

DOAG 2014



Database Resource Manager

Ulrike Schwinn

 Follow @uschwinn

Business Unit Database

Oracle Deutschland B.V. & Co KG

Blog:http://blogs.oracle.com/dbacommunity_deutsch

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. |

Agenda

- 1 Einführung – ein Kurzprofil
- 2 Setup – schnell mal Resource Manager aufsetzen!
- 3 Anwendungsfälle
- 4 Fazit, Ausblick, weitere Informationen

Database Resource Manager - Kurzprofil

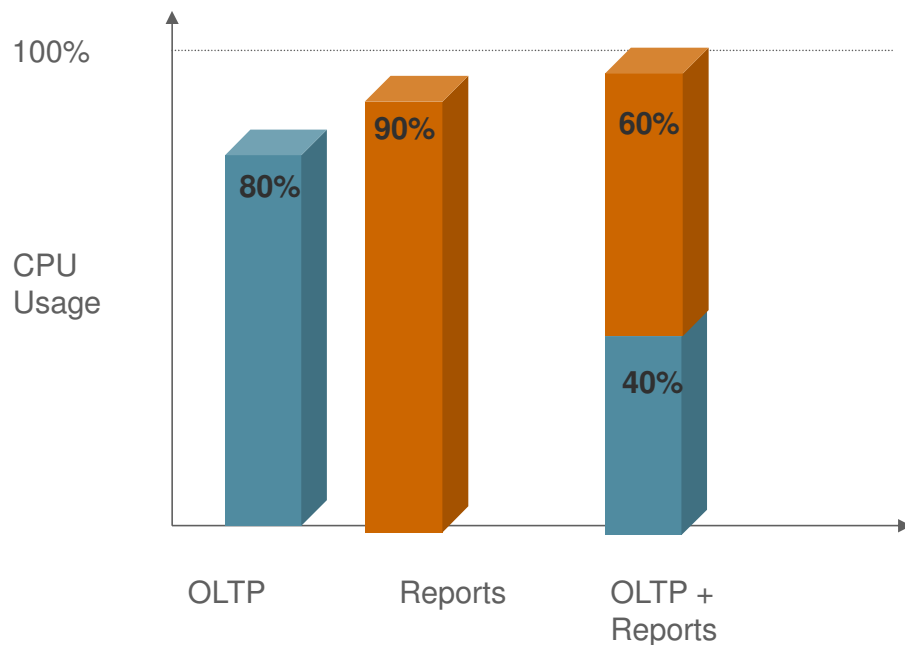


- Verfügbar in der neusten Datenbank Version 12.1.0.2
- Lizenzierung: Enterprise Edition
- Aktivierung: automatisch im Maintenance Window; sonst manuell
- Nutzung über PL/SQL Interface und über Cloud Control bzw. SQL Developer
- Integration in Multitenant Umgebungen, Scheduler, SQL Monitoring, Parallel Query Interface usw.
- Exadata Only Feature
- Voraussetzung für Instance Caging

Ressource Management mit Database Resource Manager

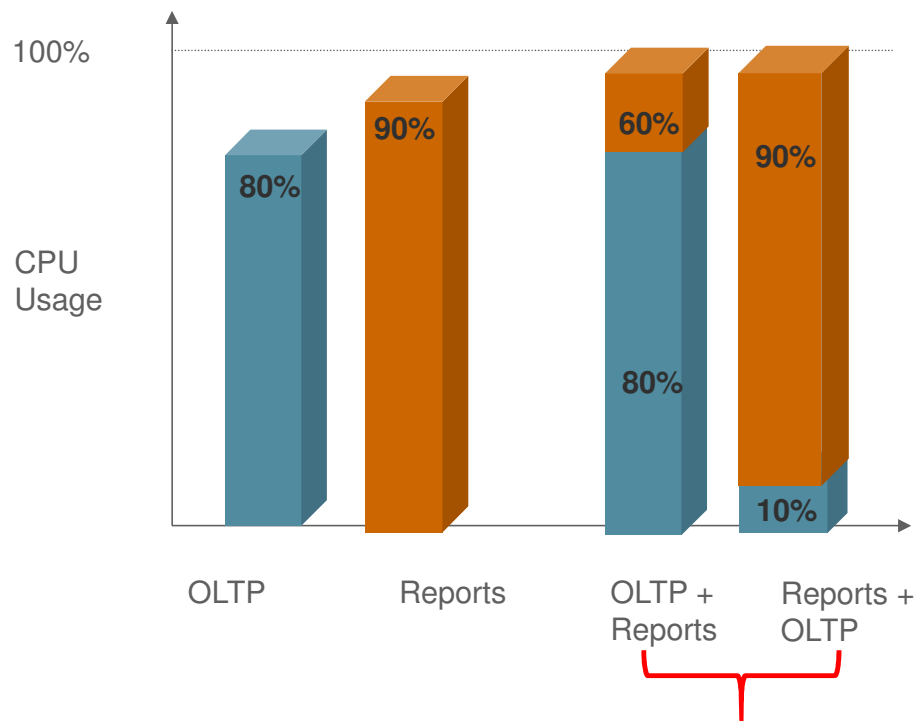
- Kann ein Minimum an Ressourcen garantieren bzw. die Verwendung von Ressource Verwendung sogar begrenzen
- Konsolidierung **mehrerer Datenbanken** auf einem Server
- Verwaltung **mehrerer verschiedener Workloads** in einer Datenbank
- Art der Ressourcen
 - CPU
 - Disk I/O
 - Runaway Queries
 - Parallel Execution
 - Idle Time

Beispiel: Workload Contention



- Falls der Datenbank Server nicht über ausreichende CPU Ressourcen verfügt, kann es zu Contention kommen.
=> Alle Workloads könnten unter Performance Engpässen leiden.

Lösung: Workload Contention



- Mit dem Resource Manager lässt sich die CPU Allokation kontrollieren.

Resource Manager im Enterprise Manager

Enterprise Targets Favorites History Search Target Name

us_sccloud036_noncdb Oracle Database Performance Availability Security Schema Administration Logged in as system | sccloud036.de.oracle.com

Resource Plans > Edit Resource Plan: DEFAULT_MAINTENANCE_PLAN Logged in as SYSTEM
View Resource Plan: DEFAULT_MAINTENANCE_PLAN

Actions Create Like Go Edit Return

General Parallelism Runaway Query Idle Time

A Resource Plan contains directives that specify how resources are allocated to Consumer Groups. For each Consumer Group, a directive specifies the amount of CPU resources are allocated. It also specifies limits, such as the maximum degree of parallelism, execution time, and amount of I/O, that each session in the Consumer Group can consume. You can enable a Resource Plan manually or automatically, using Scheduler Windows. The maximum number of consumer groups in a plan can not exceed 28.

Plan DEFAULT_MAINTENANCE_PLAN
Description Default plan for maintenance windows that prioritizes SYS_GROUP operations, leaving 5% for automated maintenance operations and 20% for other tasks.

Activate this plan
 Automatic Plan Switching Enabled

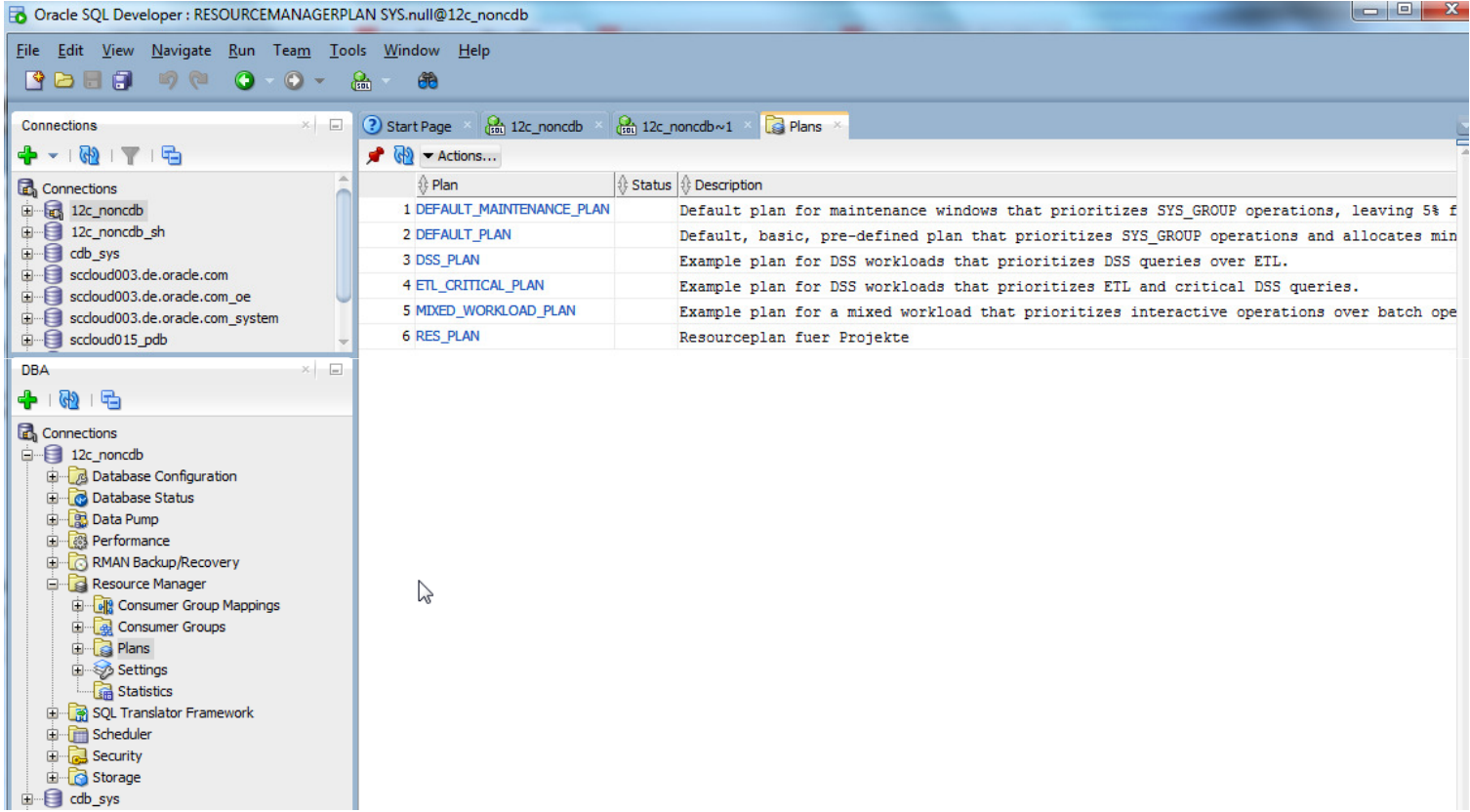
Resource Allocations

Group/Subplan	Shares	Percentage	Utilization Limit %
ORA\$AUTOTASK	5	5	90
SYS_GROUP	75	75	
OTHER_GROUPS	20	20	

Total Shares: 100

Actions Create Like Go Edit Return

Resource Manager im SQL Developer



Agenda

- 1 Einführung – ein Kurzprofil
- 2 **Setup – schnell mal Resource Manager aufsetzen!**
- 3 Anwendungsfälle
- 4 Fazit, Ausblick, weitere Informationen

Setup – was muss man wissen

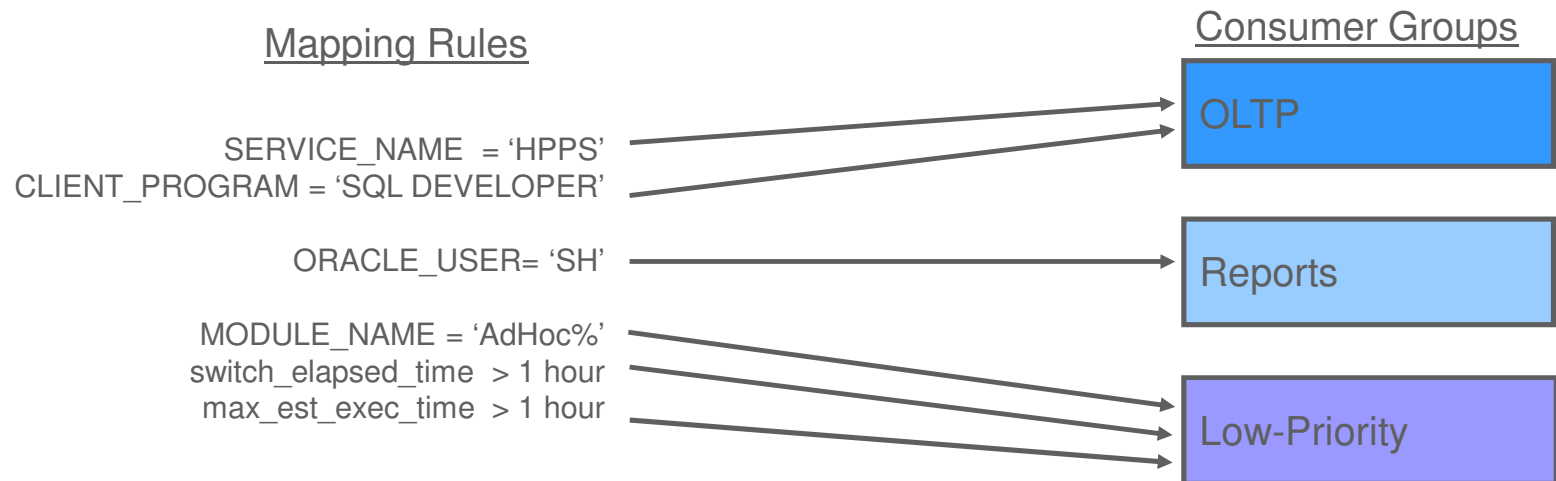
- Vertrautmachen mit einigen neuen Begriffen wie Consumer Group etc.
- Nutzung von bewährten Techniken zur Einbindung der eigenen Umgebung wie Net Service, Package DBMS_APPLICATION_INFO etc.
- Am Besten Verwendung von existierenden Skripten
 - <http://joelkallman.blogspot.co.uk/2009/08/oracle-database-resource-manager-and.html> (APEX Blog)
 - https://blogs.oracle.com/dbacommunity_deutsch/ (deutsche DBA Community Blog)
- Bevorzugen von einfachen Setup Modellen
- Immer Einplanen von Tests

3 Schritte zum Setup

1. Gruppiere Sessions mit ähnlichen Anforderungsprofilen in **Consumer Groups**
2. Alloziere Ressourcen (als **Direktive**) für die Consumer Gruppen über die Definition eines **Resource Plans**
3. Schalte Resource Plan ein

Schritt 1: Consumer Group

- Eine Consumer Group für jedes Workloadprofil, z.B.
 - **OLTP**, **Reports** und **Low-Priority** Consumer Group
- Session Mapping auf Consumer Group über definierte Regeln



Schritt 1: Consumer Group

- Vordefinierte Gruppen: SYS_GROUP, OTHER_GROUPS, BATCH_GROUP ...
- Session Attribute für Zuweisung auf Consumer Group sind
 - ORACLE_USER
 - SERVICE_NAME
 - CLIENT_OS_USER, CLIENT_PROGRAM, CLIENT_MACHINE
 - MODULE_NAME, MODULE_NAME_ACTION, SERVICE_MODULE, SERVICE_MODULE_ACTION
 - ORACLE_FUNCTION
- Priorisierung möglich
- Automatischer/manueller Wechsel zu anderer Gruppe möglich

```

DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
    (consumer_group =>'PQ_MEDIUM', comment =>'Kommentar');
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
    (attribute      => dbms_resource_manager.oracle_user,
     value          => 'SCOTT',
     consumer_group => 'PQ_MEDIUM');
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP    -- Nicht vergessen!!!!
    (grantee_name   => 'SCOTT',
     consumer_group => 'PQ_MEDIUM',
     grant_option   => true);

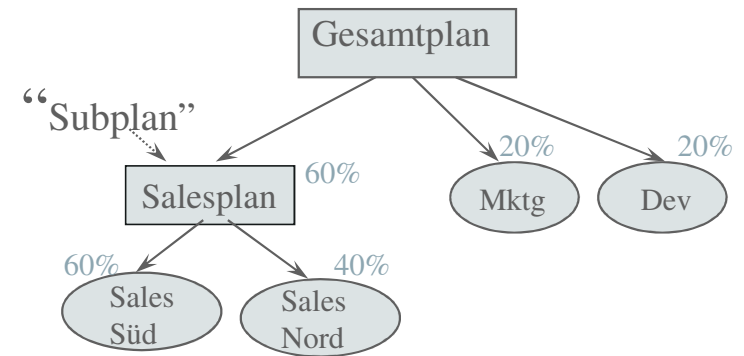
```

```

SQL> SELECT initial_rsrc_consumer_group FROM user_users;
INITIAL_RSRC_CONSUMER_GROUP
-----
PQ_MEDIUM

```

Schritt 2: Resource Plan



- Vordefinierte Pläne: DEFAULT_MAINTENANCE_PLAN, DEFAULT_PLAN, INTERNAL_PLAN, BATCH_PLAN ...
- Möglichkeiten: **Single Level** oder **Multi Level** Plan
- Die CPU Einteilung erfolgt über **prozentuale Gewichtung** (mgmt_px) oder über sogenannte **Shares**
- **Limitierung** der CPU Ressource seit 11.2 mit MAX_UTILIZATION_LIMIT bzw. UTILIZATION_LIMIT (12c)
- Immer SYS_GROUP oder OTHER_GROUPS berücksichtigen!

Schritt 2: Resource Plan

Resource Plans > View Resource Plan: RES_PLAN
View Resource Plan: RES_PLAN

Logged in as SYSTEM

Actions

General Parallelism Runaway Query Idle Time

A Resource Plan contains directives that specify how resources are allocated to Consumer Groups. For each Consumer Group, a directive specifies the amount of CPU resources are allocated. It also specifies limits, such as the maximum degree of parallelism, execution time, and amount of I/O, that each session in the Consumer Group can consume. You can enable a Resource Plan manually or automatically, using Scheduler Windows. The maximum number of consumer groups in a plan can not exceed 28.

Plan RES_PLAN
Description Resourceplan fuer Projekte
 Activate this plan
 Automatic Plan Switching Enabled

Resource Allocations

Mode: Share Advanced

Group/Subplan	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Utilization Limit %
BATCH_GROUP	0	25	0	0	0	0	0	0	35
DSS_GROUP	0	70	0	0	0	0	0	0	75
SYS_GROUP	90	0	0	0	0	0	0	0	95
OTHER_GROUPS	0	0	10	0	0	0	0	0	15


```

DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.CREATE_PLAN( plan    => 'RES_PLAN',
                                   comment => 'Kommentar');

DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(
  plan          => 'RES_PLAN',
  group_or_subplan => 'SYS_GROUP',
  comment       => 'Kommentar',
  mgmt_p1       => 90);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(
  plan          => 'PLAN1',
  group_or_subplan => 'OTHER_GROUPS',
  comment       => 'Kommentar',
  mgmt_p4       => 10,  -- nicht cpu_p4 verwenden
  parallel_degree_limit_p4 => 0,  -- gibt DOP an
  MAX_UTILIZATION_LIMIT    => 15); -- UTILIZATION_LIMIT in 12c
...
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();

```

Schritt 3: Resource Plan einschalten

- Es gilt immer ein Plan zu einer Zeit!
- Manuelle Aktivierung
 - Initialisierungsparameter **RESOURCE_MANAGER_PLAN** oder SWITCH_PLAN Prozedur

```
DBMS_RESOURCE_MANAGER.SWITCH_PLAN( plan_name => 'RES_PLAN', sid => '*' );
```

- Automatische Aktivierung
 - Nutzung von Job Scheduler Window, das den Resource Plan verwendet.

```
DBMS_SCHEDULER.SET_ATTRIBUTE ( name          => '"SYS"."FRIDAY_WINDOW"',  
                               attribute     => 'RESOURCE_PLAN',  
                               value         => 'RES_PLAN' );
```

Monitoring und Tuning

- Resource Manager konfigurieren ist ein iterativer Prozess
 1. Plan einschalten
 2. Monitoring der Anwendungsperformance in Verbindung mit den Resource Manager Metriken
 3. Anpassung der Ressourcen und erneutes Monitoring
- Falls
 - Viele Session Waits for CPU => Server ist überlastet
 - Performance einer Consumer Group nicht zufriedenstellend => Erhöhe Ressource Allokation für diese Consumer Group

Monitoring-Views

- V\$-Views

- V\$RSRC_PLAN aktiver Plan
- V\$RSRC_PLAN_HISTORY Plan Historie
- V\$RSRC_CONSUMER_GROUP Ressource pro Gruppe
- V\$RSRC_CONS_GROUP_HISTORY Historie der Statistiken
- V\$RSRC_SESSION_INFO Statistiken pro Sessions
- **V\$RSRCMGRMETRIC_HISTORY** Real-time Metriken für jede Consumer Group

- Einige DBA_Views für das Setup

- DBA_RSRC_CONSUMER_GROUP_PRIVS
- DBA_RSRC_GROUP_MAPPINGS

Monitoren mit Views: Ressourcen pro Gruppe

```
SQL> SELECT name, active_sessions,  
           consumed_cpu_time, cpu_waits, cpu_wait_time  
           FROM v$rsrc_consumer_group;
```

NAME	ACTIVE_SESSIONS	CONSUMED_CPU_TIME	CPU_WAITS	CPU_WAIT_TIME
GRUPPE_HIGH	2	26913930	3688	527354
GRUPPE_LOW	1	19442872	1976	1586315
OTHER_GROUPS	1	360010470	5598	22412666
SYS_GROUP	2	42449366	321	168697
_ORACLE_BACKGROUND_G ROUP_	21	0	0	0

Monitoren mit Views: Session genau

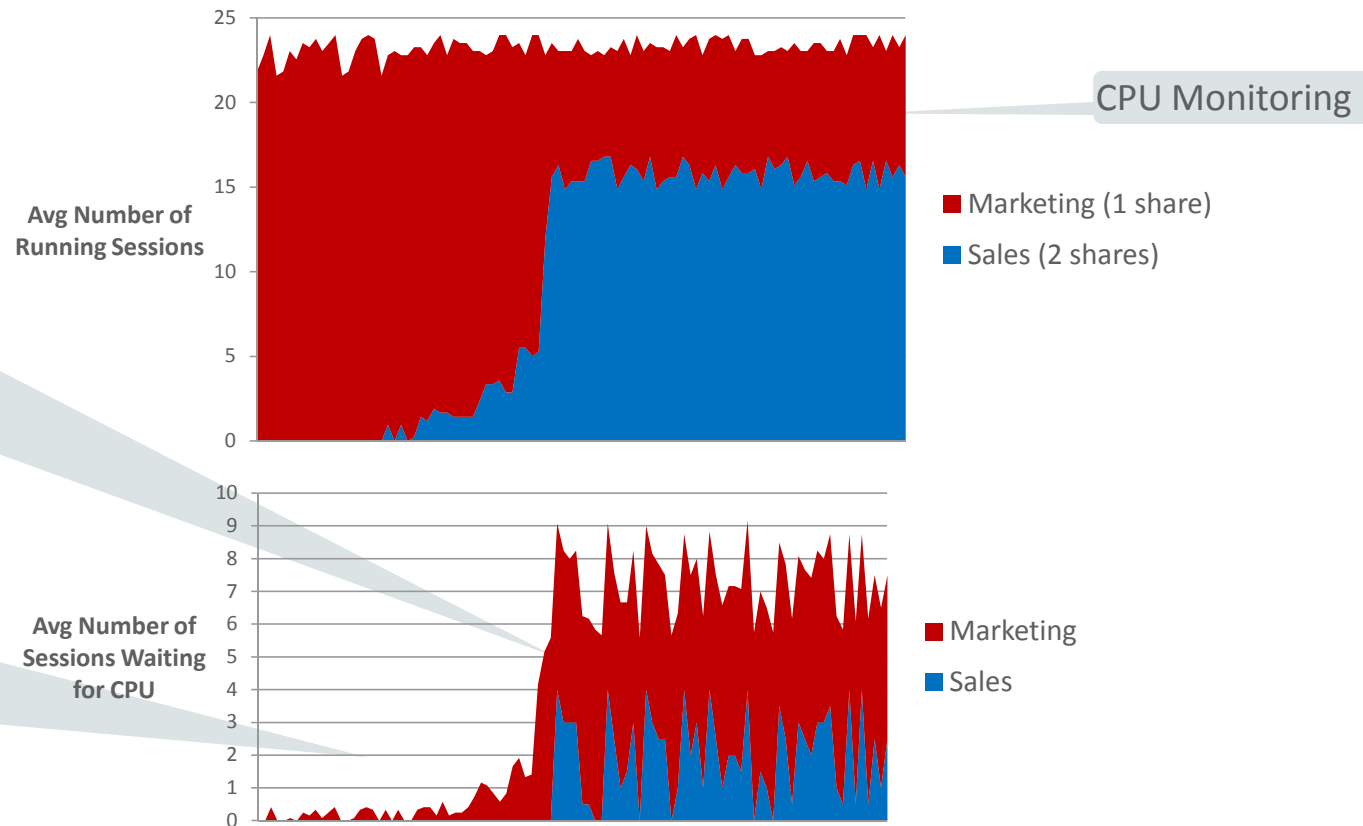
```
SQL> SELECT s.username username, s.osuser osuser, co.name gruppe,  
se.state, se.consumed_cpu_time cpu_time, se.cpu_wait_time  
FROM v$rsrc_session_info se, v$rsrc_consumer_group co,  
v$session s  
WHERE se.current_consumer_group_id = co.id and s.sid=se.sid ...
```

SYS	oracle	SYS_GROUP	RUNNING	61	252
SYS	oracle	SYS_GROUP	WAITING	2218	0
SYS	oracle	SYS_GROUP	WAITING	123	24
SCOTT	oracle	GRUPPE_HIGH	WAITING FOR CPU	222846	391045
SCOTT	oracle	GRUPPE_HIGH	RUNNING	217903	338729
SCOTT	oracle	GRUPPE_HIGH	WAITING FOR CPU	251877	393051
SH	oracle	GRUPPE_LOW	WAITING FOR CPU	83391	742496
SH	oracle	GRUPPE_LOW	WAITING FOR CPU	84375	752397
SH	oracle	GRUPPE_LOW	WAITING FOR CPU	113008	773229

Mit Cloud Control oder `v$rsrsrcmgrmetric_history`

Falls viele Sessions aus beliebigen Consumer Groups auf CPU warten => Server mit mehr Ressourcen ausstatten

Falls keine Session einer Consumer Group wartet => Resource Manager nicht aktiv für diese Sessions

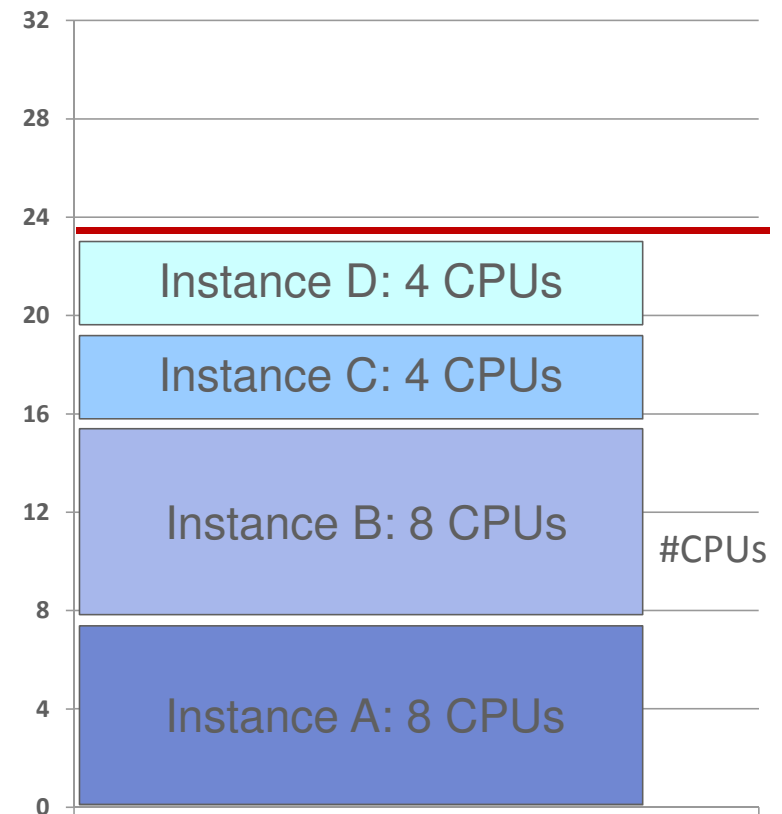


Agenda

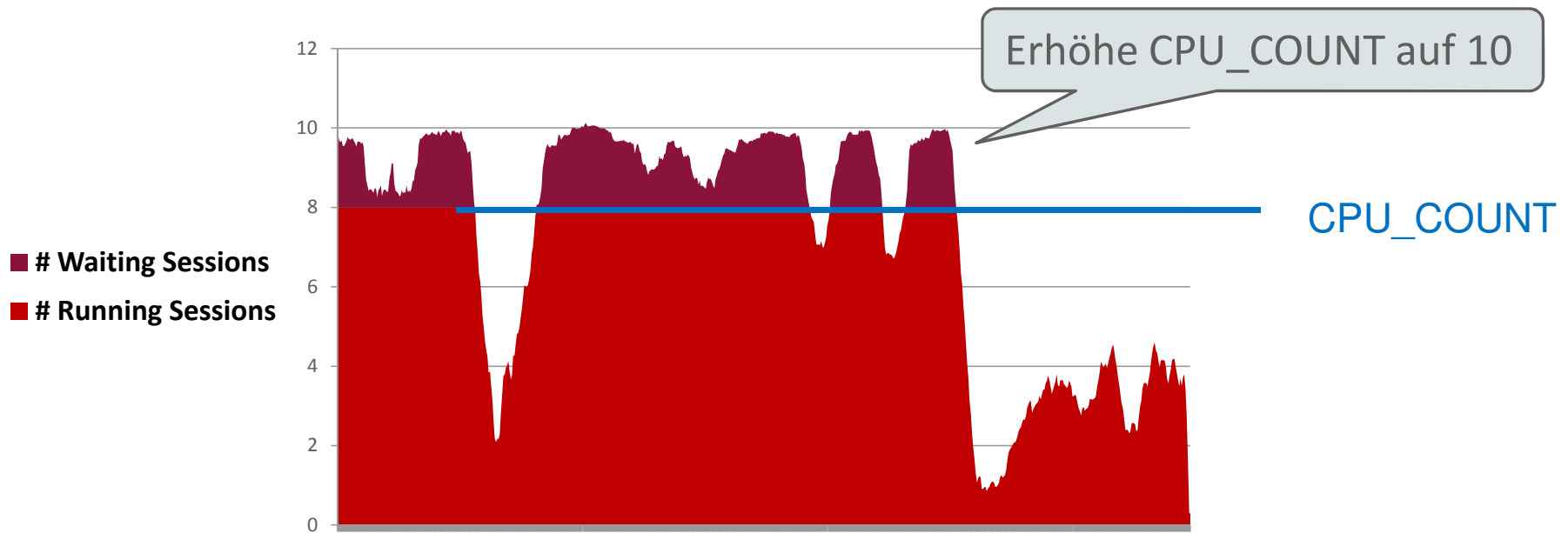
- 1 Einführung – ein Kurzprofil
- 2 Setup – schnell mal Resource Manager aufsetzen!
- 3 **Anwendungsfälle**
- 4 Fazit, Ausblick, weitere Informationen

Anwendungsfall 1: Instance Caging

- Dynamisches Setup in 2 Schritten
 1. CPU_COUNT = max #CPU Threads
 2. RESOURCE_MANAGER_PLAN = [Planname]
- Vermeide große Änderungen (CPU_COUNT)
- Einsatzmöglichkeiten
 - Aufteilung der vorhandenen CPUs auf Instanzen
 - “Over-Subscribe” der CPUs
- Support auf allen Plattformen
- Keine Zuweisung dedizierter CPUs



Instance Caging: CPU Nutzung und Tuning



Anwendungsfall 2: Langläufer/Runaway Queries

- Mögliche Ursache: nicht optimales SQL
- Kann Gesamtperformance beeinflussen
- Abhilfe mit Resource Manager: Grenzen und Aktionen definieren

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (  
  PLAN                => 'Runaway_plan',  
  GROUP_OR_SUBPLAN   => 'START_GROUP',  
  COMMENT             => 'Max CPU 10 minutes before switch',  
  MGMT_P1            => 75,  
  SWITCH_GROUP       => 'LOW_GROUP', -- neu in 12c: LOG_ONLY  
  SWITCH_TIME        => 600);
```

Beispiele für ein Setup

Consumer Group für spezielle Aufgaben:
Langläufer: **>=15 Minuten CPU**



Switch zu “Low Priority”
Consumer Group!

Consumer Group für Reporting
Langläufer: **60 Minuten Elapsed Time**



Abbrechen!

Consumer Group für Ad-Hoc Queries
Langläufer: **>=4h estimated execution time**



Nie Ausführen!

Grenzen und Aktionen

Definiere Thresholds:

- ✓ Estimated Execution Time
- ✓ Elapsed Time **Neu in 12c**
- ✓ CPU Time
- ✓ # I/Os
- ✓ # I/O Bytes
- ✓ # logical I/Os **Neu in 12c**

Mögliche Aktionen:

- ✓ Wechsel zu “lower-priority” Group
- ✓ Abbruch eines Calls
- ✓ Abbruch der Session
- ✓ Dokumentation (LOG_ONLY) in SQL Monitor **Neu in 12c**

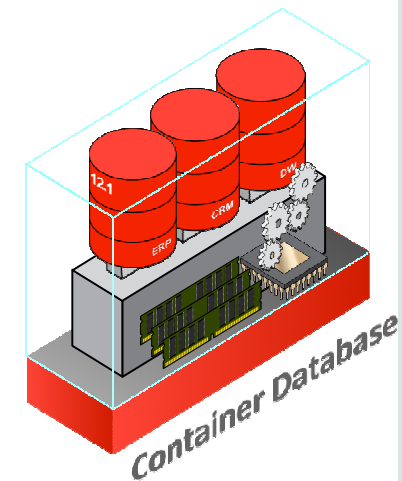
Monitoring in `v$sql_monitor` ab 12c

```
SQL> SELECT username, elapsed_time, plsql_exec_time, sql_text, cpu_time,  
           rm_last_action, rm_last_action_reason, rm_last_action_time, rm_consumer_group  
           FROM v$sql_monitor WHERE username is not null;
```

```
USERNAME                               ELAPSED_TIME PLSQL_EXEC_TIME  
-----  
SQL_TEXT  
-----  
CPU_TIME      RM_LAST_ACTION  
-----  
RM_LAST_ACTION_REASON                RM_LAST_A RM_CONSUMER_GROUP  
-----  
SH                               378358      0  
select /*+ use_nl(c) parallel ordered*/ count(*) from sh.sales s,sh.customers c where  
c.cust_id=s.cust_id and cust_first_name='Dina'  
  
10998      SWITCH TO OTHER_GROUPS  
  
SWITCH_CPU_TIME                    19-FEB-14 OTHER_GROUPS
```

Anwendungsfall 3: Multitenant Architektur

- Neues Konzept der CDBs und PDBs und gleichzeitige Einführung von PDB und CDB Resource Plänen
- **CDB Resource Plan** enthält Direktiven für die PDBs
 - Mögliche Ressourcen: CPU und Parallel Servers (PX)



CDB Resource Plan			
Pluggable Database	Shares	Guaranteed CPU	Maximum CPU
Sales	2	$2/4 = 50\%$	100%
Marketing	1	$1/4 = 25\%$	100%
Support	1	$1/4 = 25\%$	100%

CDB Resource Plan

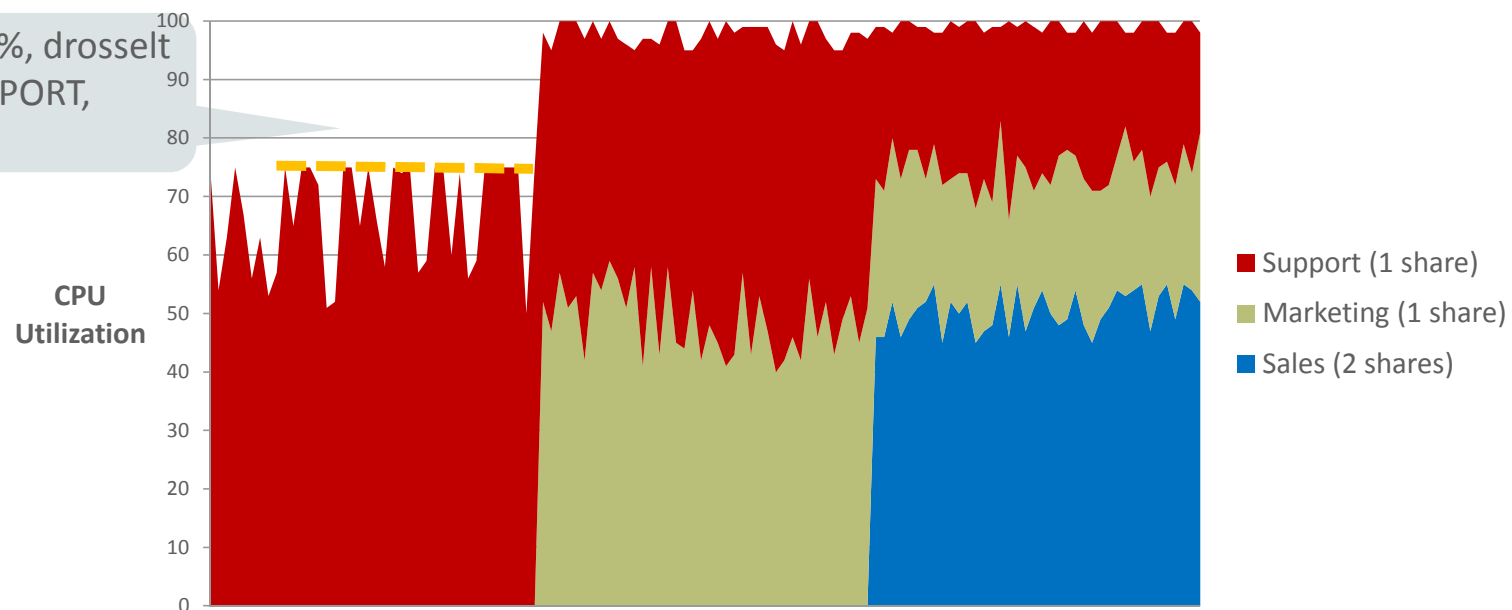
“Utilization Limits” für CPU Begrenzung einer PDB

CDB Resource Plan				
Pluggable Database	Shares	Utilization Limit	Guaranteed CPU	Maximum CPU
(Default directive)	1	75%		
Sales	2		2/5 = 40%	100%
Marketing	default (1)	default (75%)	1/5 = 20%	75%
Support	default (1)	default (75%)	1/5 = 20%	75%
ERP	default (1)	default (75%)	1/5 = 20%	75%

Eine Default Direktive für neue PDBs

CDB Resource Plan

“Utilization Limit” von 75%, drosselt die Consumer Group SUPPORT, obwohl CPU verfügbar ist



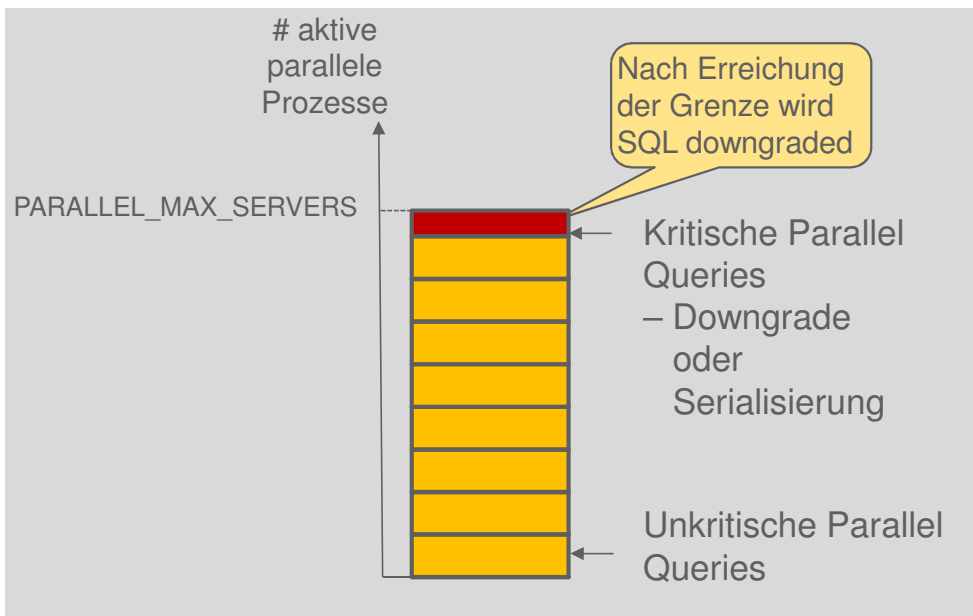
Utilization Limits für gleichförmige Performance bzw. Begrenzung

PDB Resource Plan

- **PDB Resource Pläne** enthalten Direktiven für die PDB Consumer Groups
- Aktivierung im jeweiligen Container
- Vor: Es muss ein CDB Plan existieren!
- Vergleichbar mit NON CDB; allerdings mit Einschränkungen

Non-PDB Resource Plan	PDB Resource Plan
Multi-level Ressource Pläne	Nur Single-level Ressource Pläne
Bis zu 32 Consumer Groups	Bis zu 8 Consumer Groups
Sub Pläne	Keine Sub Pläne

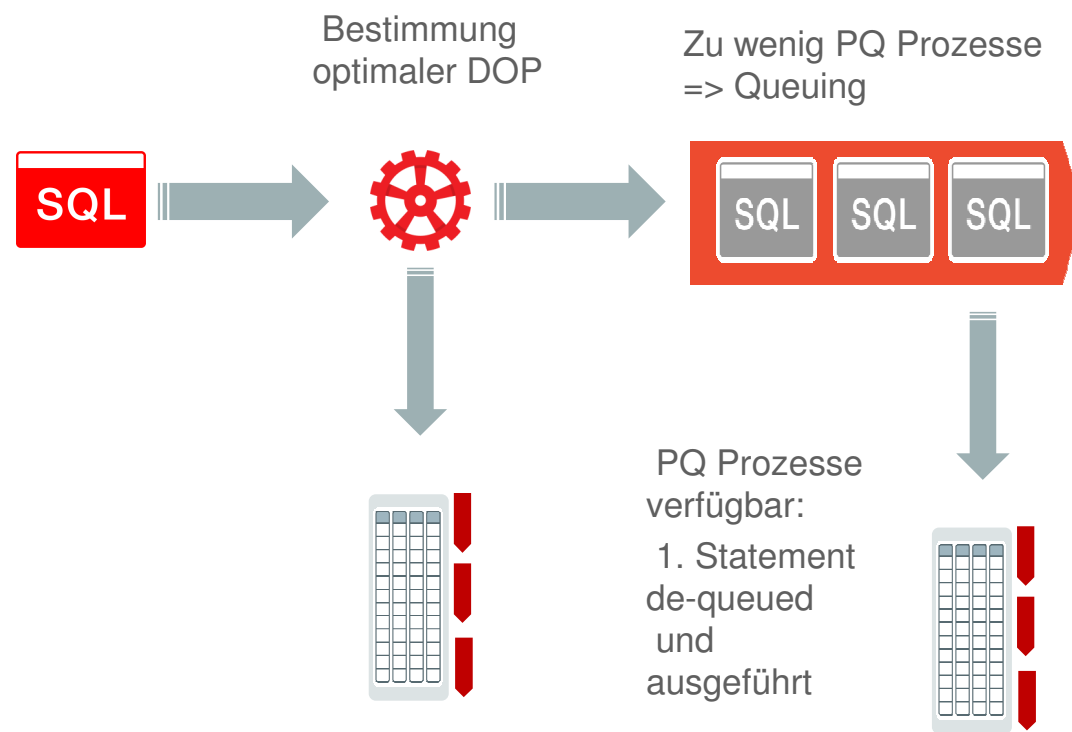
Anwendungsfall 4: Parallel Statement Queuing



- Wieso Parallel Statement Queuing?
 - Verhinderung von DOP Downgrades
 - Kontrolle über die Reihenfolge der Ausführung

=> kritische parallele Queries immer zuerst!

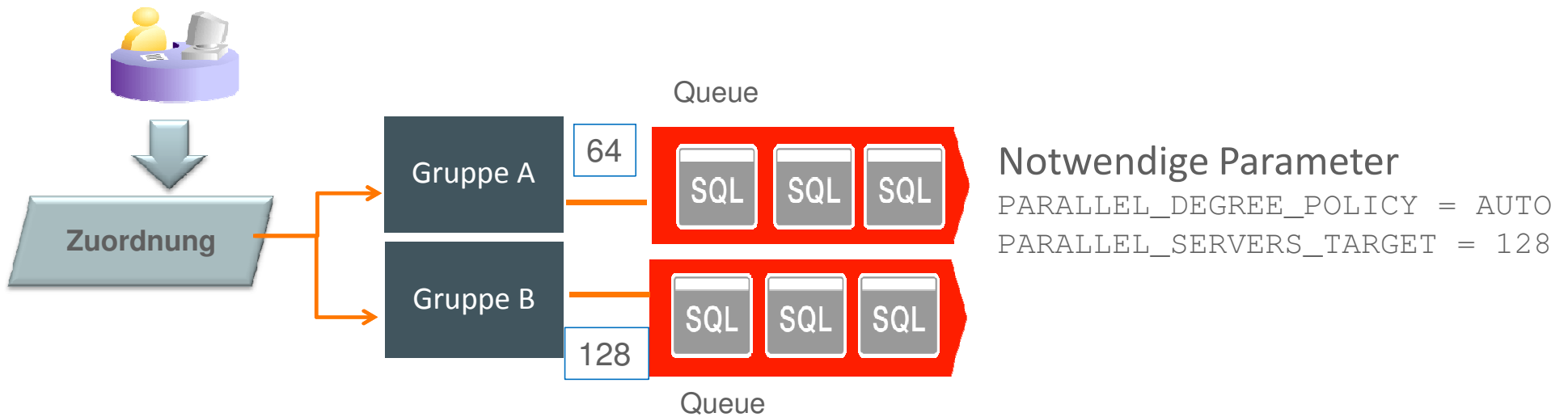
Parallel Statement Queuing ohne Resource Manager



Notwendige Parameter

```
PARALLEL_DEGREE_POLICY = AUTO  
PARALLEL_SERVERS_TARGET = #PQs
```

Statement Queuing mit Resource Manager am Beispiel



Im Resource Manager für Gruppe A:

```
mgmt_p1 = 40%  
parallel_degree_limit_p1 = 8  
parallel_target_percentage = 50 => 64 PX  
parallel_sever_limit = 50 (ab 12c)
```

Im Resource Manager für Gruppe B:

```
mgmt_p1 = 60%  
parallel_degree_limit_p1 = 32  
parallel_target_percentage = 100 => 128 PX  
parallel_sever_limit = 100 (ab 12c)
```

Statement Queuing mit Resource Manager

- Begrenzung der Parallel Server von nicht kritischen Workloads
- 12c: Kritische Statements von Consumer Group CRITICAL werden nicht ge-queued!

Festlegung der Ausführungs Reihenfolge der Consumer Group Queues

Kritisches SQL wird nie ge-queued

Limitiere die Parallel Server, die von der Consumer Group genutzt werden können

Limitiere den DOP

Consumer Group	Allocation (mgmt_p1)	Critical <small>Neu in 12c</small>	Parallel Server Percentage	Parallel Degree Limit
Critical	80	Yes		
Report	15	No	50	8
Other	5	No	25	4

Agenda

- 1 Einführung – ein Kurzprofil
- 2 Setup – schnell mal Resource Manager aufsetzen!
- 3 Anwendungsfälle
- 4 **Fazit, Ausblick, weitere Informationen**

Fazit

- Resource Manager arbeitet nur unter Last!
- Aber: Zu hohe Gesamtlast resultiert in “Sessions Waiting for CPU” (siehe *resmgr:cpu_quantum waits* im AWR)
- Einfache Pläne bevorzugen!
- SYS_GROUP und OTHER_GROUPS nicht vergessen!
- Von Zeit zu Zeit monitoren
- Einsatz von Parallel Statement Queuing bei der Priorisierung von konkurrierenden Workloads (Voraussetzung AUTO DOP)



Informationen, White Paper, MOS ...

- Instance Caging White Paper
<http://www.oracle.com/technetwork/database/focus-areas/performance/instance-caging-wp-166854.pdf>
- Resource Manager White Paper
<http://www.oracle.com/technetwork/database/focus-areas/performance/resource-manager-twp-133705.pdf>
- Best Practices for Database Consolidation on Exadata White Paper
<http://www.oracle.com/technetwork/database/features/availability/exadata-consolidation-522500.pdf>
- Master MOS Note 1339769.1
 - Bug Fixes, Step-by-step Konfiguration für einige Szenarien, Monitoring und Tuning Skripte
- MOS Note 1338988.1: Scripts and Tips for Monitoring CPU Resource Manager