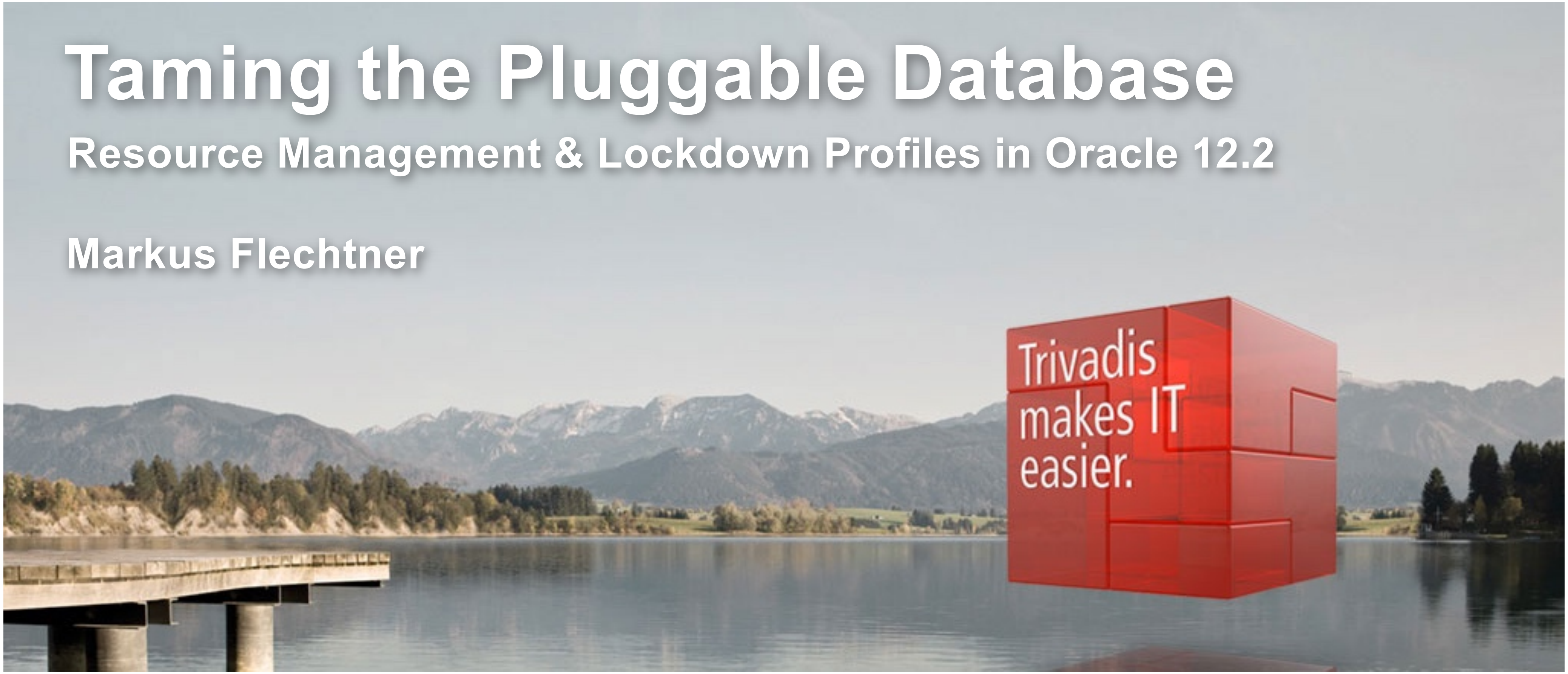
Taming the Pluggable Database

Resource Management & Lockdown Profiles in Oracle 12.2

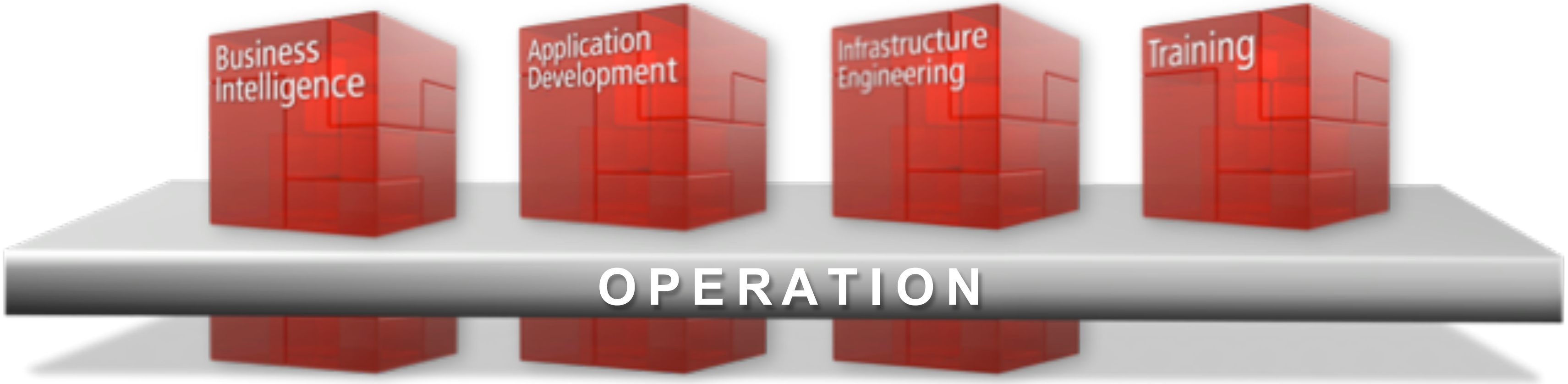
Markus Flechtner



BASLE ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

Our company.

Trivadis is a market leader in IT consulting, system integration, solution engineering and the provision of IT services focusing on ORACLE and Microsoft technologies in Switzerland, Germany, Austria and Denmark. We offer our services in the following strategic business fields:



Trivadis Services takes over the interactive operation of your IT systems.

About Markus Flechtner

- Principal Consultant, Trivadis, Duesseldorf/Germany, since April 2008
- Discipline Manager Infrastructure Database @Trivadis
- Working with Oracle since the 1990’s
  - Development (Forms, Reports, PL/SQL)
  - Support
  - Database Administration
- Focus
  - Oracle Real Application Clusters
  - Database Upgrade and Migration Projects
- Teacher
  - O-RAC – Oracle Real Application Clusters
  - O-NF12CDBA – Oracle 12c New Features for the DBA

Blog:
https://markusdba.net/

@markusdba

Motivation

- In a Multitenant Database, ..
  - .. access to common resources like OS and network should be restricted
  - .. the system resources like CPU, memory and I/O should be distributed among the PDBs in a controlled way
  - .. charging the customers by resource usage (storage, I/O, CPU) should be possible

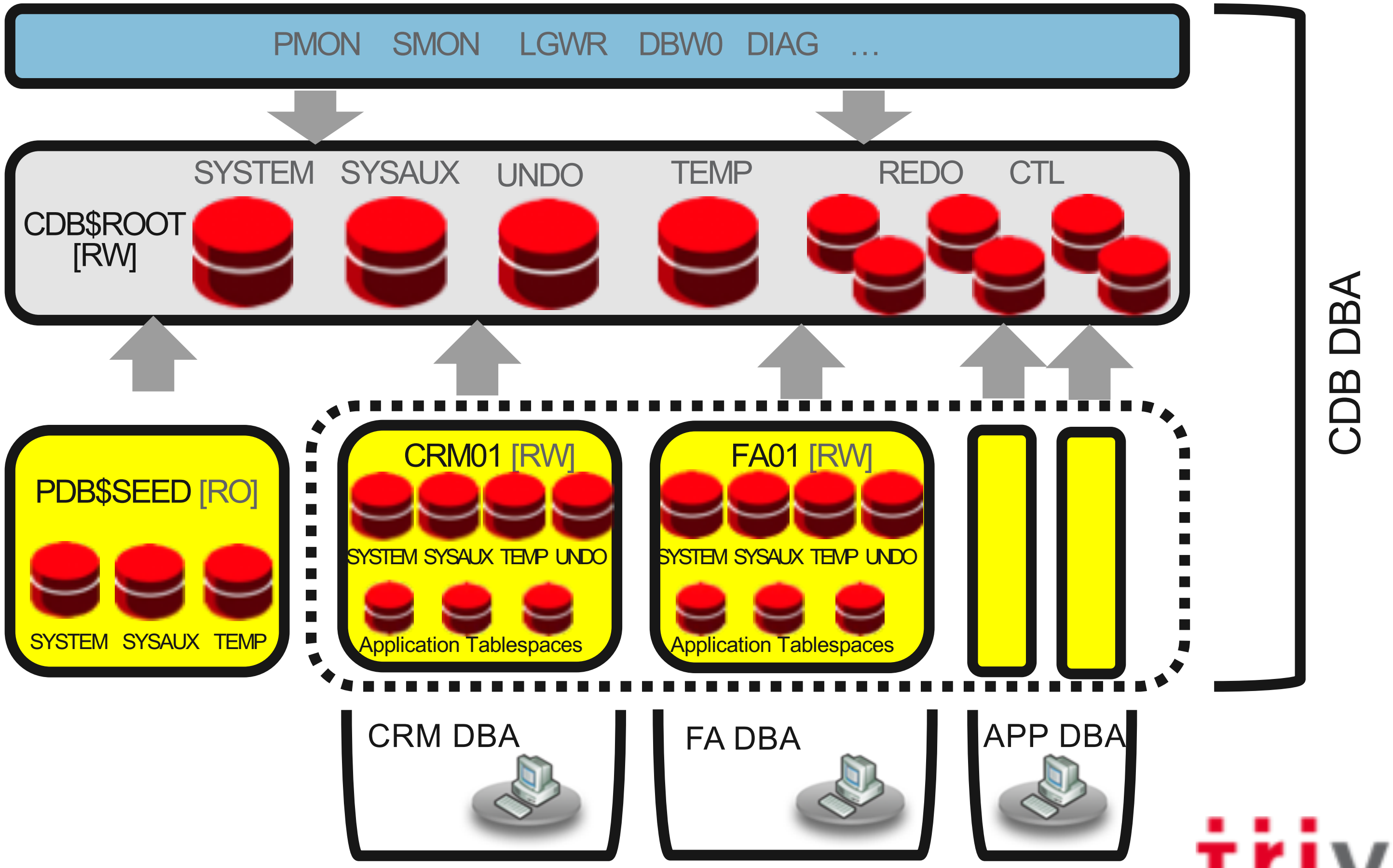
- Oracle Database 12c Release 2 brings improvements in all three areas

Agenda

1. **Container Database Architecture**
2. **Resource Management**
   - I/O
   - CPU
   - Memory
3. **Lockdown Profiles**
4. **Miscellaneous**
5. **Summary**

Container Database Architecture

Container Database Architecture (1) – The Big Picture

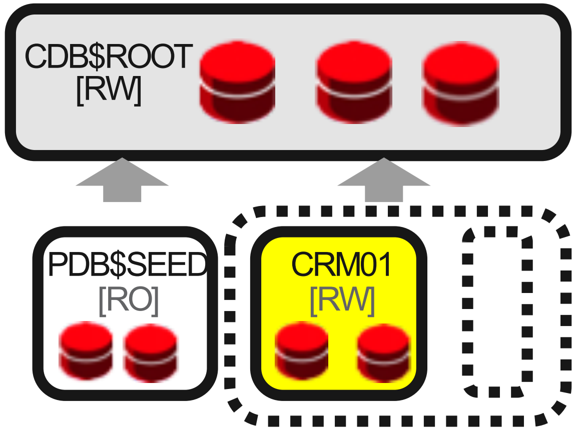


Container Database Architecture (2)

- The new Container Database Architecture (“**multitenant architecture”)** enables an Oracle database to work as a **container database** (CDB)
- A CDB consists of:
  - The root container **CDB$ROOT** – stores system metadata and common users/roles
  - The seed container **PDB$SEED** – default template used for cloning (cannot be modified)
  - Up to **252** user created **Pluggable Databases** (PDB) (in 12.2 up to 4096 on Exadata and Oracle Cloud)
- A new database architecture designed for:
  - consolidation/database virtualization
  - fast and easy provisioning
  - separation of administrative duties
  - rapid movement of user data (unplug/plug)
- Pluggable databases are **compatible** with traditional non-CDB (same behaviour from the application point of view)

Container Database Architecture (3)

- All database containers **share** the default block size
- PDBs created in a CDB share the character set with CDB$ROOT
  - PDBs which are plugged into a CDB can have a different character set (binary subset of the CDB$ROOT-character set) (12.2)
- All PDBs **belong** to the same CDB$ROOT container and **share**:
  - the Oracle instance (background processes, shared memory segment, semaphores)
  - the system metadata stored in the ROOT (Oracle-supplied PL/SQL code, metadata for the data dictionary objects)
  - redo thread(s) and control files
  - UNDO tablespace (12.1, still available in 12.2 but not recommended)
- Each PDB
  - has its own SYSTEM, SYSAUX and UNDO (12.2) tablespace
  - has it's own temporary tablespace
  - is a self-contained set of database data files



Resource Management

Resource Management for Container Databases (1)

- Adequate resource management is essential for Multitenant Databases
- Very often, with DBaaS Service Level Agreements (SLAs) guarantee a certain amount of resources
- Oracle can manage the following resources
  - CPU
  - Number of parallel server processes
  - Memory (since Oracle 12.2)
  - I/O (since Oracle 12.2)
- On Exadata and Oracle Super Cluster I/O-management is available since Oracle 12.1
  - Not covered in this presentation

Resource Management for Container Databases (2)

- For some of the resource limits, the following procedure applies
  - Resource limits are **defined** in CDB$ROOT
  - Resource limits are **activated** in the PDB ("ALTER SYSTEM ..")
  - PDB has to be bounced (close/open) after defining the limit on PDB level
- To prevent a PDB administrator from disabling these resource limits, PDB resource management should be combined with Lockdown Profiles
  - Disable "ALTER SYSTEM .." on PDB level

Managing CPU Resources

- Two ways to limit CPU resources
  - Parameter CPU_COUNT (on PDB level) – since Oracle 12.2
  - Resource Manager
- Parameter CPU_COUNT
  - Dynamic parameter
  - Allows instance caging on PDB level
  - Resource Manager must be enabled (RESOURCE_MANAGER_PLAN set)

```
SQL> ALTER SYSTEM SET cpu_count = 4;
```

Resource Manager 12c for Multitenancy

- Within container databases it is possible to use Resource Manager to control resource consumption of individual tenants (PDBs)
- CDB Level – between PDBs
  - Define resource allocation to specific PDBs
  - Limit resource utilizations for specific PDBs
  - CDB DBA can give more resources to more important PDBs
  - System resource shares and limits can be configured
- PDB Level – within PDBs
  - Define resource allocations within specific PDBs
- No Resource Manager for CDB$ROOT

Example for CDB Resource Plan

	Shares	Utilization Limit %	Parallel Server Limit %
Default per PDB	1	100	100
PDB1	1	50	20
PDB2	2	75	20
PDB3	3	100	100

- PDB1 gets guaranteed 1 share of total 6, so 17% of system resources (CPU, Exadata I/O Bandwidth, queued parallel statements) – PDB2 33% – PDB3 50%
- PDB1 can utilize max. 50% of system resources
- PDB1 can utilize max. 20% of parallel server processes specified by init parameter parallel_servers_target
- One row in the table is defined as a CDB plan directive (= default)

CDB Resource Plan (1)

- Create a pending area and a new CDB plan

```
SQL> execute dbms_resource_manager.create_pending_area;
SQL> execute dbms_resource_manager.create_cdb_plan(
   2 plan                   => 'MY_PLAN',
   3 comment                => 'My CDB plan'
   4 );
```

- Create a CDB plan directive for each PDB

```
SQL> execute dbms_resource_manager.create_cdb_plan_directive(
   2 plan                  => 'MY_PLAN',
   3 pluggable_database    => 'PDB1',
   4 comment               => 'give 1 share',
   5 shares                => 1,
   6 utilization_limit     => 50,
   7 parallel_server_limit => 20
   8 );
```

CDB Resource Plan (2)

- Update the Default Directive

```
SQL> execute dbms_resource_manager.update_cdb_default_directive(
  plan                        => 'MY_PLAN',
  new_shares                  => 1,
  new_utilization_limit       => 100,
  new_parallel_server_limit => 100 );
```

- Update the directive for the automated maintenance tasks in CDB$ROOT

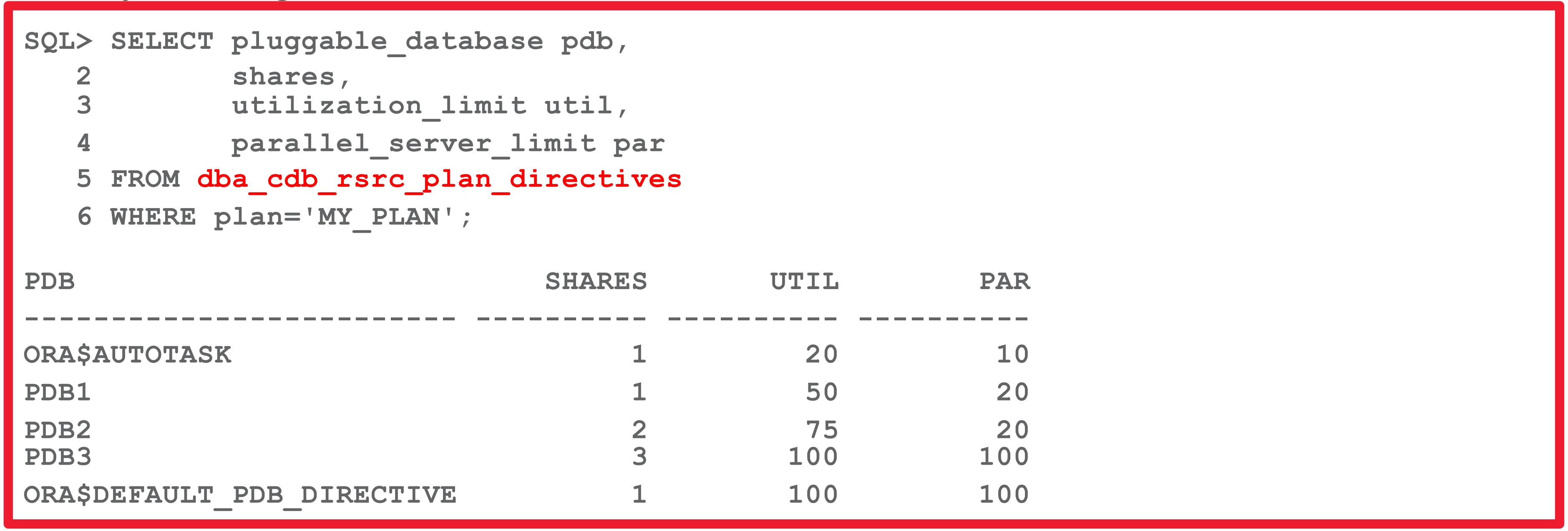
```
SQL> execute dbms_resource_manager.update_cdb_autotask_directive(
   2 plan                        => 'MY_PLAN',
   3 new_shares                  => 1,
   4 new_utilization_limit       => 20,
   5 new_parallel_server_limit => 10);
```

- Validate and submit

```
SQL> execute dbms_resource_manager.validate_pending_area;
SQL> execute dbms_resource_manager.submit_pending_area;
```

Query CDB Resource Plans

- Query the assigned directives



```
SQL> SELECT pluggable_database pdb,
  2         shares,
  3         utilization_limit util,
  4         parallel_server_limit par
  5 FROM dba_cdb_rsrc_plan_directives
  6 WHERE plan='MY_PLAN';

PDB                          SHARES       UTIL        PAR
------------------------- ---------- ---------- ----------
ORA$AUTOTASK                      1         20         10
PDB1                              1         50         20
PDB2                              2         75         20
PDB3                              3        100        100
ORA$DEFAULT_PDB_DIRECTIVE         1        100        100
```

Activate a CDB Resource Plan

- Activate instantly

```
SQL> ALTER SYSTEM SET resource_manager_plan = MY_PLAN
```

- Regularly via Scheduler Window

```
SQL> execute dbms_scheduler.create_window(
   2 window_name     => 'MY_WINDOW',
   3 resource_plan   => 'MY_PLAN',
   4 start_date      => to_timestamp_tz('24.07.2013 20:00:00',
   5                                'DD.MM.YYYY HH24:MI:SS'),
   6 repeat_interval => 'freq=daily',
   7 duration        => INTERVAL '2' HOUR
   8 );
```

Combined CDB and PDB Resource Plan Usage

- CDB resource plans and PDB resource plans can be combined:

CDB

PDB	Shares	Util Limit
PDB1	1	20 %
PDB2	2	100 %
PDB3	3	50 %

PDB3

Cons. Group	Shares	Util Limit
OLTP	1	30 %
BATCH	2	20 %
BOARD	2	50 %

- How much CPU resources gets BOARD in PDB3 ?
  - Guaranteed is: 3/6 * 2/5 = 6/30 = 20%
  - Limited to: 50 % * 50 % = 25 %

Performance Profiles

12.2

- A performance profile is a collection of resource manager settings for PDBs
- E.g. SLA level (gold, silver, bronze)
- Easy way to modify the resource limits for a group of PDBs

```
DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
  plan => 'newcdb_plan',
  profile => 'gold',
  shares => 3,
  utilization_limit => 100,
  parallel_server_limit => 100);
```

- Activation on PDB level (static parameter DB_PERFORMANCE_PROFILE)

```
ALTER SYSTEM SET DB_PERFORMANCE_PROFILE=gold SCOPE=SPFILE;
```

Memory Management for PDBs (1)

12.2

The following memory related parameters can be set on PDB level:

Parameter	Meaning
DB_CACHE_SIZE	Minimum guaranteed buffer cache for the PDB
SHARED_POOL_SIZE	Minimum guaranteed shared pool for the PDB
PGA_AGGREGATE_LIMIT	Maximum PGA size for the PDB
SGA_MIN_SIZE	Minimum SGA size for the PDB
SGA_TARGET	Maximum SGA size for the PDB
INMEMORY_SIZE	Maximum Size of the In-Memory-Column-Store

Memory Management for PDBs (2)

12.2

- Requirements
  - Parameter NONCDB_COMPATIBLE=FALSE in CDB$ROOT
  - MEMORY_TARGET not set in CDB$ROOT
- Restrictions for SGA related parameters (if SGA_TARGET=0)
  - Sum of all values for SGA (DB_CACHE_SIZE, SGA_MIN_SIZE, SHARED_POOL_SIZE) for all PDBS must no be higher than 50% of the corresponding value for CDB$ROOT
- Restrictions (PGA)
  - PGA_AGGREGATE_LIMIT: not more than PGA_AGGREGATE_LIMIT in CDB$ROOT
- INMEMORY_SIZE
  - Over-Provisioning allowed

I/O Rate Limits for PDBs (1)

12.2

- New parameters on container level (CDB$ROOT or PDB):

```
ALTER SYSTEM SET MAX_IOPS = 1000 SCOPE = BOTH
```

```
ALTER SYSTEM SET MAX_MBPS = 5 SCOPE = BOTH
```

- To disable a limit, set the parameter to 0 (Default)
- Event "resmgr: I/O rate limit“ (V$SYSTEM_EVENT, V$SESSION_EVENT) is raised when the limit is hit
- Values set in CDB$ROOT are the default for PDBs
- Not supported on Exadata
- DBWR-I/O, Controlfile-I/O and Password file are exempted
- Parameters are not supported on Non-CDBs
("ORA-56739: cannot modify max_iops or max_mbps parameter")

I/O Rate Limits for PDBs (2) - Example

12.2

```
SQL> REM No limits set
SQL> SELECT * FROM DBA_TAB_COLUMNS;
[..]
.. Takes 11 seconds
SQL> alter system set max_iops=80 scope=both;
SQL> alter system set max_mbps=8 scope=both;
SQL> alter system flush buffer_cache;
SQL> alter system flush shared_pool;
SQL> SELECT * FROM DBA_TAB_COLUMNS;
[..]
.. Takes 23 seconds
SQL> select con_id,event,time_waited from v$session_event
  2 where event='resmgr: I/O rate limit';
CON_ID EVENT                                      TIME_WAITED
------ ---------------------------------------- -----------
     0 resmgr: I/O rate limit                            95
     1 resmgr: I/O rate limit                           215
```

Database Maintenance Jobs (1)

- Since Oracle 10g, Oracle runs maintenance jobs (e.g. Statistics Gathering, Tuning Advisor etc.) in a defined maintenance window
- Default window
  - Weekdays: 10 PM – 2 AM
  - Weekend: 6 AM – 2 AM
- In a Multitenant database this window is the default window for all PDBs!
  - ➔ in a CDB with a huge number of PDBs this can lead to high load during the maintenance window

Database Maintenance Jobs (2)

- **Workarounds:**
  - Define different maintenance windows for the PDBs (depending on SLA, ETL jobs etc.)
  - Change the number of concurrent maintenance jobs (Default value is 2)

```
ALTER SYSTEM SET AUTOTASK_MAX_ACTIVE_PDBS = 4 SCOPE = BOTH
```

  - Deactivate automatic maintenance jobs on PDB level (Default: TRUE)

```
ALTER SYSTEM SET ENABLE_AUTOMATIC_MAINTENANCE_PDB = FALSE SCOPE = BOTH
```

Other resource-relevant Parameters on PDB-level

12.2

Parameter	Meaning
MAX_DATAPUMP_JOBS_PER_PDB	Maximum number of concurrent DataPump Jobs in the PDB (Value in CDB$ROOT is the default for the PDBs)
SESSIONS	Maximum of concurrent sessions in the PDB (available since 12.1)

- Not limiting the number of sessions in a PDB can lead to the problem that one PDB takes all available sessions (instance parameter SESSIONS)
  - ➔ no logins to the other PDBs and to CDB$ROOT possible

Resource Monitoring - V$RSRCPDBMETRIC (1)

12.2

- The view **V$RSRCPDBMETRIC** contains resource usage data for the last minute
- **V$RSRCPDBMETRIC_HISTORY** (with the same structure) contains date for the last hour
- The AWR-View **DBA_HIST_RSRC_PDB_METRIC** (Diagnostic Pack required!) contains persistent snapshots of the view V$RSRCPDBMETRIC
- Can be used for PDB charging (e.g. by used memory, I/O etc.)

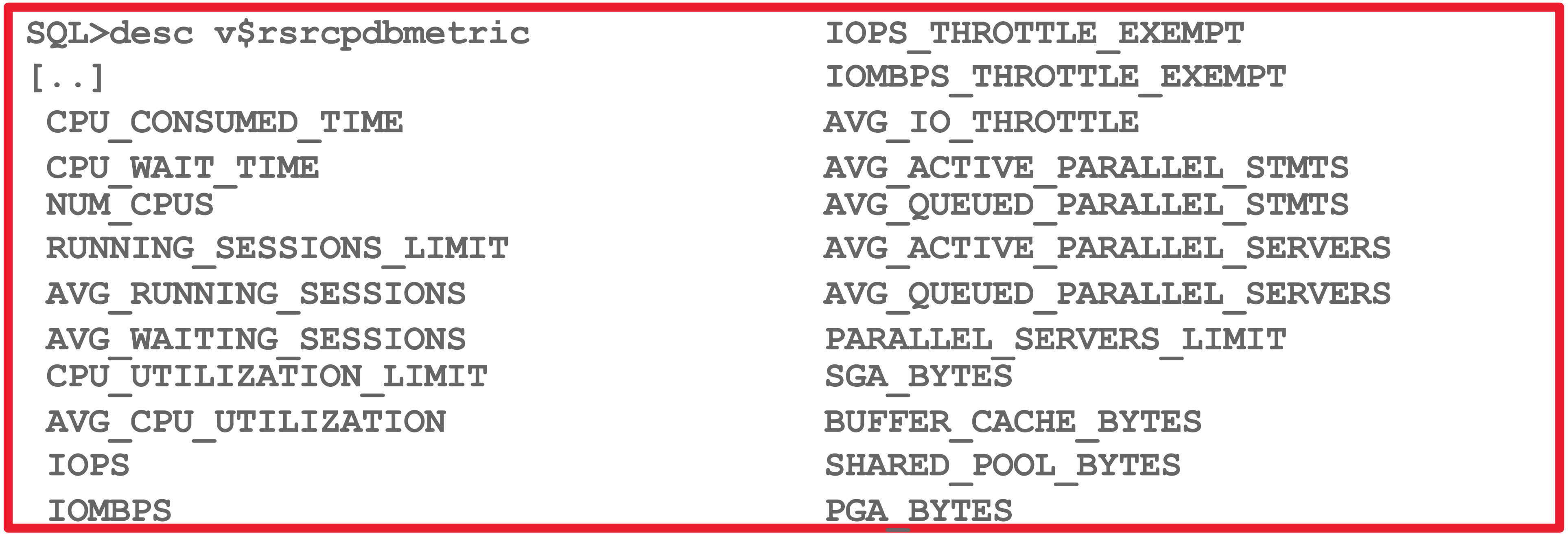
```
SELECT r.CON_ID, p.PDB_NAME, r.IOPS,r.SGA_BYTES, r.SHARED_POOL_BYTES
FROM V$RSRCPDBMETRIC r, CDB_PDBS p
WHERE r.CON_ID = p.CON_ID;

CON_ID PDB_NAME   IOPS       SGA_BYTES SHARED_POOL_BYTES
------ --------- ---------- ---------- -----------------
3      PDB01     .440423759   86405592           7401944
4      PDB02     .464230449   51885608          12138024
```

Resource Monitoring - V$RSRCPDBMETRIC (2)

12.2

- What is logged in V$RSRCPDBMETRIC?



```
SQL>desc v$rsrcpdbmetric                 IOPS_THROTTLE_EXEMPT
[..]                                     IOMBPS_THROTTLE_EXEMPT
 CPU_CONSUMED_TIME                       AVG_IO_THROTTLE
 CPU_WAIT_TIME                           AVG_ACTIVE_PARALLEL_STMTS
 NUM_CPUS                                AVG_QUEUED_PARALLEL_STMTS
 RUNNING_SESSIONS_LIMIT                  AVG_ACTIVE_PARALLEL_SERVERS
 AVG_RUNNING_SESSIONS                    AVG_QUEUED_PARALLEL_SERVERS
 AVG_WAITING_SESSIONS                    PARALLEL_SERVERS_LIMIT
 CPU_UTILIZATION_LIMIT                   SGA_BYTES
 AVG_CPU_UTILIZATION                     BUFFER_CACHE_BYTES
 IOPS                                    SHARED_POOL_BYTES
 IOMBPS                                  PGA_BYTES
```

Lockdown Profiles

PDB Lockdown Profiles (1) – Use cases

12.2

- Enforce the separation of duties in a Container Database:
  - CDB administrator: "infrastructure administrator"
  - PDB administrator: "application DBA with restricted privileges
- Control Feature Usage on PDB level
  - Create a CDB with all options
  - Disable options on PDB level which were not ordered by the DBaaS-customer
- Disable access to OS and network resources
  - E.g. use of packages like UTL_FILE, UTL_MAIL, UTL_HTTP, ..

PDB Lockdown Profiles (2)

12.2

- Restrict feature usage on PDB level
- Areas
  - Network access
  - Common user or object access
  - Administrative features
  - XML database access
  - Database options (e.g. Partitioning)
- Default lockdown profiles (empty, i.e. no limits defined)
  - SAAS
  - PUBLIC_DBAAS
  - PRIVATE_DBAAS

PDB Lockdown Profiles (3)

12.2

- Create a lockdown profile in CDB$ROOT

```
CREATE LOCKDOWN PROFILE demo_lckdprf;
ALTER LOCKDOWN PROFILE demo_lckdprf DISABLE STATEMENT = ('ALTER SYSTEM');
ALTER LOCKDOWN PROFILE demo_lckdprf ENABLE STATEMENT = ('ALTER SYSTEM') CLAUSE
= ('flush shared_pool');
ALTER LOCKDOWN PROFILE demo_lckdprf DISABLE FEATURE = ('NETWORK_ACCESS');
ALTER LOCKDOWN PROFILE demo_lckdprf DISABLE OPTION = ('Partitioning');
```

PDB Lockdown Profiles (4)

12.2

- Set the lockdown profile on PDB level (static parameter PDB_LOCKDOWN)
  - Bounce the PDB to activate the lockdown profile

```
ALTER SESSION SET CONTAINER=PDB1;
ALTER SYSTEM SET PDB_LOCKDOWN = demo_lckdprf SCOPE = SPFILE;
ALTER PLUGGABLE DATABASE PDB1 CLOSE;
ALTER PLUGGABLE DATABASE PDB1 OPEN;
```

- **Recommendation:** Disable „ALTER SYSTEM“ via lockdown profile
  - However, this makes it difficult to disable the lockdown profile ☹
- When PDB_LOCKDOWN is set in CDB$ROOT, this will be the default lockdown profile for all PDBs
  - A "local" lockdown profile set in a PDB overrides the global profile

PDB Lockdown Profiles (5)

12.2

- What happens in the PDB?

```
SQL> ALTER SYSTEM FLUSH BUFFER_CACHE;
Error at line 1:
ORA-01031: insufficient privileges

SQL> ALTER SYSTEM FLUSH SHARED_POOL;
System altered.

SQL> CREATE TABLE .. PARTITION BY ..
ERROR at line 1:
ORA-00439: feature not enabled: Partitioning
```

PDB Lockdown Profiles (6)

12.2

- Options, which can be disabled:
  - Database Queueing
  - Partitioning
- Statements, which can be disabled:
  - ALTER DATABASE
  - ALTER PLUGGABLE DATABASE
  - ALTER SESSION
  - ALTER SYSTEM
- For statements, specific clauses can be enabled/disabled.

```
ALTER LOCKDOWN PROFILE demo_prf DISABLE STATEMENT = ('ALTER SYSTEM') CLAUSE =
('SUSPEND', 'RESUME');
```

trivadis
makes IT easier.

PDB Lockdown Profiles (7)

12.2

- Features (excerpt)
  - AWR Access
  - Network Access (UTL_TCP, UTL_HTTP, UTL_MAIL, UTL_SNMP, UTL_INADDR and DBMS_DEBUG_JDWP, XDB Protocols
  - JAVA
  - OS Access (UTL_FILE or DBMS_FILE_TRANSFER)

- Please see "Oracle 12.2 SQL Reference" for a complete list

Miscellaneous

OMF/ASM – CREATE_FILE_DEST

12.2

- CREATE_FILE_DEST specifies the default location for OMF data files in the pluggable database (PDB). When not set, the PDB inherits the value from the root container.

```
CREATE_FILE_DEST = +U01
```

- CREATE_FILE_DEST can be specified when creating a PDB

```
SQL> create pluggable database DEMOPDB2
  2 admin user admin identified by manager role=(DBA) create_file_dest='+DATA';
```

- Trying to create a file outside CREATE_DEST results in an error

```
SQL> create tablespace DEMO datafile '/u01/oradata/TVDCDB2/DEMOPDB2/demo.dbf';
*
ERROR at line 1:
ORA-65250: invalid path specified for file - /u01/oradata/TVDCDB2/DEMOPDB2/demo.dbf
```

PDB OS User

12.2

- Ability to set the identity of the operating System user for PDBs
- Define OS user by setting the parameter **PDB_OS_CREDENTIAL** in the PDB
- Create a credential with **DBMS_CREDENTIAL.CREATE_CREDENTIAL**

```
BEGIN DBMS_CREDENTIAL.CREATE_CREDENTIAL (
  credential_name => 'CDB1_PDB1_OS_USER', username => 'os_admin',
  password => 'password');
END;
```

- Limited OS interactions
  - External jobs that do not already have an operating system credential specified
  - External table per-processors
  - PL/SQL library executions

Summary

Summary

- Resource management and resource monitoring is a must for DBaaS
- Oracle 12.2 can manage all kinds of resources (CPU, I/O, memory)

- Lockdown Profiles allow fine granular access to features and administrative commands

- Be careful when defining resource limits!

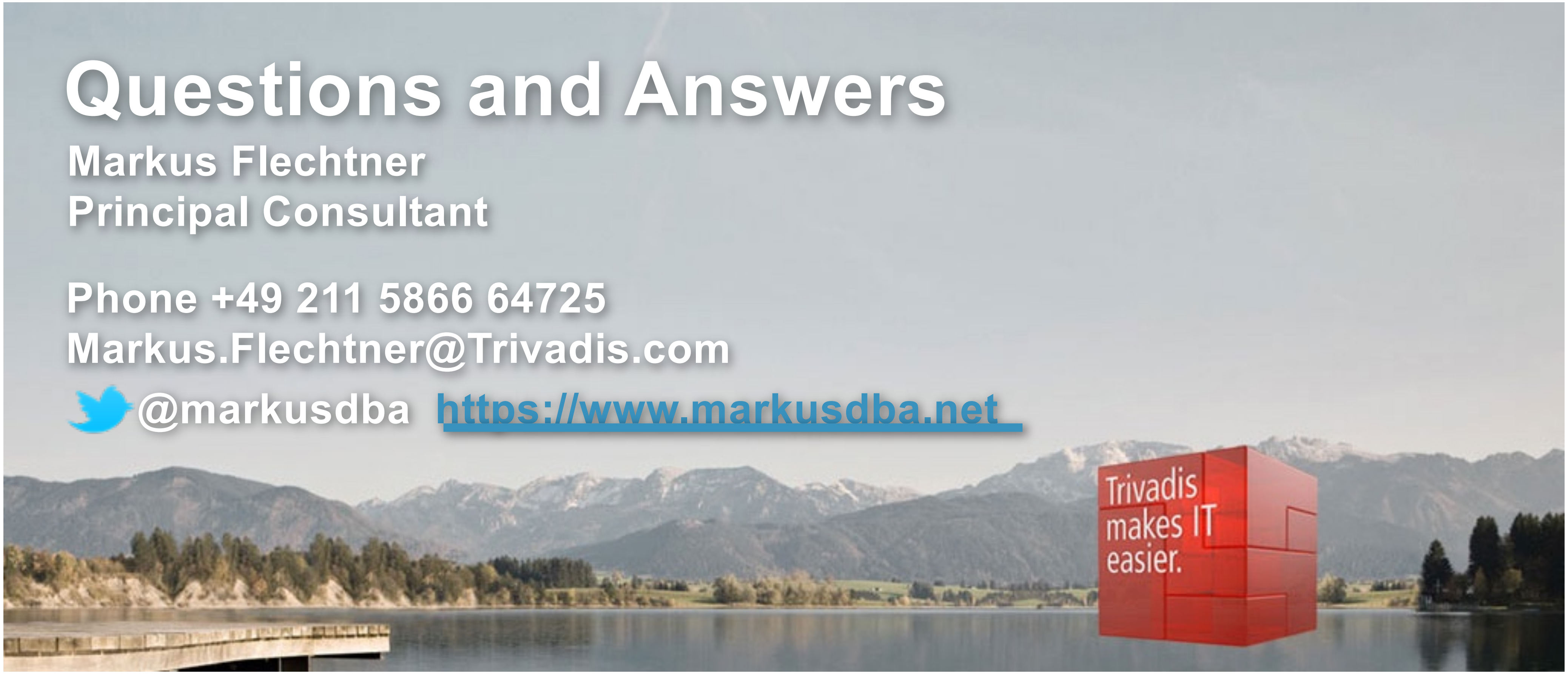
Further Information

- MOS Note 2171135.1: Managing OS Resources Among PDBs Using PDB Performance Profiles - 12.2 New Feature
- MOS-Note 2170772.1: How to Control and Monitor the Memory Usage (Both SGA and PGA) Among the PDBs in Multitenant Database- 12.2 New Feature
- MOS-Note 2326708.1: How to Provision PDBs, based on CPU_COUNT
- Whitepaper PDB Isolation: http://www.oracle.com/technetwork/database/multitenant/learn-more/isolation-wp-12c-3614475.pdf

Questions and Answers

Markus Flechtner
Principal Consultant

Phone +49 211 5866 64725
Markus.Flechtner@Trivadis.com
@markusdba https://www.markusdba.net



Download the slides from https://www.slideshare.net/markusflechtner

Please don‘t forget the session evaluation – Thank you!