Databases Clone using ACFS
Infrastructure at your Service.

About me

David Hueber
COO
Principal Consultant

+41 79 963 43 68
david.hueber[at]dbi-services.com
Who we are
dbi services

Experts At Your Service
> Over 50 specialists in IT infrastructure
> Certified, experienced, passionate

Based In Switzerland
> 100% self-financed Swiss company
> Over CHF6 mio. turnover

Leading In Infrastructure Services
> More than 120 customers in CH, D, & F
> Over 40 SLAs dbi FlexService contracted

Databases Clone using ACFS
Agenda

1. Introduction
2. ACFS principle
3. ACFS principle
4. Making ACFS based clones
Introduction
What? Why? How?

- Duplication vs Replication
- Requirements for duplication
- Challenges
- Cloning solutions

Databases Clone using ACFS
Introduction
Duplication vs Replication

Replication
- High availability
- Disaster Recovery
- Data consolidation / Business Intelligence
- Real-time or near real-time
- Can be cross
  - Platform
  - RDBMS

Duplication
- Provide a copy of a database
  - RMAN duplicate
- On-demand process

Databases Clone using ACFS
Introduction
Requirements for duplication

Application lifecycle
  > Application development
  > Application validation

Training

Upgrade tests

Troubleshooting
  > Reproduce issues
Introduction

Challenges

Operational

> Not the easiest operation
> Methods (Active vs Backup)
> Backup sharing or copy
> Frequency
> Duplicate workload
> Developer on-demand requirements

Resources

> Duplication duration
> Space consumption
Introduction
Cloning solutions

Snapshots
- Fast generation
- Reduce storage
- Different technologies
  - Commit-On-Write (COW)
  - Redirect-On-Write
  - Incremental snapshots

Available tools
- Delphix
- ACFS
- ZFS snapshots – Snap Management Utility
- NetApp SnapManager for Oracle
- ...and more

Databases Clone using ACFS
ACFS principle
Multi-function file system

What is ACFS?
Commit-On-Write principle
ACFS implementation
Limitations

Databases Clone using ACFS
ACFS principle
What is ACFS?

Automatic Storage Management Cluster File System
- Part of ASM
- Multi-platform
- Based on ADVM – Volume Manager

Multiple features
- Security realms and rules
- Encryption
  - Files
  - Volumes
- Snapshots
- Replication
ACFS principle
What is ACFS?

Snapshots
> Based on Commit-On-Write
> 2 types
  > Read-only (can be converted)
  > Read-write
> Done on File System level
  > Max 256 file systems mount per environment
  > Max $2^{40}$ files per file system
  > 1023 snapshots per file system
> Snapshots of snapshots supported
> Requires 11.2.0.3 ADVM compatibility on Disk Groups
ACFS principle
Commit-On-Write

Databases Clone using ACFS
Principle

- Stores only the metadata
- Read requests to unchanged data go to the original volume

Advantages

- Space efficient
- Creation is overhead-less
- Protection against logical errors

Drawbacks

- CPU and I/O overhead during changes on original data
- No protection against physical errors
ACFS principle
ACFS implementation
ACFS principle
Limitations

Single node
- Oracle Restart does not
  - Support root resource – File systems
  - Load ACFS drivers automatically
  - Support data files on ACFS
- Requires full Clusterware installation

Configuration
- Latest version certified on Oracle Linux 7
  - ACFS drivers not included for OL7 – requires PSU installation
  - ADVM default compatibility still on 11.2.0.0.0
Making ACFS based clones
Easy provisioning

- Provisioning architecture
- Preparing the environment
- Cloning procedure
- Cloning PDBs
Making ACFS based clones
Provisioning architecture

Production environment

Duplicate

Gold image
Snapshot W/R

Gold image

Test / Dev environment

Snapshot W/R

Snapshot W/R

Snapshot W/R

Gold image
Snapshot would clone all databases
Making ACFS based clones
Provisioning architecture

Golden image
- Copy of production environment
  - Usually static
- Generated using RMAN duplicate
  - Can be done during low activity period
- Dynamic images possible with replication
  - Data Guard, Dbvisit, Golden Gate...
- Avoid performance overhead on production database

Snapshot W/R
- Allow changes on snapshot databases
- Golden image in snapshot W/R
  - Workaround for file system level clone
  - Increase snapshot granularity
Making ACFS based clones
Preparing the environment

Single node setup
> Run Grid Infrastructure software only installation
> Run `$GRID_HOME/crs/config/config.sh`
> 2 network interfaces – public & private
> VIP and SCAN

ACFS configuration
> Check ADVM drivers
> Missing on Oracle Linux 7

```
grid@vmoraacfs01:/home/grid/ [+ASM] acfsdriverstate loaded
ACFS-9204: false
grid@vmoraacfs01:/home/grid/ [+ASM] acfsdriverstate installed
ACFS-9204: false
```
Making ACFS based clones
Preparing the environment

ACFS configuration
> Install PSU
> Need to be run as root
> grid and oracle users need access to the patches
> Re-check drivers status

[root@vmoraacfs02 ~]# acfsdriverstate installed
ACFS-9203: true
[root@vmoraacfs02 ~]# acfsdriverstate loaded
ACFS-9203: true
Making ACFS based clones
Preparing the environment

ACFS configuration
> Create volumes & format file system

ASMCMD> volcreate -G DATA -s 700G datavol
ASMCMD> volinfo -G DATA -aDiskgroup
Name: DATA
Volume Name: DATAVOL
Volume Device: /dev/asm/datavol-499
... 
Usage: Mountpath:

grid@vmoraacfs01:/home/grid/  [+ASM] mkfs.acfs /dev/asm/datavol-499
mkfs.acfs: version = 12.1.0.2.0
mkfs.acfs: on-disk version = 39.0
mkfs.acfs: volume = /dev/asm/datavol-499
mkfs.acfs: volume size = 751619276800 ( 700.00 GB )
mkfs.acfs: Format complete.
Making ACFS based clones
Preparing the environment

ACFS configuration
  > Declare the file system to the cluster

```bash
root@vmoraacfs02 ~]# srvctl add filesystem -device /dev/asm/recovg-174 -path /u99/fast_recovery_area/ -user oracle

[root@vmoraacfs02 ~]# df -h
```

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
<th>Used</th>
<th>Avail</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda3</td>
<td>10G</td>
<td>1.8G</td>
<td>8.3G</td>
<td>18%</td>
<td>/</td>
</tr>
<tr>
<td>/dev/mapper/vgorabin-lvorabin</td>
<td>40G</td>
<td>18G</td>
<td>23G</td>
<td>45%</td>
<td>/u01</td>
</tr>
<tr>
<td>/dev/sdal</td>
<td>2.0G</td>
<td>131M</td>
<td>1.9G</td>
<td>7%</td>
<td>/boot</td>
</tr>
<tr>
<td>/dev/asm/redovg-167</td>
<td>9.0G</td>
<td>760M</td>
<td>8.3G</td>
<td>9%</td>
<td>/u01/oradata</td>
</tr>
<tr>
<td>/dev/asm/datavg-385</td>
<td>700G</td>
<td>19G</td>
<td>690G</td>
<td>3%</td>
<td>/u02/oradata</td>
</tr>
<tr>
<td>/dev/asm/recovg-174</td>
<td>98G</td>
<td>307M</td>
<td>98G</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>/u99/fast_recovery_area</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Making ACFS based clones
Cloning procedure

2 cases
> Gold Image database never change between refresh
> Gold Image may change

Fix Gold Image database
> Keep the database in mount status
> Make offline clone
  > Snapshot the database
  > Re-create control file and spfile

Gold Image changing
> Make an online clone
Making ACFS based clones
Cloning procedure

1. Prepare control file and spfile

   SQL> alter database backup controlfile to trace
   as '/tmp/cre_ctl_CLONE.sql';

   SQL> create pfile='/tmp/initCLONE.ora' from spfile;

2. Synchronize the source database

   SQL> alter system checkpoint;
3. **Snapshot the source database**

```sql
SQL> alter database begin backup;
[acfsutil]
[root@vmoraacfs02 ~]# acfsutil snap create -w -p <SourceDB> <CloneDB> /u02/oradataacfsutil
snap create: Snapshot operation is complete.

SQL> alter database end backup;
```

4. **Run a log switch**

```sql
SQL> alter system archive log current;
```
Get snapshot creation time

[root@vmoraacfs02 ~]# acfsutil snap info /u02/oradata/
snapshot name: DBDEV1
snapshot location: /u02/oradata/.ACFS/snaps/DBDEV1
RO snapshot or RW snapshot: RW
parent name: DBREF1
snapshot creation time: Thu Oct 27 10:06:40 2016
number of snapshots: 2
snapshot space usage: 1641201664 ( 1.52 GB )
Cloning procedure

6. Get archivelogs name and completion time

```
SQL> select name, completion_time from v$archived_log
       where sequence >=XX;
```

7. Adjust snapshot path

```
[oracle@vmoraacfs02 ~]# mv
  /u02/oradata/.ACFS/snaps/<CloneDB>/<SourceDB>
  /u02/oradata/.ACFS/snaps/<CloneDB>/<CloneDB>
```

8. Create folders

```
mkdir -p /u01/app/oracle/admin/DBDEV1/adump
mkdir -p /u01/oradata/DBDEV1/onlinelog
mkdir -p /u01/oradata/DBDEV1/controlfile
mkdir -p /u99/fast_recovery_area/DBDEV1/archivelog/YYYY_MM_DD/
```
Making ACFS based clones
Cloning procedure

9. Copy last generated archivelogs

10. Adjust pfile and create spfile
   > Remove `control_files` parameter
   > Replace `<SourceDB>` by `<CloneDB>`
   
   ```sql
   SQL> create spfile '${ORACLE_HOME}/dbs/spfile<CloneDB>.ora'
   from pfile='/tmp/init<CloneDB>.ora';
   ```

11. Copy password file
   
   ```bash
   cp $ORACLE_HOME/dbs/orapw<SourceDB> $ORACLE_HOME/dbs/orapw<CloneDB>
   ```
Create control file for the new database

- Use script from control file trace with RESETLOGS option
- Replace REUSE command by SET

```sql
CREATE CONTROLFILE SET DATABASE "DBDEV1" RESETLOGS ARCHIVELOG MAXLOGFILES 16
...
LOGFILE
  GROUP 1 ('/u01/oradata/DBDEV1/onlineLog/o1_mf_1_d0f3p5b4_.log',
  '/u01/oradata/DBDEV1/onlineLog/o1_mf_1_d0f3p5p9_.log') SIZE 50M BLOCKSIZE 512,
...
DATAFILE
'/u02/oradata/.ACFS/snaps/DBDEV1/DBDEV1/datafile/o1_mf_system_d0f3p7so_.dbf',
...
'CHARACTER SET WE8MSWIN1252;
```
Making ACFS based clones
Cloning procedure

13
Catalog archivelogs
> Optional
> Ease the database recovery

RMAN> catalog start with
'/u99/fast_recovery_area/<CloneDB>/archivelog/YYYY_MM_DD/';

14
Recover database
> Snapshot time
> Last archivelog completion time

SQL> recover automatic database until time '<arc_completion_time>'
using backup controlfile snapshot time '<snap_creation_time>';
Making ACFS based clones
Cloning procedure

15
Open database in resetlogs
SQL> alter database open resetlogs;

16
Re-create TEMP file
> Statement available in controlfile trace

17
Change database DBID
> Optional

```
oracle@vmoraacfs02:/tmp/ [DB1] nid target=/
...
Database has been shutdown, open database with RESETLOGS option. Successfully changed database ID. DBNEWID - Completed successfully.
```
Making ACFS based clones
Cloning procedure

18 Open database in resetlogs

SQL> alter database open resetlogs;

19 Add database in cluster
> Will guarantee auto-start of the resource

srvctl add database -db <CloneDB> -oraclehome $ORACLE_HOME 
-dbtype SINGLE -role PRIMARY 
-spfile $ORACLE_HOME/dbs/spfile<CloneDB>.ora 
-pwfile $ORACLE_HOME/dbs/orapw<CloneDB> 
-dbname <ClonedB> -startopt open -stopopt immediate 
-instance <CloneDB> -node <ServerName>
Making ACFS based clones

Demo
Making ACFS based clones
Cloning PDBs

SQL> CREATE PLUGGABLE DATABASE PDB3 FROM PDB2
SNAPSHOT COPY;

ACFSUTIL automatically triggered in background
Making ACFS based clones
Cloning PDBs

Requirements

- Source PDB must be ACFS
- Source PDB must be in Read-Only

SQL> ALTER SESSION SET DB_CREATE_FILE_SET='<ACFS_PATH>';
Making ACFS based clones
Cloning PDBs

Automation

- PDBaaS solution provided
- Based on APEX
Let’s meet at booth 242
Appendix
Workshop Icons

Tips / information
Warning
Highlighting
Best practice

Demo
Exercice
Negative / contra
Positive / pro

No phone
Apéro

Databases Clone using ACFS
Appendix

IT Icons

- Database
- Database (highlighted)
- Workstation
- Server1
- Server2
- Notebook
- Data Cube
- Switch or hub
- Scanner
- Printer

Databases Clone using ACFS