Practice

Managing Resumable Space Allocation

Practice Target

In this practice, you will implement the steps that are normally involved in managing resumable space allocation.

Practice Assumptions

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

Managing Resumable Space Allocation

In the following steps, you will implement the steps that are normally involved in managing resumable space allocation.

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

First, we would like to check if the user `HR` is granted the `RESUMABLE` system privilege. Without this privilege, the user is unable to enable the resumable space allocation in its sessions.

2. Connect to `PDB1` as `SYSTEM`, then verify that `HR` has the `RESUMABLE` system privilege.

`HR` is not granted the `RESUMABLE` privilege.

```
sqlplus system/ABcd##1234@//srv1/pdb1.localdomain

SELECT 'Yes' FROM DBA_SYS_PRIVS WHERE GRANTEE='HR' AND PRIVILEGE='RESUMABLE';
```

3. Grant the `RESUMABLE` privilege to `HR`.

```
GRANT RESUMABLE TO HR;

# verify:
SELECT 'Yes' FROM DBA_SYS_PRIVS WHERE GRANTEE='HR' AND PRIVILEGE='RESUMABLE';
```

Second, let's examine the resumable timeout set in the system.

4. Retrieve the value of the `RESUMABLE_TIMEOUT` parameter.

The command shows that the parameter value is zero, which means by default the users cannot enable the resumable space allocation. We would like to keep the parameter value at the system level unchanged and allow the users who want to enable the resumable space feature to set their timeout values.

```
show parameter RESUMABLE_TIMEOUT
```

5. Create a non-expandable tablespace with a total size of 10M. Grant unlimited quota on the tablespace to `HR`.

```
CREATE TABLESPACE RS DATAFILE SIZE 10M AUTOEXTEND OFF;

ALTER USER HR QUOTA UNLIMITED ON RS;
```

6. Login as `HR` to `PDB1`, then run the code that follows to insert large random data into the tablespace.

The code should return the following error because the tablespace free space is not enough to accommodate the required data growth. This is the normal behavior when the resumable space allocation is not enabled.

```
ORA-01653: unable to extend table SOE.TEST by xxx in tablespace RS
```

```
conn hr/ABcd##1234@//srv1/pdb1.localdomain

CREATE TABLE TEST(A CHAR(250)) TABLESPACE RS;

BEGIN
 FOR I IN 1..100000 LOOP
  INSERT INTO TEST (A) VALUES ( DBMS_RANDOM.STRING('U',250));
  IF MOD(I,100) =0 THEN
   COMMIT;
  END IF;
 END LOOP;
 COMMIT;
END;
/
```

7. Enable the resumable space allocation and set the resumable timeout to 8000 seconds.

To enable the resumable space allocation, the user must enable the resumable feature **and** set the resumable timeout.

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 8000;
```

8. Run the loading code again then go to the next step.

The code should hang because the statement is suspended. In real life scenarios, the DBA or the user would be notified by email or SMS about this suspended session. We might configure the required notification mechanism using the `AFTER SUSPEND` trigger.

```
BEGIN
 FOR I IN 1..100000 LOOP
  INSERT INTO TEST (A) VALUES ( DBMS_RANDOM.STRING('U',250));
  IF MOD(I,100) =0 THEN
   COMMIT;
  END IF;
 END LOOP;
 COMMIT;
END;
/
```

9. Open SQL Developer and connect to `PDB1` as `SYSTEM`.

This is the session that the DBA should use to look into the issue after receiving the notification.

10. Retrieve information about the session with the suspended statement. Examine the information retrieved by the view.

```
SELECT * FROM DBA_RESUMABLE;
```

11. As a resolution, add a datafile to the tablespace.

```
ALTER TABLESPACE RS ADD DATAFILE SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 1G;
```

12. Go to the other session and wait for its code to finish execution.
You can query the view `DBA_RESUMABLE` to watch the progress of the statement execution.

Cleanup

13. Drop the `TEST` table. Exit SQL*Plus.

```
DROP TABLE TEST PURGE;
exit
```

14. In SQL Developer, drop the testing tablespace with its datafiles. Exit SQL Developer.

```
DROP TABLESPACE RS INCLUDING CONTENTS AND DATAFILES;
```

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

Summary

- With resumable space allocation feature, operations that produces data and could not continue due to free space issues would suspend instead of raising error. The DBA would then resolve the issue and the suspended statement would continue its operation as normal.