Export Baseline
   expdp parfile=base_exp.par
   <one-time parameters>

3. Update Application

1. Preparation (PDB)
   - None

2. Clone Baseline PDB as new PDB
   create pluggable database <PDB>
   from <Base-PDB>;

3. Update Application in Clone PDB
Update Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Export Baseline
   expdp parfile=base_exp.par
   <one-time parameters>

3. Update Application

4. Failure: Drop schema and reimport
   drop user <schema> cascade;
   impdp parfile=base_imp.par
   <one-time parameters>

1. Preparation (PDB)
   - None

2. Clone Baseline PDB as new PDB
   create pluggable database <PDB>
   from <Base-PDB>;

3. Update Application in Clone PDB

4. Failure: Drop cloned PDB
   drop pluggable database <PDB>;}
Update Baseline - Comparison

1. Preparation (schema)
   - Create directory
   - Prepare Datapump default parameters

2. Export Baseline
   expdp parfile=base_exp.par
   <one-time parameters>

3. Update Application

4. Failure: Drop schema and reimport
   drop user <schema> cascade;
   impdp parfile=base_imp.par
   <one-time parameters>
   Success: n/a

1. Preparation (PDB)
   - None

2. Clone Baseline PDB as new PDB
   create pluggable database <PDB>
   from <Base-PDB>;

3. Update Application in Clone PDB

4. Failure: Drop cloned PDB
   drop pluggable database <PDB>;
   Success: Exchange PDB with cloned PDB
   drop pluggable database <Base-PDB>;
   alter pluggable database <PDB>
   rename to <Base-PDB>;
Operational Changes

- PDBs are accessible via
  - Listener + Service
  - Local connect + alter session set container
Operational Changes

- PDBs are accessible via
  - Listener + Service
  - Local connect + alter session set container

- What about application?
Operational Changes

- PDBs are accessible via
  - Listener + Service
  - Local connect + alter session set container

- What about application?
  - Fine: application is using host:port/service anyway
Operational Changes

- PDBs are accessible via
  - Listener + Service
  - Local connect + alter session set container

- What about application?
  - Fine: application is using host:port/service anyway

- What about DBA scripts?
  - They do „connect / as sysdba“ all over

```bash
#!/bin/sh
export ORACLE_SID=MYDB
export ORACLE_HOME=/u01/app
LOGFILE=/tmp/logfile.txt
ORACLE_HOME/bin/sqlplus -s OR
connect / as sysdb
set pages 0
set heading off
set feedback off
spool $LOGFILE
select SYSDATE from dual;
spool off
exit
```

Operational Changes

- PDBs are accessible via
  - Listener + Service
  - Local connect + alter session set container

- What about application?
  - Fine: application is using host:port/service anyway

- What about DBA scripts?
  - They do „connect / as sysdba“ all over
  - With multitenant we end up in the CDB instead of PDB
Operational Changes

- First approach
  - Touch every single script and add „alter session set container“ + extra handling?
Operational Changes

- First approach
  - Touch every single script and add "alter session set container" + extra handling?

- Anyone remembers TWO_TASK?
Operational Changes

- First approach
  - Touch every single script and add „alter session set container“ + extra handling?

- Anyone remembers TWO_TASK?

- TWO_TASK defines the default TNSNAMES.ORA alias to use
  - connect sys/password as sysdba → connect sys/password@$\{TWO\_TASK\} as sysdba

  - Just replace ORACLE\_SID with TWO\_TASK and change the connection to <user>/<password>
Conclusion
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th><strong>Standard</strong></th>
<th><strong>Schema Consolidation</strong></th>
<th><strong>Multitenant Consolidation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Patching</td>
<td>9-14 DBs $\triangleq$ 18-28h</td>
<td>3-5 DBs $\triangleq$ 6-10h</td>
<td>3-5 DBs $\triangleq$ 6-10h</td>
</tr>
<tr>
<td>Application Patching</td>
<td>4h / application</td>
<td>4h / application</td>
<td>1h / application</td>
</tr>
<tr>
<td>Refresh from Prod/Baseline</td>
<td>4-8h / application</td>
<td>Min. 8h / application</td>
<td>1h / application</td>
</tr>
<tr>
<td>RAM consumption</td>
<td>Suboptimal</td>
<td>Optimal</td>
<td>optimal</td>
</tr>
<tr>
<td>Task Complexity</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
</tbody>
</table>
Conclusion

- Assume the following
  - Quarterly database patches, 4 patches / DB + year

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Schema Consolidation</th>
<th>Multitenant Consolidation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Patching</td>
<td>9-14 DBs (\triangleq) 18-28h x 4 ≈ 92h</td>
<td>3-5 DBs (\triangleq) 6-10h x4 ≈ 32h</td>
<td>3-5 DBs (\triangleq) 6-10h ≈ 32h</td>
</tr>
</tbody>
</table>
### Conclusion

- Asume the following
  - Quarterly database patches, 4 patches / DB + year
  - 6-8 application patches / DB + year (2-4 int DBs)

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Schema Consolidation</th>
<th>Multitenant Consolidation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Patching</td>
<td>9-14 DBs $\triangleq$ 18-28h x 4 $\approx$ 92h</td>
<td>3-5 DBs $\triangleq$ 6-10h x4 $\approx$ 32h</td>
<td>3-5 DBs $\triangleq$ 6-10h $\approx$ 32h</td>
</tr>
<tr>
<td>Application Patching</td>
<td>4h / app; 2-4DBs x 6-8 $\approx$ 140h</td>
<td>4h / app; 2-4DBs x 6-8 $\approx$ 140h</td>
<td>1h / app; 2-4DBs x 6-8 $\approx$ 35h</td>
</tr>
</tbody>
</table>
### Conclusion

- Assume the following
  - Quarterly database patches, 4 patches / DB + year
  - 6-8 application patches / DB + year (2-4 int DBs)
  - 10-12 education DB refreshes / DB + year (4-6 edu DBs)

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Schema Consolidation</th>
<th>Multitenant Consolidation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Patching</td>
<td>9-14 DBs ( \triangleq ) 18-28h x 4 ( \approx 92h )</td>
<td>3-5 DBs ( \triangleq ) 6-10h x 4 ( \approx 32h )</td>
<td>3-5 DBs ( \triangleq ) 6-10h ( \approx 32h )</td>
</tr>
<tr>
<td>Application Patching</td>
<td>4h / app; 2-4DBs x 6-8 ( \approx 140h )</td>
<td>4h / app; 2-4DBs x 6-8 ( \approx 140h )</td>
<td>1h / app; 2-4DBs x 6-8 ( \approx 35h )</td>
</tr>
<tr>
<td>Refresh from Baseline</td>
<td>4-8h / app; 4-6DBs x 10-12 ( \approx 330h )</td>
<td>Min. 8h / app; 4-6DBs x 10-12 ( \approx 440h )</td>
<td>1h / app; 4-6DBs x 10-12 ( \approx 55h )</td>
</tr>
</tbody>
</table>
## Conclusion

- Asume the following
  - Quarterly database patches, 4 patches / DB + year
  - 6-8 application patches / DB + year (2-4 int DBs)
  - 10-12 education DB refreshes / DB + year (4-6 edu DBs)

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Schema Consolidation</th>
<th>Multitenant Consolidation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database Patching</strong></td>
<td>9-14 DBs (\triangleq) 18-28h x 4 (\approx) 92h</td>
<td>3-5 DBs (\triangleq) 6-10h x 4 (\approx) 32h</td>
<td>3-5 DBs (\triangleq) 6-10h (\approx) 32h</td>
</tr>
<tr>
<td><strong>Application Patching</strong></td>
<td>4h / app; 2-4DBs x 6-8 (\approx) 140h</td>
<td>4h / app; 2-4DBs x 6-8 (\approx) 140h</td>
<td>1h / app; 2-4DBs x 6-8 (\approx) 35h</td>
</tr>
<tr>
<td><strong>Refresh from Baseline</strong></td>
<td>4-8h / app; 4-6DBs x 10-12 (\approx) 330h</td>
<td>Min. 8h / app; 4-6DBs x 10-12 (\approx) 440h</td>
<td>1h / app; 4-6DBs x 10-12 (\approx) 55h</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td><strong>562h</strong></td>
<td><strong>612h (+50h)</strong></td>
<td><strong>122h (-440h)</strong></td>
</tr>
</tbody>
</table>
Conclusion

- 4-6 test DB refreshes / DB and year (2-3 test DBs), no change so not included → even more improvement with 12.2

- Much less administrative tasks, difficult to measure

- Some additional effort to adopt scripts
Conclusion

- 4-6 test DB refreshes / DB and year (2-3 test DBs), no change so not included
  - even more improvement with 12.2

- Much less administrative tasks, difficult to measure

- Some additional effort to adopt scripts

It is worth it!
Marco Mischke
Group Lead Database Projects

@DBAMarco
dbamarco.wordpress.com

+49 351 25859-2884
marco.mischke@robotron.de

www.robotron.de

Questions?