



Ressourcen Management ... alles automatisch?

Johannes Ahrends
Geschäftsführer
CarajanDB GmbH



ORACLE
ACE

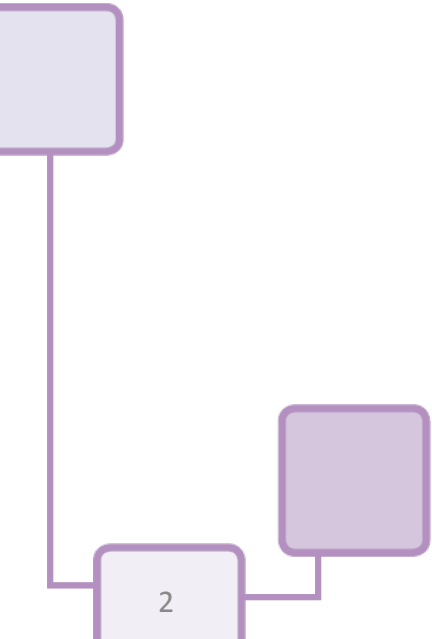
2012
DOAG
Konferenz + Ausstellung



Agenda



- CarajanDB GmbH
- Warum Ressourcen Management
- Server-Ressourcen
- Instance-Ressourcen
- Applikations-Ressourcen



- **Experten mit über 30 Jahren Oracle Erfahrung**
- **Spezialisten für**
 - Backup & Recovery
 - Hochverfügbarkeit
 - Healthchecks
 - Performance Optimierung
 - Einsatz von Oracle Standard Edition
 - Oracle in virtuellen Umgebungen und in der Cloud
 - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
 - Monitoring (Grid / Cloud Control, HLMM, Foglight, Spotlight)
- **Schulung und Workshops (Oracle, Toad)**



Warum Ressourcen Management?



Alle Vögel wollen ans Futter.

- Wer bekommt welchen Platz?
- Ist es besser, direkt an der Quelle zu sitzen oder die Samen vom Boden aufzusammeln?

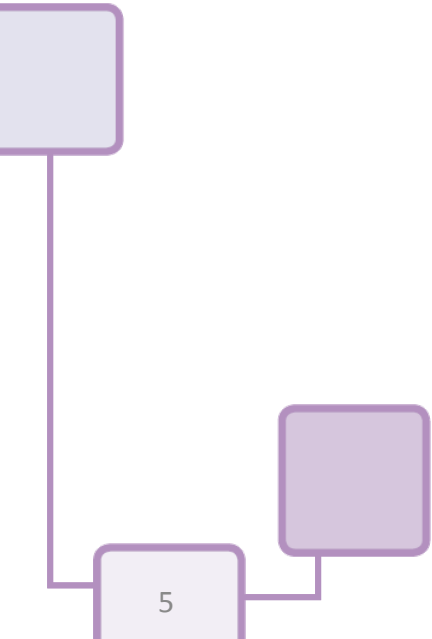
... und wer ist eigentlich der Resource-Manager?

Warum Ressourcen Management

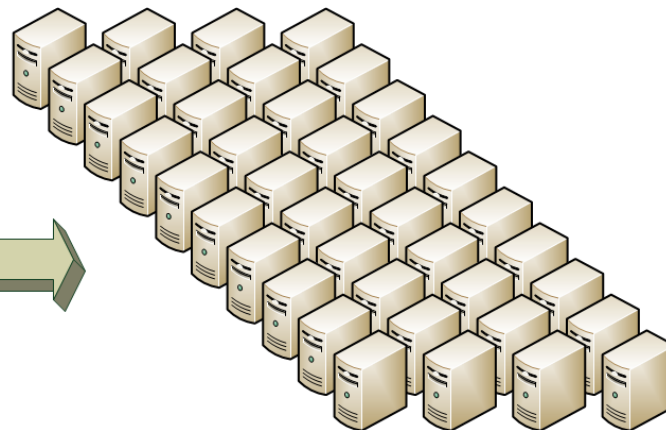
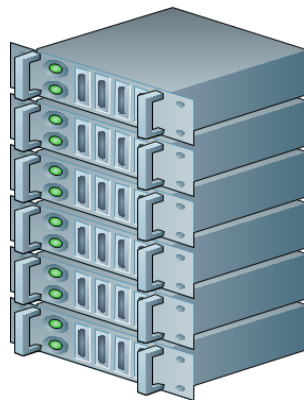


➤ Zwei Strömungen

- Virtualisierung
- Konsolidierung



- n physikalische Server mit $m (>n)$ virtuellen Servern
- I/O und Netzwerk gemeinsam
- CPU und Memory separat zugeordnet
- Notwendig:
 - Anpassung von Ressourcen im laufenden Betrieb



- n Server mit m ($>n$) Datenbanken
- I/O, Netzwerk, CPU, Memory gemeinsam
- Zuweisung von Ressourcen über Parameter
- **Notwendig:**
 - Schützen von kritischen Anwendungen
 - Verschieben von Ressourcen im laufenden Betrieb



- **Mehrere Datenbanken auf einem Server**
 - Ressourcen Management für die Instanz-Verwaltung
- **Mehrere Schemata in einer Datenbank**
 - Ressourcen Management, um Schemata oder Prozesse zu priorisieren
- **Flexible Anpassung**
 - Anpassung der Ressourcen nach Kriterien, wie z.B.
 - Geschäftsverlauf (Weihnachtsgeschäft)
 - Batchläufe
 - Administration
- **Überprovisionierung von Ressourcen**

- **Tablespace und Datafiles**
 - Können schemaspezifisch aufgeteilt werden
- **I/O Last**
 - Kann über entsprechende Plattensysteme angepasst werden
- **Memory**
 - Serverparameter
- **CPU**
 - Instance Caging

- bitmap_merge_area_size
- count
- buffer

- db_
- db_nk_cache
- db_flash_cache_
- db_keep_cache_size
- db_recycle_cache_size
- java_pool_size
- large_pool_size
- log_buffer
- memory_target
- memory_max_target
- olap_page_pool_size

- open_cursors
- pga_aggregate_target
- processes
- result_cache_max_size
- sessions
- sga_max_size
- sga_target
- shared_pool_size
- shared_pool_reserved_size
- sort_area_size
- sort_area_reserved
- sort_area_retained
- transactions
- workarea_size_policy

Oracle Verwaltung ist kompliziert!

MEMORY_TARGET

Instance Ressourcen jetzt...

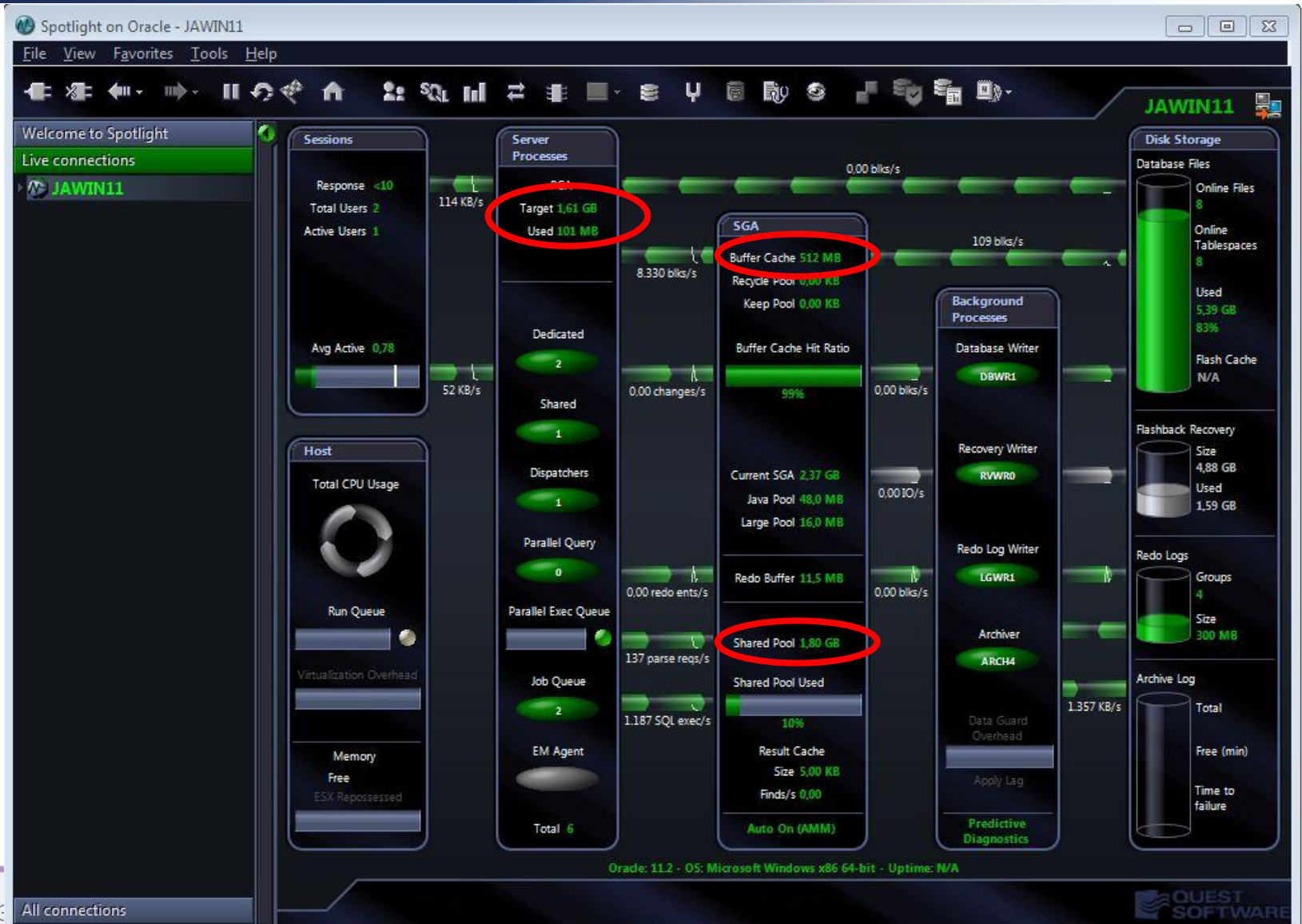
cpu_count

target

- open_cursors
- processes
- sessions

Oracle Verwaltung ist einfach!

... und in der Praxis?



Init.ora – Oracle 11g

```
jawin11.__db_cache_size=536870912
jawin11.__java_pool_size=50331648
jawin11.__large_pool_size=16777216
jawin11.__pga_aggregate_target=1728053248
jawin11.__sga_target=2566914048
jawin11.__shared_io_pool_size=0
jawin11.__shared_pool_size=1929379840
jawin11.__streams_pool_size=0
*.memory_target=4G
```

v\$sgainfo – Oracle11g

```
SQL> select * from v$sgainfo;
```

NAME	BYTES	RES
-----	-----	---
Fixed SGA Size	2262048	No
Redo Buffers	12103680	No
Buffer Cache Size	536870912	Yes
Shared Pool Size	1929379840	Yes
Large Pool Size	16777216	Yes
Java Pool Size	50331648	Yes
Streams Pool Size	0	Yes
Shared IO Pool Size	0	Yes
Granule Size	16777216	No
Maximum SGA Size	4275781632	No
Startup overhead in Shared Pool	128940664	No
Free SGA Memory Available	1728053248	

➤ Und was ist mit der „Free SGA Memory Available“?

```
SELECT * from v$pgastat  
WHERE name LIKE '%target parameter%';
```

NAME	VALUE	UNIT
-----	-----	-----
aggregate PGA target parameter	1728053248	bytes

x\$ksppi, x\$ksppcv

```
SQL> SELECT i.ksppinm name, v.ksppstvl value
       FROM x$ksppi i, x$ksppcv v
       WHERE i.indx = v.indx
              AND i.ksppinm LIKE '\_ \_ %' ESCAPE '\';
```

NAME	VALUE
-----	-----
__oracle_base	D:\oracle
__shared_pool_size	1929379840
__large_pool_size	16777216
__java_pool_size	50331648
__streams_pool_size	0
__sga_target	2566914048
__db_cache_size	536870912
__shared_io_pool_size	0
__dg_broker_service_names	
__pga_aggregate_target	1728053248

Beispiel (Oracle 11g)

```
SQL> ALTER SYSTEM SET memory_target = 2G scope=spfile;
SQL> STARTUP FORCE
SQL> SELECT * FROM v$sgainfo;
```

NAME	BYTES	RES
Fixed SGA Size	2229944	No
Redo Buffers	5173248	No
Buffer Cache Size	352321536	Yes
Shared Pool Size	385875968	Yes
Large Pool Size	16777216	Yes
Java Pool Size	16777216	Yes
Streams Pool Size	0	Yes
Shared IO Pool Size	0	Yes
Granule Size	16777216	No
Maximum SGA Size	2087780352	No
Startup overhead in Shared Pool	137554960	No
Free SGA Memory Available	1308622848	

➤ Vergrößern des Buffercaches:

```
SQL> ALTER SYSTEM SET db_cache_size=900M;  
ALTER SYSTEM SET db_cache_size=900M  
*
```

ERROR at line 1:

ORA-02097: parameter cannot be modified because specified value is invalid
ORA-00384: Insufficient memory to grow cache

➤ Verkleinern der PGA:

```
SQL> ALTER SYSTEM SET pga_aggregate_target=200M sid='*';  
System altered.
```

```
SQL> ALTER SYSTEM SET db_cache_size=900M sid='ORASE1';  
ALTER SYSTEM SET db_cache_size=900M sid='ORASE1'  
*
```

ERROR at line 1:

ORA-02097: parameter cannot be modified because specified value is invalid
ORA-00384: Insufficient memory to grow cache

➤ Alle Cache-Parameter sind jetzt Untergrenzen!

- Durchstarten der Instanz mit neuen Parametern
- Alternativ: setzen von `sga_target`

```
SQL> ALTER SYSTEM SET sga_target=1500M;
```

```
System altered.
```

```
SQL> select * from v$sgainfo;
```

NAME	BYTES	RES
-----	-----	---
Fixed SGA Size	2229944	No
Redo Buffers	5173248	No
Buffer Cache Size	1140850688	Yes
Shared Pool Size	385875968	Yes
Large Pool Size	16777216	Yes
Java Pool Size	16777216	Yes
Streams Pool Size	0	Yes
Shared IO Pool Size	0	Yes
Granule Size	16777216	No
Maximum SGA Size	2087780352	No
Startup overhead in Shared Pool	137554960	No
Free SGA Memory Available	520093696	

- **Mindestens setzen:**
 - `db_cache_size`
 - `shared_pool_size`
- **Eventuell:**
 - `cursor_sharing = FORCE`
 - Vorsicht mit tatsächlich gewünschten Literalen
- **Shared Pool kann im laufenden Betrieb kaum freigegeben werden**
 - `ALTER SYSTEM FLUSH SHARED_POOL`

... und warum „...max...?“

- Oracle 10g: `sga_target` und `sga_max_size`
- Oracle 11g: `memory_target` und `memory_max_target`
- Grund: Flexibles Verschieben von Memory Bereichen zwischen Datenbanken
 - Nur Sinnvoll, wenn mehrere Datenbanken auf einem Server
- Beispiel: Server 32 GB Memory, 2 Datenbanken:
 - ORADB1: `memory_target` = 16 GB
 `memory_max_target` = 20 GB
 - ORADB2: `memory_target` = 10 GB
 `memory_max_target` = 20 GB
 - Für bestimmte Batchläufe kann jetzt ORADB2 mehr Speicher zugewiesen werden, wenn ORADB1 gleichzeitig weniger bekommt.

... und warum „...target...“?

- **Target = Ziel**
- **Man kann schon mal über das Ziel hinausschießen!**
- `PGA_AGGREGATE_TARGET` specifies the target aggregate PGA memory available to all server processes attached to the instance. ... Oracle attempts to keep the amount of private memory below the target specified by this parameter by adapting the size of the work areas to private memory.

(Auszug aus Oracle® Database Reference 11g Release 2 (11.2))

Und was ist mit der CPU?

➤ Instance Caging

- `cpu_count=x` → explizite Zuweisung von CPUs zu Instanzen

➤ Voraussetzung

- Oracle 11g Release 2
- Resource Manager eingeschaltet

- Zuteilung von maximalen CPU-Ressourcen
- Begrenzung des Parallelisierungsgrades für Datenbankoperationen
- Begrenzung der Anzahl gleichzeitig aktiver Datenbanksitzungen
- Vorgaben für die Nutzung von Speicherplatz in Rollback-Segmenten (undo pool)

- Konsumentengruppen (Consumer Groups)
 - gruppieren Benutzer und deren Sessions.
- Ressourcenpläne (Resource Plans)
 - enthalten allgemeine Regeln zur Nutzung von Ressourcen
- Direktiven (plan directive)
 - verknüpfen Konsumentengruppen mit Ressourcenplänen und erweitern das Regelwerk durch Regeln für maximale Ressourcennutzungen für bestimmte Gruppen.

1. Resource Manager Plan

➤ Zwei unterschiedliche Pläne

- Tagesbetrieb und Nachtbetrieb
- Umschaltung über scheduler, cron, etc.

```
BEGIN
  dbms_resource_manager.create_plan(
    PLAN      => 'TAGESPLAN',
    COMMENT   => 'Plan Tagesbetrieb 8 bis 19 Uhr');
  dbms_resource_manager.create_plan(
    PLAN      => 'Nachtplan',
    COMMENT   => 'Plan Nachtbetrieb 19 bis 8 Uhr');
END;
```

2. Konsumentengruppen

➤ Gruppen für die Einordnung von Usern, etc.

```
BEGIN
  dbms_resource_manager.create_consumer_group (
    consumer_group => 'CARAJANDB_OLTP',
    COMMENT        => 'Meine OLTP GRUPPE');
  dbms_resource_manager.create_consumer_group (
    consumer_group => 'CARAJANDB_BATCH',
    COMMENT        => 'Meine Batch Gruppe');
END;
```

➤ Zuweisung von Ressourcen für Gruppen und Plänen

```
BEGIN
  dbms_resource_manager.create_plan_directive(
    PLAN                => 'TAGESPLAN',
    group_or_subplan    => 'CARAJANDB_OLTP',
    COMMENT              => 'OLTP-Ressourcen waehrend des Tages',
    mgmt_p1              => 90);
  dbms_resource_manager.create_plan_directive(
    PLAN                => 'TAGESPLAN',
    group_or_subplan    => 'CARAJANDB_BATCH',
    COMMENT              => 'BATCH-Ressourcen waehrend des Tages',
    mgmt_p1              => 0,
    mgmt_p2              => 90);
  dbms_resource_manager.create_plan_directive(
    PLAN                => 'TAGESPLAN',
    group_or_subplan    => 'OTHER_GROUPS',
    COMMENT              => 'Muss definiert werden!',
    mgmt_p1              => 0,
    mgmt_p2              => 0,
    mgmt_p3              => 100);
```

4. Schemazuordnung

- Erteilung der Erlaubnis, dass ein Schema eine Consumergruppe nutzen darf
- Sollte für jeden User gemacht werden!

```
BEGIN
  sys.dbms_resource_manager_privs.grant_switch_consumer_group (
    grantee_name      => 'JOHANNES',
    consumer_group    => 'CARAJANDB_OLTP',
    grant_option      => FALSE);
  sys.dbms_resource_manager_privs.grant_switch_consumer_group (
    grantee_name      => 'JOHANNES',
    consumer_group    => 'CARAJANDB_BATCH',
    grant_option      => FALSE);
  sys.dbms_resource_manager.set_initial_consumer_group (
    user              => 'JOHANNES',
    consumer_group    => 'CARAJANDB_OLTP');
END;
```

5. Ausnahmen bestätigen die Regel!

➤ Zuordnung von Users zu anderen Ressourcengruppen aufgrund bestimmter Regeln

- client_machine
- client_os_user
- client_program
- module_name
- module_name_action
- oracle_user
- service_module
- service_module_action
- service_name

➤ Änderung der Ressourcengruppe aufgrund der Verwendung eines bestimmten Services

```
BEGIN
  dbms_resource_manager.set_consumer_group_mapping_pri(
    explicit                => 1,  -- muss immer so stehen
    service_name           => 2,
    client_program          => 3,
    oracle_user             => 4,
    client_os_user          => 10,
    client_machine          => 9,
    module_name             => 8,
    module_name_action      => 7,
    service_module          => 6,
    service_module_action   => 5);
END;
```

2. Konsumentengruppen

➤ Gruppen für die Einordnung von Usern, etc.

```
BEGIN
dbms_resource_manager.set_consumer_group_mapping (
    ATTRIBUTE      => dbms_resource_manager.service_name,
    value          => 'PRIO_SERVICE',
    consumer_group => 'CARAJANDB_OLTP');
END;
```

PLUGGABLE DATABASE

(alles wird anders)



Fragen?

Weitere Infos:
www.doag.org
www.carajandb.com