Practice 8

Managing Services in Oracle RAC

Practice Overview

In this practice you will manage the dynamic database services. Specifically, you will:

- Create a database dynamic service
- Configure the `tnsnames.ora` file to connect to a service
- Test the failover functionality in the service
- Relocate a service from one instance to another
- Collect and view statistics on the sessions connected to a service grouped by module and action.

Practice Assumptions

- The practice assumes that you have the Oracle RAC database up and running in the virtual machines `srv1` and `srv2`.

Ahmed Baraka Oracle Database Administrator

A. Creating and Testing Services

In this section of the practice, you will create one order entry service named `soesrv`. You will then test the reaction of Oracle database service when the preferred instance crashes.

1. Open a Putty terminal window and connect to `srv1` as `oracle`.

2. Create a service named as `soesrv` and make `rac1` as its preferred instance and `rac2` as its available instance.

```
srvctl add service -db rac -service soesrv -preferred rac1 -available rac2
```

3. Start the service and make sure it successfully started.

```
srvctl start service -db rac -s soesrv

# observe the node on which the service is currently running:
srvctl status service -db rac -s soesrv
```

4. Configure the `tnsnames.ora` file to allow connections through the created service.

Copy the code from the downloadable file. Do not copy it from the PDF file.

```
vi $TNS_ADMIN/tnsnames.ora
```

```
SOESRV =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = srv-scan)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = soesrv.localdomain)
    )
  )
```

5. Verify that you can connect to the service using the local naming descriptor that you configured in the previous step. Retrieve the instance name of the node that you are connected to.

You should be connected to `rac1`.

```
sqlplus sys/oracle@soesrv as sysdba

SELECT SYS_CONTEXT('USERENV','SERVICE_NAME') FROM DUAL;
SELECT INSTANCE_NAME FROM V$INSTANCE;
```

6. Crash `rac1` instance by using the `pkill -9 -f ora_pmon_rac1` command.

```
# get the process name from the following command:
ps -ef|grep ora_pmon

# kill the process
pkill -9 -f ora_pmon_rac1
```

7. Wait for a few seconds then check the status of `soesrv` service.

The clusterware should have started the service in `rac2` immediately after killing `rac1` instance. It also restarts `rac1` instance once it detects that the instance has been crashed.

```
srvctl status service -db rac -s soesrv
```

8. Connect to `soesrv` service as `sysdba`. Retrieve the instance name of the node that you are connected to.

You should be connected to `rac2`.

```
sqlplus sys/oracle@soesrv as sysdba
SELECT INSTANCE_NAME FROM V$INSTANCE;
```

9. Relocate the `soesrv` service from `rac2` to `rac1`. Confirm the relocation was successful.

```
srvctl relocate service -db rac -service soesrv -oldinst rac2 -newinst rac1
srvctl status service -db rac -s soesrv
```

Ahmed Baraka
Oracle Database Administrator

B. Collecting Service/Module/Action Performance Statistics

In this section of the practice, you will learn how to collect performance statistics data aggregated by service, module, or action. This is very useful in some performance tuning scenarios. You will create multiple sessions connected to the database via the `soesrv` service. You will then use the performance views to display the statistics on the sessions connected to the database via this service.

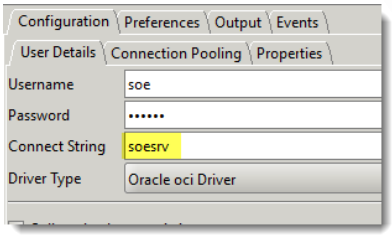
10. In the SQL*Plus session, login to `rac` as `sysdba` and create an AWR snapshot.

```
sqlplus sys/oracle@rac as sysdba
EXEC DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT
```

11. **In the hosting PC**, open the `tnsnames.ora` file in the Oracle client home and configure the connection to `soesrv` service as follows (copy the code from the downloadable file):

```
SOESRV =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.56.91)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = soesrv.localdomain)
    )
  )
```

12. Open the Swingbench and change the connection string to `soesrv`. Test the connection and save the configuration.



13. Start the Benchmark run. Wait for a few minutes.

14. In the SQL*Plus session, check the statistics on the service using `gv$service_stats`.

Having service-wide performance statistics may help to point out a performance issue but usually it is not useful enough to narrow down a performance issue case. You need to know statistics on the module and action level as well.

```
col stat_name format a35

SELECT STAT_NAME, SUM(VALUE)
FROM   GV$SERVICE_STATS
WHERE  SERVICE_NAME = 'soesrv'
GROUP BY STAT_NAME
ORDER BY STAT_NAME;
```

15. Check the performance statistics on the combination of service/module/action for the service. These statistics can be retrieved using `GV$SERV_MOD_ACT_STATS`.

You will observe that the view returns no row. This is because gathering statistics for a combination of service, module and action has not been enabled for any service/module.

```
SELECT * FROM GV$SERV_MOD_ACT_STATS;
```

16. Enable gathering statistics of service name/module/action combination for the modules '`Browse Products`' and '`Browse and Update Orders`'. Those modules are defined by the Swingbench sessions.

```
# service name/module name are case-sensitive
EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE('soesrv', 'Browse Products',
DBMS_MONITOR.ALL_ACTIONS);

EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE('soesrv', 'Browse and Update Orders',
DBMS_MONITOR.ALL_ACTIONS);

# verify:
col aggregation_type format a16
col qualifier_id1 format a18
col qualifier_id2 format a13

SELECT AGGREGATION_TYPE, QUALIFIER_ID1, QUALIFIER_ID2
FROM   DBA_ENABLED_AGGREGATIONS;
```

17. Display the service/module statistics.

The statistics retrieved for a module/action can assist in diagnosis performance issues. This is especially handy in environments where users use the same connections pool to connect to the database.

Note: the `VALUE` in the query below returns zeros for all the rows. Oracle support reported that this is bug number 5060220. They provided some workarounds, but none of them worked for me.

```
col aggregation_type format a15
col stat_name format a32
col module format a25
col action format a5

SELECT MODULE,ACTION,STAT_NAME,VALUE
FROM   GV$SERV_MOD_ACT_STATS
WHERE  SERVICE_NAME='soesrv'
ORDER  BY MODULE,STAT_NAME;
```

18. Disable the statistics gathering for the service/module that you enabled.

```
EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_DISABLE('soesrv', 'Browse Products',
DBMS_MONITOR.ALL_ACTIONS);

EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_DISABLE('soesrv', 'Browse and Update
Orders', DBMS_MONITOR.ALL_ACTIONS);
```

Note: As shown in the previous practice, beside the performance views, you can use the EM Express to display the performance data categorized by the module/action:

- Login to EM Express | **Performance** menu | **Performance Hub** | **Active Sessions** panel | **Services**
- **Activity** tab | click on **the drop down list** | **Top Dimensions** | **Module or Action**

19. Stop and exit the Swingbench.

20. Delete the archive log files.

Ahmed Baraka
Oracle Database Administrator

Summary

Applications should connect to the database via dynamic services. Dynamic database services can work on specific instances. Statistics gathered on the sessions connected to a service can be grouped by modules and actions.

Ahmed Baraka
Oracle Database Administrator