Using a split mirror as a standby database 7.4 9.2)

Use the following procedure to create a split mirror of a database for use as a standby database outside of a Db2® pureScale® environment. If a failure occurs on the primary database and it becomes inaccessible, you can use the standby database to take over for the primary database.

About this task 7.4 9.2)

If the primary database was configured for log archiving, the standby database will share the same log archiving configuration. If the log archiving destination is accessible to the standby database, the standby database will automatically retrieve log files from it during rollforward operations. However, once the database is brought out of the rollforward pending state, the standby database will attempt to archive log files to the same location used by the primary database. While the standby database will initially use a different log chain from the primary database, the primary database could eventually use the same log chain value as the standby database. This could cause the primary database to archive log files on top of the log files archived by the standby database, or vice versa , and can affect the recoverability of both databases. You should change the log archiving destination for the standby database to be different from that of the primary database to avoid recoverability issues.

Procedure

To use a split mirror as a standby database:

1. Connect to the primary database using the following command:

```
db2 connect to db_name
```

2. Suspend the I/O write operations on the primary database using the following command:

```
   db2 set write suspend for database
```

> **Note:** While the database is in suspended state, you should not be running other utilities or tools. You should only be making a copy of the database. You can optionally use the FLUSH BUFFERPOOLS ALL statement before issuing **SET WRITE** SUSPEND to minimize the recovery time of the standby database.

3. Create one or multiple split mirrors from the primary database using appropriate operating system-level and storage-level commands.

> **Note:**
> - Ensure that you copy the entire database directory, including the volume directory. You must also copy the log directory and any container directories that exist outside the database directory. To gather this information, refer to the DBPATHS administrative view, which shows all the files and directories of the database that need to be split.
> - If you specified the EXCLUDE LOGS with the **SET WRITE** command, do not include the log files in the copy.

4. Resume the I/O write operations on the primary database using the following command:

```
   db2 set write resume for database
```

5. Catalog the mirrored database on the secondary system.

> **Note:** By default, a mirrored database cannot exist on the same system as the primary database. It must be located on a secondary system that has the same directory structure and uses the same instance name as the primary database. If the mirrored database must exist on the same system as the primary database, you can use the **db2relocatedb** utility or the RELOCATE USING option of the **db2inidb** command to accomplish this.

6. Start the database instance on the secondary system using the following command:

```
   db2start
```

7. Initialize the mirrored database on the secondary system by placing it in rollforward pending state using the following command:

```
   db2inidb <database_alias> as standby
```

If required, specify the RELOCATE USING option of the **db2inidb** command to relocate the standby database:

```
   db2inidb <database_alias> as standby relocate using relocatedbcfg.txt
```

where the `relocatedbcfg.txt` file contains the information required to relocate the database.

> **Note:** You can take a full database backup using the split mirror if you have DMS table spaces (database managed space) or automatic storage table spaces. Taking a backup using the split mirror reduces the overhead of taking a backup on the production database. Such backups are considered to be online backups and will contain in-flight transactions, but you cannot include log files from the standby database. When such a backup is restored, you must rollforward to at least the end of the backup before you can issue a **`ROLLFORWARD`** command with the STOP option. Because the backup will not contain any log files, the log files from the primary database that were in use at the time the **`SET WRITE`** SUSPEND command was issued must be available or the rollforward operation will not be able to reach the end of the backup.

8. Make the archived log files from the primary database available to the standby database either by configuring the log archiving parameters on the standby database or by shipping logs to the standby database.
9. Rollforward the database to the end of the logs or to a point-in-time.
10. Continue retrieving log files and rollforwarding the database through the logs until you reach the end of the logs or the point-in-time required for the standby database.
11. Bring the standby database online by issuing the **`ROLLFORWARD`** command with the STOP option specified.

> **Note:**
>
> - The logs from the primary database cannot be applied to the mirrored database after it has been taken out of rollforward pending state.
> - If the primary database was configured for log archiving, the standby database will share the same log archiving configuration. If the log archiving destination is accessible to the standby database, the standby database will automatically retrieve log files from it while rollforward is being performed. However, once the database is brought out of rollforward pending state, the standby database will attempt to archive log files to the same location used by the primary database. Although the standby database will initially use a different log chain from the primary database, there is nothing to prevent the primary database from eventually using the same log chain value as the standby database. This may cause the primary database to archive log files on top of the log files archived by the standby database, or vice versa. This could affect the recoverability of both databases. You should change the log archiving destination for the standby database to be different from that of the primary database to avoid these issues.

Parent topic:

→ High availability through suspended I/O and online split mirror support

Related reference

→ db2inidb - Initialize a mirrored database command

→ db2relocatedb - Relocate database command

→ DBPATHS administrative view and ADMIN_LIST_DB_PATHS table function - Retrieve database paths

→ SET WRITE command

Was this topic helpful? Yes No