

Practice

Managing Database Memory

Practice Target

In this practice, you will implement the Automatic Memory Management (AMM) and the Automatic Shared Memory Management (ASMM) methods on `srv1`.

Practice Assumptions

You have the `srv1` and its **CDB** database up and running.



Configuring Automatic Memory Management (AMM)

In the following steps, you will configure AMM on `srv1`.

Note: In real life scenarios, AMM is considered for a database system when the total physical memory is 4G.

1. Create a Putty session to `srv1` as `oracle`

2. Invoke SQL*Plus and login to the CDB as `SYS`

```
sqlplus / as sysdba
```

3. Retrieve the value of the parameters `MEMORY_TARGET` and `MEMORY_MAX_TARGET`

The parameters are set to zero because we disabled the AMM at the time of creating the database.

`MEMORY_MAX_TARGET` is a static (can be changed in the SPFILE only and the instance must be restarted to take effect). `MEMORY_TARGET` is a dynamic parameter.

```
SHOW PARAMETER MEMORY_TARGET
SHOW PARAMETER MEMORY_MAX_TARGET
```

4. Set the value of the parameters `MEMORY_TARGET` and `MEMORY_MAX_TARGET` to 3200M.

When I tried to set the memory target to any value less than 3200M, I receive the following error at the time of starting up the database:

```
ORA-00838: Specified value of MEMORY_TARGET is too small, needs to be at least
3200M
```

```
ALTER SYSTEM SET MEMORY_TARGET=3200M SCOPE=SPFILE;
ALTER SYSTEM SET MEMORY_MAX_TARGET=3200M SCOPE=SPFILE;
```

5. Exit from SQL*Plus

```
exit
```

6. Disable the database auto-restart by opening the file `/etc/oratab` and set the database flag to `N`.

We disable the database auto-restart because we want to manually startup the database after the change and check on the change impact on the startup.

```
vi /etc/oratab
```

7. Change the current user to `root` and shutdown `srv1`.

We should shutdown the vm because we need to change its total allocated memory to 4G.

```
shutdown -h now
```

8. In Oracle VirtualBox, change the memory allocated to srv1 to 4G.

Settings > System > Under the Motherboard tab, set the Base Memory to 4096 MB > click on OK button.

9. Start the vm.

10. Open Putty and connect to the vm as `oracle`

11. Invoke SQL*Plus and login to the database as `SYS`

```
sqlplus / as sysdba
```

12. Startup the database.

The command should return the following error:

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

We receive this error because the amount of shared memory is less than the `MEMORY_TARGET`.

```
STARTUP
```

13. Exit from SQL*Plus

```
Exit
```

14. Run the following command to check out the amount of shared memory.

In our environment, the amount reserved for the shared memory `/dev/shm` is 1.8G.

```
df -h
```

15. Change the current user to root then run the following commands to increase the shared memory mount size to 3200 MB.

These commands temporarily change the shared memory size. To make the change persistence, we should continue with running further commands, but this is not our target for the long run.

```
umount tmpfs
mount -t tmpfs shmfs -o size=3200m /dev/shm
```

16. Change the current user to `oracle` and startup the database.

```
su - oracle
sqlplus / as sysdba
startup
```

In real life scenario, after enabling the AMM, we would allow the system to run against the database as normal and monitor the memory usage.

17. Submit the following query to retrieve the memory area sizes set by the database.

This view is our tool to watch the memory component sizes.

From the query output, the summation of "SGA Target" and "PGA Target" should be 3200 MB.

```
col COMPONENT format a22
SELECT COMPONENT, CURRENT_SIZE/1024/1024 CURRENT_SIZE_MB,
USER_SPECIFIED_SIZE/1024/1024 USER_SPECIFIED_MB
FROM V$MEMORY_DYNAMIC_COMPONENTS WHERE CURRENT_SIZE<>0;
```

18. Check the first lines of the SPFILE.

The database saves the current values of the memory areas into the header of the SPFILE. The values are saved into their corresponding initialization parameters.

When the database is shut down and later is started up, the database reads the values of those parameters and uses them to set the initialized memory area sizes.

```
HOST head -n 15 $ORACLE_HOME/dbs/spfileoradb.ora
```

After the database is under normal workload for a few hours or days, we want to make sure that the memory size is set to its optimized value. The automatic memory advisor helps us to tune the database memory target.

19. Submit the following query to retrieve the Target Memory Advisor. Study the optimal size of the memory target setting in the practice database.

The advisor estimations are not relevant because the database is not under any workload from users. Later in the course, you will learn how to apply workload stress on the database.

```
set linesize 180
SELECT * FROM V$MEMORY_TARGET_ADVICE ORDER BY MEMORY_SIZE;
```

Cleanup

20. Shutdown `srv1`
21. In Oracle VirtualBox Manager, restore `srv1` from its CDB snapshot.
22. Verify that its memory is set back to 6 GB.
23. Startup `srv1`
24. Open Putty and connect to `srv1` as `oracle`

Configuring Automatic Shared Memory Management (ASMM)

In the following steps, you will configure ASMM on `srv1`.

Note: ASMM is the recommended memory management option in Oracle databases.

25. Display the total memory size in the system.

The total memory that appears in `srv1` is 5.6G.

```
free -h
```

26. Assume that the memory usage plan would be as follows:

DB memory = %70 of total memory = $70\% * 5.6 \approx 3.9$ G

SGA size = %60 * total db memory size = $0.6 * 3.9 \approx 2.352$ G ≈ 2408 M

PGA size = %40 * total db memory size = $0.4 * 3.9 \approx 1.5$ G ≈ 1500 M

In real life scenarios, generally speaking, the amount of memory reserved for each area depends on the number of users, the transactions rate, and the average size of retrieved data.

27. Verify that AMM is disabled.

If the AMM is enabled, we cannot configure the ASMM.

```
sqlplus / as sysdba
```

```
SHOW PARAMETER MEMORY_TARGET
```

28. Set `SGA_MAX_SIZE` to a 4096M then restart the database instance.

We set this value to the maximum possible SGA size we might assign to this parameter. We want to avoid restarting the database instance when we want to set the SGA target size.

```
ALTER SYSTEM SET SGA_MAX_SIZE = 4096M SCOPE=SPFILE;
```

```
-- restart the database
SHUTDOWN IMMEDIATE
STARTUP
```

29. Set the SGA and PGA target sizes.

The second statement returns the following error:

```
ORA-00855: PGA_AGGREGATE_TARGET cannot be set because of insufficient physical memory.
```

We should receive this error when we assign a value to the PGA higher than the available memory. This is not the case in our environment. In fact, we keep receiving the same error no matter the size we assign to the parameter.

It looks like this is a bug and the only workaround that I am aware of is to send the parameter value in the SPFILE and restart the instance.

```
ALTER SYSTEM SET SGA_TARGET= 2408M SCOPE=BOTH;
ALTER SYSTEM SET PGA_AGGREGATE_TARGET= 1500M SCOPE=BOTH;
```

30. Set the PGA size in the SPFILE and restart the instance.

```
ALTER SYSTEM SET PGA_AGGREGATE_TARGET= 1500M SCOPE=SPFILE;
-- restart the database
SHUTDOWN IMMEDIATE
STARTUP
```

31. Retrieve the memory area sizes set by the database.

The total amount of memory assigned to the PGA and SGA are the same as defined.

```
col COMPONENT format a22
SELECT COMPONENT, CURRENT_SIZE/1024/1024 CURRENT_SIZE_MB,
USER_SPECIFIED_SIZE/1024/1024 USER_SPECIFIED_MB
FROM V$MEMORY_DYNAMIC_COMPONENTS WHERE CURRENT_SIZE<>0;
```

32. Query the memory target advisor.

Because the AMM is disabled, the advisor is not active.

```
SELECT * FROM V$MEMORY_TARGET_ADVICE ORDER BY MEMORY_SIZE;
```

33. Query the ASMM target advisor and determine the optimize SGA size.

We normally query this advisor after the database is under normal stress for long time. Because our database has no stress, the advisor cannot estimate any performance gain from changing the SGA_TARGET size.

```
SELECT SGA_SIZE, SGA_SIZE_FACTOR, ESTD_DB_TIME, ESTD_DB_TIME_FACTOR FROM
V$SGA_TARGET_ADVICE ORDER BY SGA_SIZE;
```

Cleanup**34. Shutdown `srv1` and restore it from its CDB snapshot.**

Summary

- Configuring AMM and the ASMM are implemented by setting their corresponding parameters. For each configuration, we have a parameter to set its maximum limit and a parameter to set the area sizes.
- It is recommended to stick with ASMM.
- Query the advisor views to check on the optimal sizes for the AMM and ASMM.

