Managing Dynamic Database Services

By Ahmed Baraka

Objectives

In this lecture you will learn how to perform the following:

- Describe the benefits about database services
- Create, start, stop, enable and disable database services
- Modify service configuration
- Relocate services
- Enable and disable parallel operations in services
- Enable statistics aggregation

About Oracle Dynamic Database Services

- Clients should connect to Oracle RAC using services
- Services are defined of applications, workloads, or modules
- Using Service benefits:
  - Integration with Resource Manager: control resource distribution on the services
  - Load balancing: control how the service sessions should be distributed on the instances
  - Tight integration with clusterware: resource profile automatically created:
    - How Oracle Clusterware should manage the service
    - Define service dependencies

About Oracle Dynamic Database Services (cont)

- AWR reports and OEM provide performance metric data for services: can be aggregated by module/action
- Multiple terms:
  - dynamic database service
  - database service
  - service connection

Default Service Connections

- Services created by default:
  - **`DB_UNIQUE_NAME or DB_NAME`**
  - **`PDB_NAME` (in a CDB)**
- Additionally, the database supports two internal services:
  - **SYS$USERS** is the default service for user sessions that are not associated with any application service.
  - **SYS$BACKGROUND** is used by background processes only.

Administering Services

- In Oracle RAC, create them using srvctl or Enterprise Manager, but do not use DBMS_SERVICE
- Service administration tasks include:
  - Create and delete a service
  - Check the status and configuration of a service
  - Start or stop a service
  - Enable or disable a service
  - Relocate a service to a different instance
  - Modify a service attribute
  - Map a service to a consumer group

Service Attributes

- Service name
- Service management policy: **`AUTOMATIC, MANUAL`**
- Instance preference | Server pool assignment
- Connection load balancing goal
- Load balancing advisory goal for run-time connection
- TAF settings
- Database role for a service

Creating, Starting, and Stopping Services

- To create a service called hrsrv with preferred instance rac1 and an available instance rac2:

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

- To start the service:

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

- To stop the service:

```
srvctl stop service -db rac -s hrsrv
```

Client Side Configuration Example to Connect to a Service

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

Enabling and Disabling Services

- To enable/disable a service:

```
srvctl enable service -db rac -service hrsrv
srvctl disable service -db rac -service hrsrv
```

- To enable/disable a service in an instance:

```
srvctl enable service -db rac -service hrsrv –instance rac1
srvctl disable service -db rac -service hrsrv –instance rac1
```

Obtaining Information about Services

- To know the status of all services in a database:

```
srvctl status service -db rac
```

- To know the status of a specific service in a database:

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

- To obtain the configuration information of a service:

```
srvctl config service -db rac -service hrsrv
```

Modifying the Configuration of Services

- Set an available instance as a preferred instance:

```
srvctl modify service -db rac -s hrsrv -instance rac2
 -preferred
```

Relocating Services

- Relocate a service from one instance to another:

```
srvctl relocate service –db rac –service hrsrv -oldinst rac1
 –newinst rac2 [-force]
```

Parallel Operations in Services

- By default, Oracle RAC may decide to execute a SQL statement using more than one instance (in parallel)
- Parallel execution introduce heavy traffic in the interconnect
- You can control the parallel execution in Oracle RAC using the parameter **PARALLEL_FORCE_LOCAL**
- Services can be used to limit the number of instances that participate in a parallel SQL operation

Gathering Performance Statistics by Service in AWR

- Service-level statistics gathered automatically by AWR
- Further granularity (Statistics Aggregation) can be enabled:
  - Service/Module
  - Service/Module/Action
- Statistics aggregation settings are persistent across instance restart.

Enabling Statistics Aggregation

- Monitoring all actions in a module:

```
DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(SERVICE_NAME =>
'HRSRV', MODULE_NAME=> 'PAYROLL', ACTION_NAME => NULL);
```

- Monitoring specific action in a module:

```
EXECUTE DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(SERVICE_NAME
=> 'HRSRV', MODULE_NAME=> 'PAYROLL',
 ACTION_NAME => 'EXCEPTIONS PAY');
```

- **DBA_ENABLED_AGGREGATIONS** view to verify that you have enabled monitoring for application modules and actions

Obtaining Information about Services Performance

V$SERVICE_STATS

V$SERVICE_EVENT

V$SERVICE_WAIT_CLASS

V$SERVICEMETRIC

$SERVICEMETRIC_HISTORY

V$SERV_MOD_ACT_STATS

DBA_ENABLED_AGGREGATIONS

DBA_ENABLED_TRACES

Enabling Tracing Aggregation

- Enable tracing for all actions in a module:

```
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE(
   SERVICE_NAME => 'HRSRV',
   MODULE_NAME  => 'PAYROLL',
   ACTION_NAME  => DBMS_MONITOR.ALL_ACTIONS,
   WAITS        => TRUE,
   BINDS        => FALSE,
   INSTANCE_NAME=> NULL);
```

- Use trcsess tool to collect generated traces it into a single file
- Disable the tracing once the required data is obtained

Summary

In this lecture you should learnt how to perform the following:

- Describe the benefits about database services
- Create, start, stop, enable and disable database services
- Modify service configuration
- Relocate services
- Enable and disable parallel operations in services
- Enable statistics aggregation