Exploiting EMC Timefinder and Oracle Recovery Manager

By

Nitin Vengurlekar -Advisory Support Consultant, Oracle
Steve Maestas – Technical Architect, EMC
Stephan Haisley- Advisory Support Engineer, Oracle

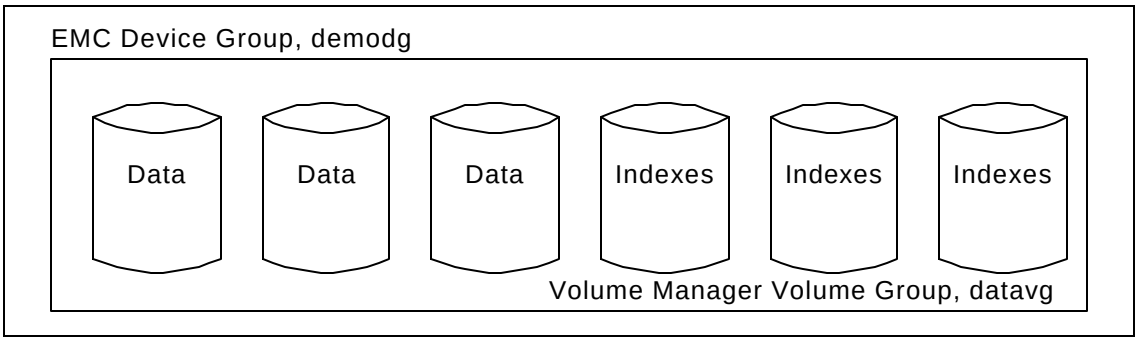
Abstract

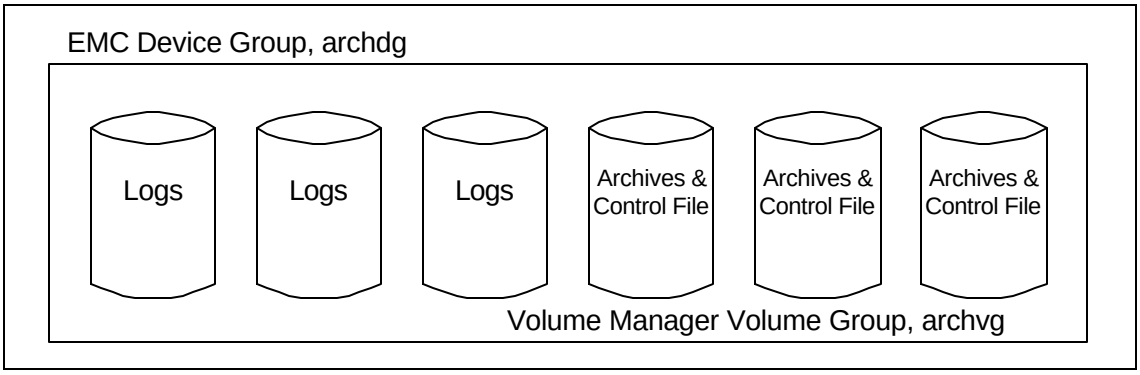
A method for backup and restore of databases using Oracle Recovery Manager (RMAN), EMC Symmetrix and Timefinder is described. The backups execute on a separate host computer, which is typically dedicated for this purpose. When implemented correctly, the method realizes substantial reduction of performance degradation in the production database when creating backups, and provides the opportunity for fast recovery.
The methods used here provides the flexibility necessary to allow for various types of recovery, which includes full, datafile, tablespace, and logical. Full database recovery is described in detail, followed by a brief description of datafile, tablespace, and logical recovery. This paper does not address Oracle RMAN Proxy Copy.[1]

[1] In addition to the methods described in this white paper, it is also possible to leverage Oracle RMAN Proxy Copy. Oracle RMAN Proxy Copy feature is specifically designed to allow RMAN to manage splitting mirror backup and restore procedures. The prerequisites require that a backup product is deployed that supports the proxy copy interface with EMC TimeFinder. The EMC Data Manager is one such product that provides this capability. This paper does not address proxy copy or EDM, for more information on these products please visit the EMC or Oracle web site.

Assumptions:

- Both hosts are connected to the same Symmetrix, where the Oracle volumes are kept.
- An RMAN recovery catalog is used.
- If using a host logical volume manager, then the LVM must be configured identically on both nodes.
- The backup host has Net8 connectivity to the Recovery Catalog database.
- If the backups are output to disk, then the disk must be accessible to the production host.
- The backup host should have a copy of init.ora and a password file (if one exists).
- The init.ora file on the backup host must point to the appropriate control file.
- The Production controlfile location(s) are different from than that of datafiles.
- There are two groupings of disks/LUNS that must occur. One is at the Host's Logical Volume Manager level. This type of group is typically called a Volume Group or Disk Group. The other grouping of LUNS is specific to EMC's TimeFinder and is called Device Group. If using an LVM, there should be at least two disk groups; one containing all datafiles, and another containing the archive logs and control files. Following the same logic for the LUNs, there should be at least two EMC Device Groups, one for the datafiles and the other for archive logs and control files. See illustration below.





Concepts

EMC Timefinder provides a specialized type of disk volume mirror called a Business Continuance Volume (BCV). A BCV is a mirror image of an associated primary disk volume. The mirror can be split, allowing the BCV to be accessed separately. An Oracle instance running on the backup host thus sees a point-in-time frozen image of the production database. On the backup host computer, the split-off BCV has the same EMC hardware address but may appear on a different SCSI address than on the production host computer. The backup host can now use this copy of the Oracle database for backups, testing, and/or development. RMAN can use this copy to perform the backup. By utilizing TimeFinder technology, the performance impact on production during the backup window is minimized. In addition, there are recovery benefits that can be realized by proper database layout and TimeFinder capabilities.

Recovery Manager is Oracle's utility to manage the backup, and more importantly the recovery, of the database. It eliminates operational complexity while providing superior performance and availability of the database. Recovery Manager debuted with Oracle8 to provide DBAs an integrated backup and recovery solution. Recovery Manager determines the most efficient method of executing the requested backup, restore, or recovery operation, and then executes these operations in concert with the Oracle database server. Recovery Manager and the server automatically identify modifications to the structure of the database and dynamically adjust the required operation to adapt to the changes.

Procedures and Steps

The following sections describe the procedures required to backup an Oracle database using RMAN and EMC's BCV Technology.

In these examples, "demodg" is the EMC Device Group that logically associates the production datafiles and BCV copies. The "archdg" EMC device group logically associates the production archive logs, controlfile, and the BCV copy. The "demovg" is the Veritas (LVM) Disk Group that contains the Oracle datafiles. These datafiles are contained in four mount points, "u01-u04". The "archvg" disk group holds the archive logs, redo logs, and control files. This archvg Veritas Disk Group consists of one mount point, "arch" and the control files exist under the "/arch/control_files" directory.

The corresponding commands are specified in the illustration box. Note, for each step there may be two methods that are shown. The first command will show Oracle specific commands, and must be done through a database connection with "sysdba" privileges. The second method may show an EMC command that automates potential scripting. All EMC and LVM commands need to be run as root. Lastly, to take advantage of the EMC commands, the proper environmental variables must be set.[2]

Throughout the sections, two hosts are illustrated: a *Production host*, which runs the production application and houses the production datafiles; and the *Backup host*, which will be used to perform the RMAN backups, and possibly used for logical, or database restores. The backup and restore procedures in the following sections are illustrated in chronological order and broken down by host.

[2] The PATH, SYMCLI_RDB_CONNECT, SYMCLI_TYPE, must be set

Backup Procedure Methods

Production Host

1. Establish the EMC BCV Device Groups with their Production copies.

```
symmir –g demodg establish –noprompt

symmir –g archdg establish –noprompt
```

2. Force a log switch. While not strictly necessary, this will flush the redo buffers and help to limit the number of logs that must be applied to the backups to eliminate their fuzziness.

```
SQL>alter system archive log current;
                or
symioctl archive log
```

3. Place the required tablespaces in backup mode[3].

```
SQL> alter tablespace TBSNAME begin backup;

NOTE:  The above command must be done for every required tablespace, and is typically scripted
to dynamically account for changes to the physical structure of database since the last backup.
        or
symioctl begin backup -noprompt

The above EMC command will dynamically build a list of tablespaces to place in backup mode.
```

4. Split the BCV volumes containing the Oracle datafiles.

```
symmir –g demodg split -instant –noprompt
```

5. Take the database out of hot backup mode.

```
SQL> alter tablespace TBSNAME end backup;

NOTE:  The above command must be done for every required tablespace, and is typically scripted
to dynamically account for changes to the physical structure of database since the last backup.
        or
symioctl end backup -noprompt
```

[3] If instant splits are being used, then it is no longer necessary to use 'alter system suspend/resume'

The above EMC command will dynamically build a list of tablespaces to take out of backup mode.

6. Force the current log to be archived. This ensures that all redo needed to make the backed up files consistent after they are restored has been archived.

```
SQL>alter system archive log current;
        or
symioctl archive log
```

7. Resynchronize the RMAN catalog with the production database. This adds the most recent archive log data to the recovery catalog.

Once connected to the RMAN catalog and the target production database, issue the command:
RMAN> resync catalog;

8. Create two copies of a backup controlfile. One copy of the control file, called "control_start" will be used to start the database in mount mode on the backup server. The second copy, named "control_backup", will be part of the backup set backed up by RMAN. The reason two copies are used is because, once the database is started (on Backup host) in mount mode, the SCN is updated in the control file, and thus it is no longer a consistent copy of the control file. Therefore, it is necessary to have another copy, deemed a pristine version, that can be backed up.

   The init.ora file's "CONTROL_FILES" parameter on the backup server must point to the "control_start" copy of the control file.
   (CONTROL_FILES=/arch/control_files/control_start)

```
RMAN> run {
 allocate channel foo type disk;
 copy current controlfile to '/arch/control_files/control_bakup';
 copy current controlfile to '/arch/control_files/control_bakup';
   }
```

9. Split the BCV Volume containing the archive logs. The backing up the archive logs can occur multiple times a day as required.

```
symmir –g archdg split –noprompt
```

[4] These control files must not exist prior to issuing the alter database command, or it will fail.

Backup Host

10. If you are using an LVM, then all the volume group information (which includes disk layout metadata) from the Production host needs to be imported into the Backup host. Use the LVM utility to import the Logical Volume Manager Volume Groups. In our example, there should be two Volume Groups that exist on Production; one for the datafiles and one for archive logs. Therefore, these two Volume Groups need to be imported on the Backup host. This step is required each time backups are performed.

```
vxdg –C  import demovg-id
vxvol –g demovg startall        ← Note, these Veritas commands may need
                                  to be reviewed, and tested, to  make sure
                                  they fit your environment.  Addtionally,
                                  these commands will differ between LVMs
vxdg –C import archvg-id
vxvol –g archvg startall
```

11. Perform a filesystem check on the all the LUNS that make up the Veritas Volume Groups, “demovg” and “archvg”.

```
fsck –y /dev/dsk/c1t2d0s2
fsck –y /dev/dsk/c1t2d1s2
fsck –y /dev/dsk/c1t2d2s2
fsck –y /dev/dsk/c1t2d3s2
```

12. Mount all filesystems of the Volume Groups that were imported. (if not done automatically from the import command from step 10). At this point, the production files should all be visible on the backup host, and should have all the same file names that they had on the production host. ← This step is not required if using raw devices.

```
mount  /u01
mount /u02
mount /u03
mount /u04
mount  /arch

Note:  If entries do not exist in the /etc/vfstab or /etc/filesystems, you must issue the appropriate
command to mount the filesystems .
```

13. Startup and mount the database. Mount the database using the backup control file that was created in the backup procedure called "control_start". Note, the init.ora file should have the CONTROL_FILES variable point to the /arch/control_files/control_start file.

```
SQL>startup mount
```

14. Backup the database using RMAN, including the archive log files and backup controlfile from Step #8.

The "backup controlfile" must be backed up, and then the database files must be backed up, along with the archive log files. Once connected to the recovery catalog and the target database on the Backup host (which is now mounted), issue the following commands[5]:

```
run { allocate channel t1 type 'SBT_TAPE';
                  #Can allocate several channels if using several
                  # tape drives backup
backup
format 'ctl_%d_%s__%p_%t'
controlfilecopy '/arch/control_files/control_bakup ';
  backup
      full
      format 'db_%d_%s_%p_%t'
      (database);
   backup
      format 'al_%d_%s_%p_%t'
(archivelog all);
  release channel t1;
}
```

Note, if required, incremental RMAN backups can also be employed.

[5] The script will backup the controlfile copy, the database (full), and then archive log files using a time constraint.

Post Backup Procedures

There are two options for post backup procedures.

Option 1

Once the backup has been completed it is recommended to leave the datafile's Volume Group imported and mounted on the backup host. Leaving the volumes mounted will allow us to take advantage of the TimeFinder and RMAN technology, if database recovery is required after this backup was created. This will be explained in the Recovery section of this document.

However, the filesystems that make up the archive Volume Group should be unmounted, and the Volume Group should be deported. The BCV Device Group for the archives will be then re-established with the production copy.

15. Unmount the archive log's filesystems.

```
umount /arch
```

16. Deport the "archive log" Volume Group.

```
deport –g archvg
```

goto Step 18

Option 2

For security precautions, it may be necessary to umount the datafile's Volume Group (but still leaving it imported) on the backup host. This will prevent accidental updates to this database. However, the filesystems that make up the archive Volume Group should be unmounted, and the Volume Group should be deported. The BCV Device Group for the archives will be then re-established with the production copy.

15. Shut down the instance on the backup host.

```
SQL> shutdown immediate
```

16. Unmount the archive log and datafile filesystems.

```
umount /arch
umount /u01
umount /u02
umount /u03
umount /u04
```

17. Deport the "archive log" Volume Group.

deport –g archvg

goto step 18.

Production Host

18. Re-establish the "archive log" EMC Device Group BCV to the production system. This will incrementally synchronize the BCV with the production archive volumes. The incremental process will only copy the tracks that have changed since the last time the BCV was split. When this step is completed, the BCVs are considered to be in a synchronized and established state. Leave the BCV in the established state until the next backup window, or until a Recovery of the database is required.

symmir –g archdg establish -noprompt ← to re-establish EMC archdg disk group

Restore Procedure Methods:

There are four methods of restore that where considered in this white paper. The methods are full, tablespace, datafile, and logical recovery. This white paper goes into detail on how to perform full database restores. Tablespace, datafile, and logical recovery are covered in Appendix I.

Full database recovery implies that all the datafiles will be restored. For full database media recovery, restore can be done either on the Production Host or on the Backup Host. Production host recovery means that the files will be restored from the BCVs to the production devices, and the Oracle recovery will be performed on the production host. Backup host recovery means that the Oracle recovery will be preformed on the backup host against the BCV copy of the database. Then the recovered files will be restored from the BCV to the production devices.

Method One – Restore on Production Host
If recovery time is critical, it is recommended that the recovery be done on the Production Host. This method will provide minimal down time, while protecting the "good" copy (BCV) of the database. This technique is very fast since only the changed tracks on the LUNs that make up the database will be restored back. In addition, the changes are being updated through the architecture of the Symmetrix and not over the network and/or from tape. Only the required archive logs need to be applied for recovery. It is also possible to take advantage of the "instant" restore capability of TimeFinder by performing a protected restore[6]. This type of restore will write protect the BCV copy, thus allowing archives to be applied immediately after the BCV restore command is issued. If corruption is reintroduced or a mistake is made during the recovery procedure, the BCV can once again be used to perform a quick restore of the Production Host's database. If this option is not available, it is recommended that the restore process be fully completed, and the BCV copy is split prior to applying Oracle logs; i.e. initiating recovery. This will ensure that a good online copy of the database is available.

Method two – Restore on Backup Host
The second method is used to recover the database on the backup host. Then the EMC TimeFinder software is used to restore the "recovered" database from the Backup Host to the Production Host. This method is beneficial for sites that require recovery be tested and validated, before restoring to production. This method still provides a fast recovery procedure, but requires additional steps[7].

Both of these methods assume that the backup set that will be used in the recovery exists on the BCVs. If an older backup is necessary for recovery, then the appropriate files will have to be restored from tape.

[6] Protected restore is available at 5x68 Enginuity Code
[7] However, if a mistake is made during the recovery procedures, and the BCV copy becomes corrupted, then the Oracle datafiles must be restored from tape. This may potentially increase recovery time.

This white paper does not go into detail on datafile, tablespace, or logical recovery. There is a brief description of each method in the “Datafile, Tablespace, and Logical Recovery” section.

Recover Database on Production Host Methods:

Production Host

Below are some examples of restore and recovery commands that can be used to restore and recover the database on the production host. Mount points /u01 through /u04 contain the datafiles and mount point /arch houses the archive logs.

This procedure is based on the following assumptions:

- The archdb device group has established BCVs (see step 18 of the backup procedure)
- The filesystems with archived logs is not mounted on the backup host (see step 16 of the backup procedure).
- The demodg device group does not have established BCVs.
- The filesystems with datafiles are still mounted on the backup host before initiating these steps.
- All controls files are lost and need to be restored. In the case that all controlfiles are not lost, see Appendix.

1. Shutdown the database

```
SQL>shutdown immediate
```

2. Unmount all filesystems for datafiles on Production host.

```
umount /u01
umount /u02
umount /u03
umount /u04
```

Unmount all filesystems for datafiles on Backup Host

```
umount /u01
umount /u02
umount /u03
umount /u04
```

3. Export the datafile's Logical Volume Manager Volume Group on Backup Host.

```
export –g datavg
```

Production Host

4. Perform a protected BCV restore to the Production volumes (do not do an establish, as this will overwrite the BCV).

```
symmir –g datadg –protect –noprompt restore
```

If protected restore is not available because of the level of EMC Symmetrix Enginuity Code[8] then perform the following:
Wait until the BCV has completely restored the Production Database, and split the BCV, and then start the recovery of the database.

5. Mount the filesystems.

```
mount /u01
mount /u02
mount /u03
mount /u04
```

6. Startup nomount the database on the production host.

```
SQL> startup nomount;
```

7. Restore controlfile and mount the database

```
RMAN> run {
   allocate channel t1 type 'SBT_TAPE';
   restore controlfile;
   alter database mount;
   release channel t1;
}
```

8. For Oracle 8i, in case that archived logs are not lost and they are still on disk, catalog all archived logs which are created after last backup (or resync):

```
RMAN> run {
  allocate channel for maintenance type disk;
  catalog archivelog '/arch/arch_1_123.arc';
  catalog archivelog '/arch/arch_1_124.arc';
}
```

Oracle 9i will automatically locate the archived logs on the disk, so this step is not required for Oracle 9i.

[8] The EMC Symmetrix Enginuity Code is the firmware on the Symmetrix. You must be at a base level of 5x66 Enginuity Code to perform most of these tasks; however, the 5x68 is required for protected restores.

9. Perform complete or point in time recovery with RMAN. If archived logs were lost, this will restore all the archives and apply them. If performing incomplete recovery, then set the *until time* or *until SCN* markers.

```
RMAN> run {
   allocate channel t1 type 'SBT_TAPE';
   set until time '20-oct-01 16:00';
   recover database;
   release channel t1;
}
```

10. After successful recovery. Shutdown the database.

```
SQL> shutdown immediate
```

11. Test and validate the database, run dbverify. This will validate the datafiles.[10]

```
dbv '/oradata/demo8i/system01.dbf'   ← run dbv against all datafiles.
```

12. After you are sure that all files are correctly restored and recovered, you can open the database using the RESETLOGS option. The option RESETLOGS is required because we have restored the controlfile. The open with option RESETLOGS will create a new incarnation of the database, which must be also registered in the RMAN.

```
RMAN> startup mount;
RMAN> sql 'alter database open resetlogs';
RMAN> reset database;
```

It is recommended to take a backup as soon as the database has been restored and opened using RESETLOGS.

13. Follow the backup procedures and perform a backup of the recovered production database.

[9] Db verify is also required in 9i.
[10] This step is required, since Timefinder is used in the restore process, instead of RMAN.

Restore Database on Backup Host Methods:

This procedure is based on the following assumptions:

- The archdb device group has established BCVs (see step 18 of the backup procedure)
- The filesystems with archived logs is not mounted on the backup host (see step 16 of the backup procedure).
- The demodb device group does not have established BCVs.
- The filesystems with datafiles are still mounted on the backup host.
- All controls files are lost and need to be restored. In the case that all controlfiles are not lost, see Appendix.

Production Host

1. Shutdown the database.

```
SQL> shutdown immediate
```

2. Because the assumption that the BCV mirroring is established for archived logs (see step 18 the backup procedure), split off the "archive log" EMC Device Group BCV.

```
symmir –g archdg split -noprompt
```

Backup Host

3. Import the archive log's Logical Volume Manager Volume Group.

```
import –g  archvg
```

4. Mount the filesystems with archived logs:

```
mount /arch
```

5. Startup nomount the database.

```
RMAN> startup nomount
```

6. Restore controlfile and mount the database

```
RMAN> run {
   allocate channel t1 type 'SBT_TAPE';
   restore controlfile;
   alter database mount;
   release channel t1;
}
```

7. For Oracle 8i, in case that archived logs are not lost, catalog all archived logs which are created after last backup (or resync):

```
RMAN> run {
   allocate channel for maintenance type disk;
  catalog archivelog '/arch/arch_1_123.arc';
  catalog archivelog '/ arch/arch_1_124.arc';
}
```

   Oracle 9i will automatically locate the archived logs on the disk, so this step is not required for Oracle 9i.

8. Perform complete or point in recovery within RMAN. Issue recover database (this will restore all the archives and apply them). If performing incomplete recovery, then using RMAN, set until time or until SCN markers.

```
RMAN> run {
   allocate channel t1 type 'SBT_TAPE';

   set until time '20-oct-01 16:00';
   recover database;
   release channel t1;
}
```

9. After successful recovery. Shutdown the database.

```
SQL> shutdown
```

10. Test and validate the database, run dbverify. This will validate the datafiles[12] ←

```
dbv '/oradata/demo8i/system01.dbf'   ← run dbv against all datafiles.
```

[12] This step is required, since Timefinder is used in the restore process, instead of RMAN.

11. After you are sure that all files are correctly restored with EMC and recovered, you can open the database on the backup host using the RESETLOGS option. The new incarnation of the database must be also registered in the RMAN.

```
RMAN> startup mount;
RMAN> sql 'alter database open resetlogs';
RMAN> reset database;
```

It is recommended to take a backup as soon as the database has been restored and opened using RESETLOGS.

12. Unmount all filesystems for datafiles and archives

```
umount /u01
umount /u02
umount /u03
umount /u04
umount /arch
```

Production Host

13. Unmount all filesystems for datafiles and archives.

```
umount /u01
umount /u02
umount /u03
umount /u04
umount /arch
```

14. Perform a BCV restore to the Production volumes (do not do an establish, as this will overwrite the BCV). Optionally, after the BCVs are re-established and the BCV copies are restored to the production copies, the backup procedure can be followed to perform a fast backup of the recovered database.

```
symmir –g demodg restore -noprompt
symmir –g archdg restore –noprompt      ← re-establish to the production system.
```

15. Verify that the BCV restore process has completed.

```
symmir –g demodg query
symmir –g archdg query
```

16. Optionally split the device groups again, so that the BCVs can be kept as a pristine backup copy of the newly restored database.

```
symmir –g demodg split –instant –noprompt
symmir –g archdg split –instant –noprompt
```

17. Mount the filesystems.

```
mount /u01
mount /u02
mount /u03
mount /u04
mount /arch
```

18. Start the database.

```
SQL>startup
```

19. Follow the backup procedures and perform a backup of the recovered production database.

Appendix

I. Datafile, Tablespace, and Logical Recovery Scenarios

The method of data layout of the Oracle database in conjunction with TimeFinder used in this white paper, can still be used to perform datafile, tablespace, and logical recovery. Recovery requirements, depending on the level and time , will dictate which of these methods are viable. The advantage of having the TimeFinder copy is realized in not having to recovery datafiles from tape.

1. Scenario one: Datafile recovery (due to a lost or corrupted datafile) where a "good" version of the datafile(s) exists in the available BCV image. The datafile(s) can be copied to production [using rcp, dd, or ftp] and recovered. This method can potentially save time in the recovery process by not having to restore from tape. This method requires the datafiles/tablespaces are laid out such that they are on their own "disks" or LUNs, and the Volume Manager Volume Groups must be created for each datafile/tablespace. Thus, this method will allow the TimeFinder technology to be utilized in performing fast recovery at a datafile or tablespace level. An alternative method is to utilize EMC InfoMover software that will use "ftp" like commands to move the datafiles across the SAN instead of over the network. Thus for large datafiles/tablespaces, this provides a fast restore method.

2. Scenario two: datafile or tablespace recovery where recovery is prior to previous nights backup, thus BCV image. In this case RMAN may be used to restore/recover the affected datafile(s) to production host from tape. If the recovery time of the database does not meet the Service Level Agreements, multiple BCV Copy images of the same production Oracle database can be kept online.

3. Scenario three: recovery at a logical level, where "bad" data is loaded in the database, or end user error has deleted table(s). It may not be feasible to restore datafiles and apply the archive logs to a point in time. This would imply that data/transactions could be lost in "non-effected" areas of the database. A method to recover from this type of corruption is to take a copy of the production database, mount on a test/backup server, and logically remove the bad data. Once the method is tested and the database copy is validated the SQL statements can be run in production to delete the bad data. Prior to running the SQL code on the production host, it is recommended that a separate BCV copy is synchronized and split.

II. No Control File loss scenario

In the cases where all the control files are not lost, the restoration of the controlfile is not necessary. For cases such as these, follow the recommended procedures described in recovery section, with following exceptions:

- Control files should not be restored.
- Opening the database with resetlogs (and RMAN "reset database") is required only in case of incomplete database recovery.

However, if database recovery is performed on the Backup Host, then the current controlfile [on the Production Host] must be copied to the Backup Host and the "control_files" init.ora parameter must point to this file location. This procedure has same exceptions as above.

III. Archive Log Backup Best Practices

Due to the fact that RMAN is being used on the backup host, the automatic deletion of backed up archive logs should not be used. It is recommended to delete all archive logs on the primary host after they have been backed a predefined number of times. If the requirement is to back up each archived redo log twice, the following sequence could be followed:

1. Backup the archive logs on the backup host.

2. Query the RMAN catalog from SQL*Plus to see which files have been backed up twice:

```
SELECT name, dbinc_key
FROM rc_database_incarnation
ORDER BY 1,2;

SELECT name,  x.bkcount
FROM (SELECT b.name, count(*) bkcount
        FROM rc_backup_redolog a, rc_archived_log b
        WHERE a.DBINC_KEY=&DBInc
        AND b.dbinc_key = a.dbinc_key
        AND a.thread#= b.thread#
        AND a.sequence# = b.sequence#
        ORDER BY name) x
WHERE x.bkcount > 1;
```

The second SQL script will prompt for a DB Incarnation key, which should be selected from the first query results. The list of filenames shown can then be used to delete them from the primary host. These scripts should be run after each successful backup, and could be automated with minor adjustments.

Archive logs may be requiring more frequent backups than datafiles. This may be the case if the system generates so much archive logs that the archive location becomes full. If intermittent archive log backups are required, then follow the steps described in the Backup Section.

IV. Restrictions

- If an LVM is used, then any logical volumes that span different BCVs must ALL be split at the same time. Database corruption may result if they are not all split at the same time.

V. Backup Host Considerations

The Oracle instance on the backup host can be opened to access the database. However, this will require media recovery, since the datafiles are in hot backup mode. If the database is opened on the backup host, then the following restrictions apply:

1. If a backup is to be performed, it must be done *before* opening the database, never after. The reason is that when the database is opened, crash recovery is performed, backing out any transactions that were in progress when the BCVs were split.

2. If the backup database is opened read/write, then it must not be allowed to archive any logs. The reason is that if these logs were to be used by the production database, they would corrupt the production system since these logs do not really belong to the production database. Furthermore, if any modifications are made to the database on the backup host, and the files are then backed up, the backup files will contain modifications that were never made on the production host, which can cause data corruption. If the backup database is in archivelog mode, it should be set to noarchivelog mode before opening it read/write.

3. In Oracle8i, the read-only database feature is available. If the database on the backup host is opened, it should be opened in read-only mode. If opened in read-only mode, it is advised build locally managed temporary tablespaces for temporary segments. This will allow you to run queries, which require sorting or other temporary segment writes while in read-only mode.

4. TNS Issues

When connecting to the target databases and the RMAN catalog, users must consider the following:

1. Always connect to the same recovery catalog, whether running a job on the production or backup host.
   When running jobs on the production host:

   ```
   #rman target system/manager@prod rcvcat rman/rman@rcat
   ```

   When running jobs on the backup host:

   ```
   #rman target system/manager@bkup rcvcat rman/rman@rcat
   ```

2. Make sure you are connecting to the production host as the target when running jobs on the production host

3. Make sure that you are connecting to the backup host as the target when running jobs on the backup host

The following is a sample tnsnames.ora configuration file used in RMAN/BCV integration.

tnsnames.ora

```
# ----------------------------------

PROD =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(PORT=1521)(HOST=PRODHOST) ) )

(CONNECT_DATA =(SERVICE_NAME = PROD) ) )

BKUP =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(PORT=1521)(HOST=BKUPHOST) ) )

(CONNECT_DATA =

(SERVICE_NAME = BKUP) ) )

# Entry for the Recovery Catalog

RCAT =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(PORT=1521)(HOST=DALLAS)) )

(CONNECT_DATA =

(SERVICE_NAME = rcat) ) )
```

Acknowledgments

The authors of this paper would like to acknowledge the following persons for extending their efforts in validating and verifying this methodology:

1. Senad Dizdar, Oracle RMAN Development
2. Steve Wertheimer, Oracle RMAN Development
3. Ken Taylor, EMC Engineering
4. Tammy Bednar, Oracle HA Product Manager