



Automatic Memory Management

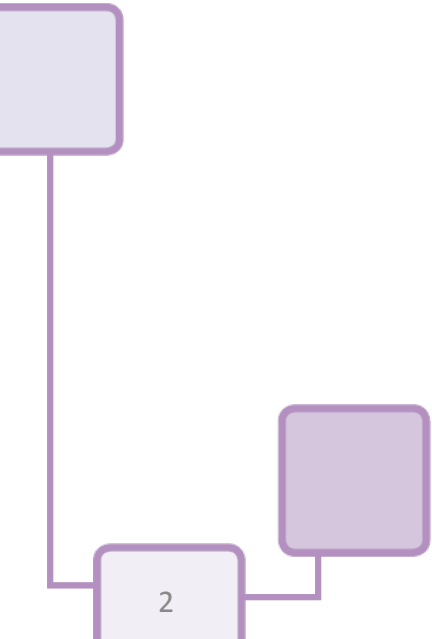
... geht das wirklich alles automatisch?

Johannes Ahrends
Geschäftsführer CarajanDB

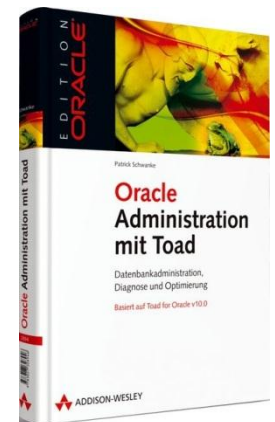


Agenda

- **Historie**
- **Voraussetzung bei Linux**
- **Vergleich Version 10.2 / 11.2**
- **Beispiel**



- 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)



- Oracle Verwaltung ist kompliziert!**

➤ Oracle8, 8i

- Diverse Parameter für SGA und PGA
 - db_buffer_cache, shared_pool_size, sort_area_size, java_pool_size, large_pool_size, log_buffer, ...

➤ Oracle9i

- db_cache_size ersetzt db_block_buffer
- Unterschiedliche Blocksizes bei Tablespaces verlangen unterschiedliche db_nk_cache_size
- Ersetzen der sort_area_size:
 - pga_aggregate_target => Gesamtgröße aller Sortierbereiche
 - workarea_size_policy => Ein- oder Ausschalten
- Neuer Parameter: streams_pool_size

➤ Oracle 10g

- Zusammenfassung der Parameter für SGA
 - `sga_target` => Gesamtheit aller SGA Caches
 - `sga_max_size`

➤ Oracle 11g

- Ein Parameter für alle Memorybereiche auf dem Server
 - `memory_target`
 - `memory_max_target`

➤ Bis Oracle 10g: System-V Style

```
[root@ora11gr2 ~]# ipcs -m
```

```
----- Shared Memory Segments -----
```

key	shmid	owner	perms	bytes	nattch	status
0x00000000	65536	oracle	600	393216	2	dest
0x00000000	98305	oracle	600	393216	2	dest
0x00000000	131074	oracle	600	393216	2	dest
0x00000000	163843	oracle	600	393216	2	dest
0x00000000	196612	oracle	600	393216	2	dest
0x00000000	229381	oracle	600	393216	2	dest
0x00000000	262150	oracle	600	393216	2	dest
0x00000000	294919	oracle	600	393216	2	dest
0x00000000	327688	oracle	600	393216	2	dest
0x75fc1f04	285376521	oracle	660	348127232	34	
0x00000000	425994	oracle	600	393216	2	dest

➤ SGA ist ca. 330 MB, 34 Prozesse greifen darauf zu (Server- und Hintergrundprozesse)

➤ Ab Oracle 11g: Posix Style

```
[root@ora11gr2 ~]# ipcs -m
```

```
----- Shared Memory Segments -----
```

key	shmid	owner	perms	bytes	nattch	status
0x00000000	65536	oracle	600	393216	2	dest
0x00000000	98305	oracle	600	393216	2	dest
0x00000000	131074	oracle	600	393216	2	dest
0x00000000	163843	oracle	600	393216	2	dest
0x00000000	196612	oracle	600	393216	2	dest
0x00000000	229381	oracle	600	393216	2	dest
0x00000000	262150	oracle	600	393216	2	dest
0x00000000	294919	oracle	600	393216	2	dest
0x00000000	327688	oracle	600	393216	2	dest
0x00000000	285442057	oracle	660	4096	0	
0x00000000	425994	oracle	600	393216	2	dest
0x00000000	285474827	oracle	660	4096	0	
0x75fc1f04	285507596	oracle	660	4096	0	

➤ Die SGA ist jetzt scheinbar nur 4 KB groß und hat keine weiteren Prozesse

- Posix Struktur erlaubt die Allokierung, Deallokierung und Verschiebung von „Granulen“ zwischen SGA und PGA.
- Verwaltet über shmfs oder tmpfs

```
[oracle@ora11gr2 ~]$ df -k /dev/shm
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
tmpfs	614400	335688	278712	55%	/dev/shm

```
[oracle@ora11gr2 ~]$ ls -l /dev/shm
```

```
-rw-r-----. 1 oracle dba 4194304 Jan  9 12:57 ora_VMLIN112_131075_0
-rw-r-----. 1 oracle dba 4194304 Jan  9 12:57 ora_VMLIN112_131075_1
-rw-r-----. 1 oracle dba          0 Dec 20 13:59 ora_VMLIN112_131075_10
-rw-r-----. 1 oracle dba 4194304 Jan  9 12:53 ora_VMLIN112_131075_100
-rw-r-----. 1 oracle dba 4194304 Jan  9 12:57 ora_VMLIN112_131075_101
```

➤ Beim ersten Startup:

```
SQL> startup
```

```
ORA-00845: MEMORY_TARGET not supported on this system
```

➤ Alert-File:

```
oracle@ora11gr2 trace]$ tail -3 alert_PS112.log
```

```
Starting ORACLE instance (normal)
```

```
WARNING: You are trying to use the MEMORY_TARGET feature. This  
feature requires the /dev/shm file system to be mounted for at least  
2097152000 bytes. /dev/shm is either not mounted or is mounted with  
available space less than this size. Please fix this so that  
MEMORY_TARGET can work as expected. Current available is 629145600  
and used is 0 bytes. Ensure that the mount point is /dev/shm for this  
directory.
```

```
memory_target needs larger /dev/shm
```

➤ Anpassen des Eintrags in /etc/fstab, z.B.:

```
tmpfs          /dev/shm      tmpfs    size=2000m    0 0
```

➤ **Anpassen des Eintrags in /etc/fstab, z.B.:**

```
tmpfs          /dev/shm      tmpfs    size=2000m    0 0
```

➤ **Besonderheit Oracle Linux 6:**

Editieren von: /etc/rc.d/rc.sysinit

```
mount -f /dev/shm >/dev/null 2>&1
```

Ändern in:

```
mount /dev/shm >/dev/null 2>&1
```

init.ora – Oracle 10g

```
jawin10.__db_cache_size=1174405120  
jawin10.__java_pool_size=16777216  
jawin10.__large_pool_size=16777216  
jawin10.__shared_pool_size=352321536  
jawin10.__streams_pool_size=0  
*.pga_aggregate_target=524288000  
*.sga_target=1572864000
```

v\$sgainfo – Oracle10g

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

NAME	BYTES	RES
-----	-----	---
Fixed SGA Size	2077840	No
Redo Buffers	14696448	No
Buffer Cache Size	1174405120	Yes
Shared Pool Size	352321536	Yes
Large Pool Size	16777216	Yes
Java Pool Size	16777216	Yes
Streams Pool Size	0	Yes
Granule Size	16777216	No
Maximum SGA Size	1577058304	No
Startup overhead in Shared Pool	67108864	No
Free SGA Memory Available	0	

Init.ora – Oracle 11g

```
jawin11.__db_cache_size=687865856
jawin11.__java_pool_size=16777216
jawin11.__large_pool_size=16777216
jawin11.__oracle_base='D:\oracle'#ORA...
jawin11.__pga_aggregate_target=738197504
jawin11.__sga_target=1358954496
jawin11.__shared_io_pool_size=0
jawin11.__shared_pool_size=603979776
jawin11.__streams_pool_size=16777216
*.memory_target=2097152000
```

v\$sgainfo – Oracle11g

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

NAME	BYTES	RES
Fixed SGA Size	2254768	No
Redo Buffers	5148672	No
Buffer Cache Size	687865856	Yes
Shared Pool Size	603979776	Yes
Large Pool Size	16777216	Yes
Java Pool Size	16777216	Yes
Streams Pool Size	16777216	Yes
Shared IO Pool Size	0	Yes
Granule Size	16777216	No
Maximum SGA Size	2087780352	No
Startup overhead in Shared Pool	134217728	No
Free SGA Memory Available	738197504	

➤ 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	738197504	bytes

Demobeispiel 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_block_buffer`
 - `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.

- **Patrick Schwanke:**
 - Blog: *databasesincloud.wordpress.com*
- **Johannes Ahrends:**
 - Blog: *streetkiter.wordpress.com*
- **www.carajandb.com**



Fragen?

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