In Oracle Database 19c, \*\*Service Relocation\*\* refers to moving an Oracle service from one instance to another within a Real Application Cluster (RAC) environment. This is useful for load balancing, maintenance, or reducing resource contention across instances in the RAC. The service relocation operation can be performed without disrupting users connected to other services in the cluster.

There are two main ways to relocate a service in Oracle 19c:

1. Using \*\*SRVCTL\*\* (Server Control Utility)

2. Using \*\*DBMS\_SERVICE\*\* package

1. \*\*Using SRVCTL to Relocate a Service\*\*

The `srvctl relocate service` command is used to relocate services between instances in a RAC environment.

#### Syntax:

```bash

srvctl relocate service -db <db\_name> -service <service\_name> -oldinst <source\_instance> -newinst <target\_instance>

```

#### Parameters:

- `-db <db\_name>`: Name of the database.

- `-service <service\_name>`: The service name to be relocated.

- `-oldinst <source\_instance>`: The source instance where the service is currently running.

- `-newinst <target\_instance>`: The target instance to which the service will be relocated.

#### Example:

Relocate the service `my\_service` from instance `instance1` to `instance2` for a database `mydb`:

```bash

srvctl relocate service -db mydb -service my\_service -oldinst instance1 -newinst instance2

```

### 2. \*\*Using DBMS\_SERVICE Package\*\*

You can also relocate a service using the `DBMS\_SERVICE` package, which is particularly useful when you want more control over the relocation process or when scripting relocation as part of a PL/SQL procedure.

#### Example:

```sql

BEGIN

DBMS\_SERVICE.stop\_service('my\_service', DBMS\_SERVICE.SERVICE\_ALL\_INSTANCES);

DBMS\_SERVICE.start\_service('my\_service', 'instance2');

END;

/

```

In this example:

- The service `my\_service` is stopped on all instances.

- The service is then started on `instance2`.

### Key Points:

- \*\*Graceful Relocation\*\*: The `srvctl relocate service` command moves the service to the new instance while allowing in-flight transactions to complete, which ensures minimal disruption.

- \*\*Load Balancing\*\*: Service relocation can help distribute workloads across instances more evenly in a RAC setup.

- \*\*Maintenance\*\*: It allows for instance maintenance without service interruption by moving services off the instance before any planned downtime.

Let me know if you'd like further details on any specific use case!