How to Backup & Recovery Enormous Databases ?

Husnu Sensoy

UKOUG Conference Series
Technology & E-Business Suite
ICC, Birmingham | 30th November- 2nd December

husnu.sensoy@gmail.com

December 2, 2009

Content

1. Introduction
2. Incrementally Updated Backup (IUB) 101
   - How does Incrementally Updated Backup work ?
   - How to Recover in Case of a Failure ?
   - What do you need for IUB?
   - Economical Evaluation of IUB Strategy
3. JeS for IUB
   - Motivation
   - Hardware Choice
   - Redundancy Configuration
4. RMAN Tips for IUB
   - Compression
   - Other Tips for RMAN
5. Fine Tuning IUB Strategy with ZFS
   - Motivation
   - Optimizing Storage Requirement
   - Fast Backup Health Check
   - Multiple Image Copies
   - More
6. Conclusion

Who am I ?

- VLDB Expert
- Oracle ACE
- Oracle DBA of 2009
- Oracle Blogger in ▸ The great grandson of Husnu Sensoy
- Speaker in various meetings like Open World, User Groups, and Universities
- Master of Science Student at on *I/O Scheduling on Grid Environments*

What is the bare minimum for a good B&R solution ?

A Backup & Recovery solution is good if

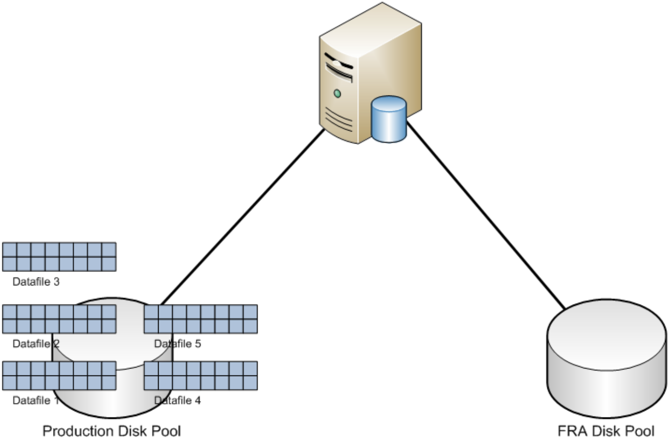
- If you can perform full database recovery *fast*.
- If your backups are not pain in the neck of your database.
- If you can validate your backup(s) health *quickly* before a crash occurs.
- If you keep the cost of backup recovery minimum without sacrifying anyone above.

Good B&R Solution vs Enormous Databases

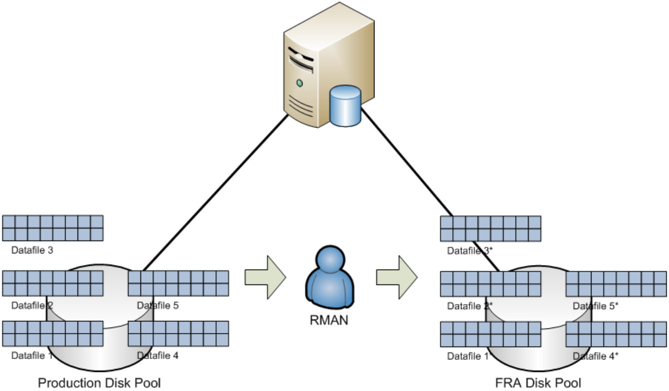
Majority of B&R solutions in the market can not be defined as good for enormous databases if

- Backup duration is a function of database size (Think about full daily backups)
- Recovery duration is a function of database size (Think about incremental backups)
- They have enormous licensing cost (Think about storage level solutions)

Toy Database Architecture



Day 1

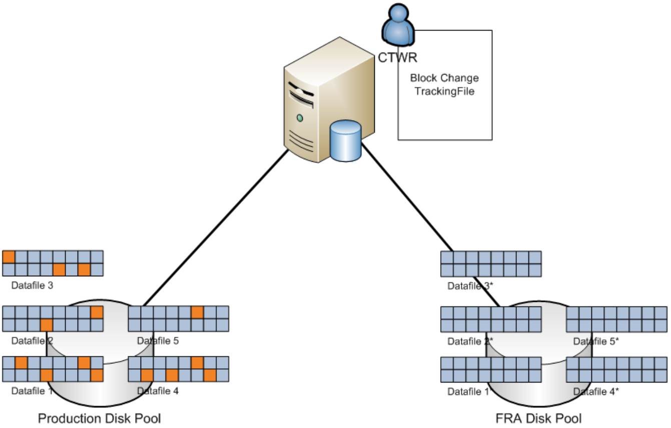

Datafile 3
Datafile 2
Datafile 5
Datafile 1
Datafile 4
Production Disk Pool
RMAN
Datafile 3*
Datafile 2*
Datafile 5*
Datafile 1*
Datafile 4*
FRA Disk Pool

RMAN Script You Need

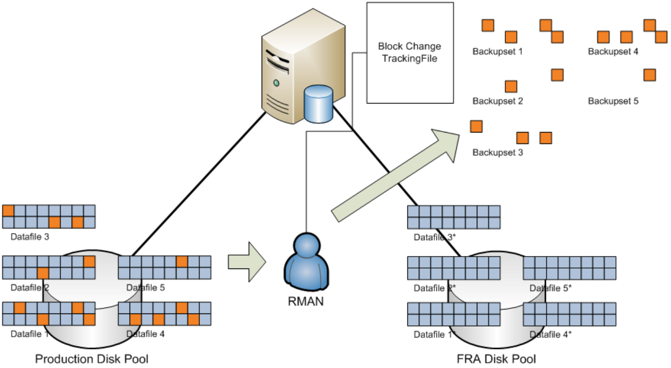
```
run{
 backup as compressed backupset check logical incremental level 1
 for recover of copy with tag DAILY_COPY database
 filesperset 1;

}
```

Do Some Change



Day 2+ Phase I

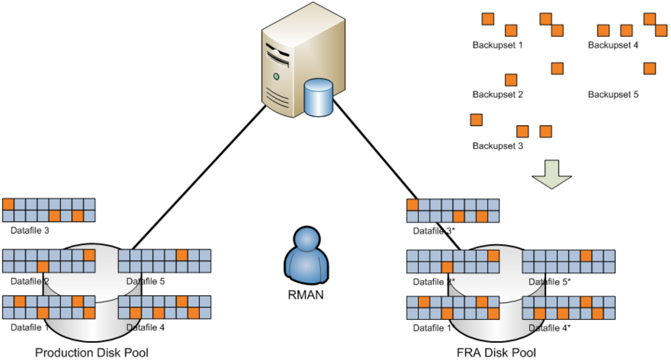

Block Change TrackingFile
Backupset 1
Backupset 4
Backupset 2
Backupset 5
Backupset 3
Datafile 3
Datafile 2
Datafile 5
Datafile 1
Datafile 4
Production Disk Pool
RMAN
Datafile 3*
Datafile 2*
Datafile 5*
Datafile 1*
Datafile 4*
FRA Disk Pool

RMAN Script You Need

```
run{
 backup as compressed backupset check logical incremental level 1
 for recover of copy with tag DAILY_COPY database
 filesperset 1;

}
```

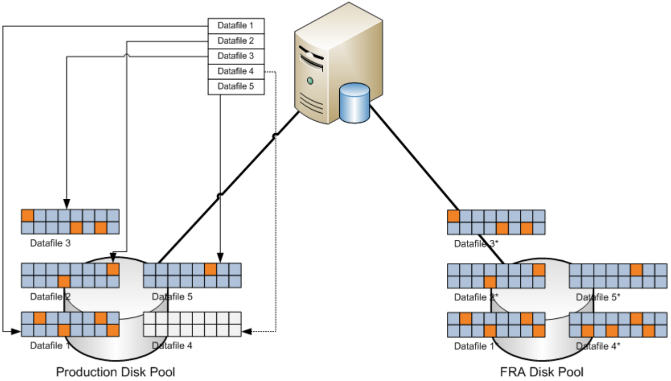
Day 2+ Phase II



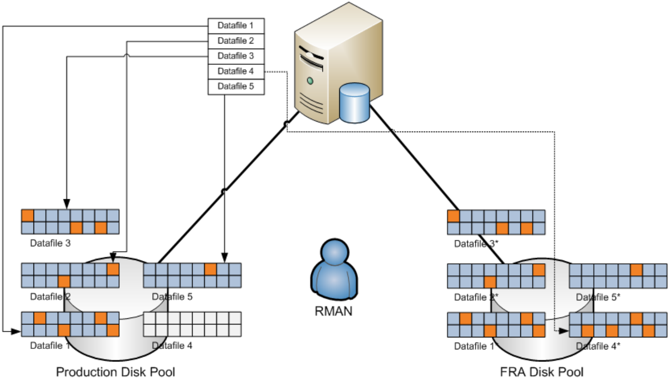
RMAN Script You Need

```
run{
 recover copy of database with tag DAILY_COPY;
}
```

You have lost the datafile 4



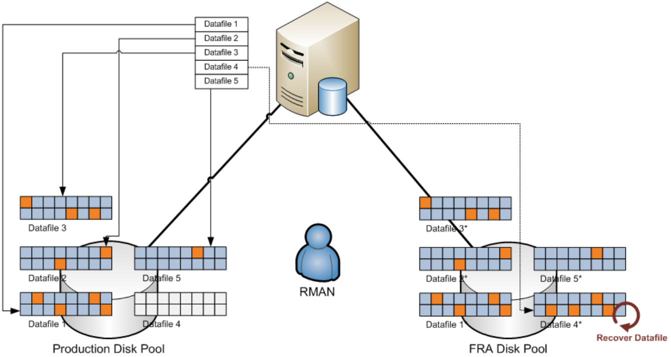
Switch to Copy



RMAN Script You Need

```
run{
 switch datafile 4 to copy;
}
```

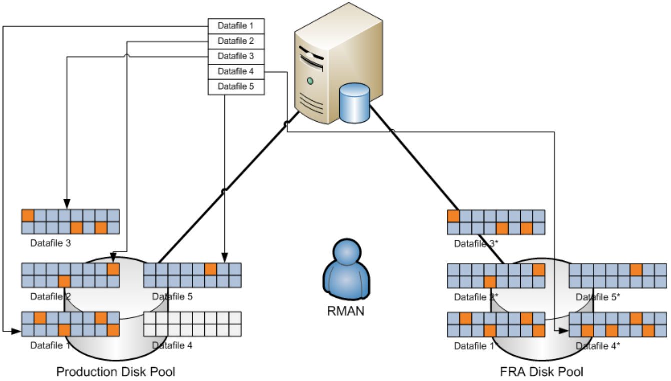
Recover it


Datafile 1
Datafile 2
Datafile 3
Datafile 4
Datafile 5
Datafile 3
Datafile 2
Datafile 5
Datafile 1
Datafile 4
Production Disk Pool
RMAN
Datafile 3*
Datafile 2*
Datafile 5*
Datafile 1
Datafile 4*
Recover Datafile
FRA Disk Pool

RMAN Script You Need

```
run{
 recover datafile 4;
}
```

Online it


Datafile 1
Datafile 2
Datafile 3
Datafile 4
Datafile 5
Datafile 3
Datafile 2
Datafile 5
Datafile 1
Datafile 4
Production Disk Pool
RMAN
Datafile 3*
Datafile 2*
Datafile 5*
Datafile 1*
Datafile 4*
FRA Disk Pool

RMAN Script You Need

```
run{
 sql 'alter database datafile 4 online';
}
```

Here is your crew for IUB I

FRA >

- You should use FRA for IUB.
- Incremental backup sets and archive logs can be located in arbitrary locations although they **should not**, however only legitimate location for image copies is FRA. If RMAN can not find them in FRA, it will start a copy to FRA from scratch.
- Size of the storage pool that FRA resides should be slightly larger than the storage pool hosting your database.

Here is your crew for IUB II

RMAN >

- RMAN is the only way to utilize IUB.
- RMAN is/will be the only B&R tool that is eligible to read,write, and modify Oracle blocks which is a must for a methodology like IUB.

Here is your crew for IUB III

Storage Pool Technology >

- In theory any mount point can be used for `db_recovery_dest` parameter.
- But when it comes to realities of practical life your options are limited with 3-4 different solutions like ASM,ACFS, and ZFS because those technologies allows you to create arbitrarily large containers without getting the burden of FS check in case of a problem.

Some Numbers

- Cost of a single Sun StorageTek T10K tape driver with theoretical read/write throughput rate of 300MB/s(binary tape compression enabled
- TB cost of tier one SAN storage is around 2000 - 3000$
- TB cost of a tape cartridge is around 150 $
- Assume that we are at the stage of architecting the backup & recovery solution for our new 100 TB enormous database.

IUB Solution

- We bought a tier one SAN box of size 150 TB (to accommodate archivelogs and incremental backups also)
- It costs around 300.000 - 450.000$
- It can deliver practically 3-5 GB/s backup speed for sequential workload.

Full Tape Backup Solution

- We bought 4 - 8 T10K tape drivers.
- A single copy of my data warehouse will cost around 80.000 - 160.000$
- I still can deliver 1 - 3 GB/s backup & recovery rate in theory

Conclusion

As all of us can conclude that IUB is almost 4 times expensive than well - known tape backup solution.
And we all accepted it as an inefficient solution.
And reject to invest on it.

And One Day ...

End one day your enormous database crashes due to an incorrectable disk problem in your production storage (It is not as rare as you think when it comes to human errors)...

Let's Do a Full DB Restore

- Assume that you are lucky and can reach up to theoretical limits of your driver pool

Let's Do a Full DB Restore

- Assume that you are lucky and can reach up to theoretical limits of your driver pool
- Then restoring 100 TB will take :
    - 33333 seconds

Let's Do a Full DB Restore

- Assume that you are lucky and can reach up to theoretical limits of your driver pool
- Then restoring 100 TB will take :
    - 33333 seconds
    - or 555 minutes

Let's Do a Full DB Restore

- Assume that you are lucky and can reach up to theoretical limits of your driver pool
- Then restoring 100 TB will take :
  - 33333 seconds
  - or 555 minutes
  - or 10 hours

Let's Do a Full DB Restore

- Assume that you are lucky and can reach up to theoretical limits of your driver pool
- Then restoring 100 TB will take :
  - 33333 seconds
  - or 555 minutes
  - or 10 hours
- All of us know that I will not be completed before 24 hours :)

Let's Do a Full DB Restore

- Assume that you are lucky and can reach up to theoretical limits of your driver pool
- Then restoring 100 TB will take :
  - 33333 seconds
  - or 555 minutes
  - or 10 hours
- All of us know that I will not be completed before 24 hours :)
- If you were using IUB it will take just a few minutes to bring up whole DB including Google search time.

Conclusion of DBA

- Do not stick on backup cost and understand well enough what you are sacrifying.

Conclusion of DBA

- Do not stick on backup cost and understand well enough what you are sacrifying.
- Explain money holders well enough what they are sacrifying.

Conclusion of DBA

- Do not stick on backup cost and understand well enough what you are sacrifying.
- Explain money holders well enough what they are sacrifying.
- Keep in mind that for enormous databases tape drivers are acceptable only for long term archival solutions.

I am a DBA. Why to bother about storage ?

- Keep in mind that in case of a failure in production disk pool, you will be surviving on FRA pool for a period of time until you recover the original pool. So an arbitrary FRA disk pool performance is not expectable.
- One of the major reasons why people take IUB as a luxury solution is that only storage solution they know is SAN. But there are more if you combine the right tools
- As we will discuss in case of an erroneous storage configuration, DB may not be recoverable even that you rely on IUB.

Several Options for Hardware

Here are some of your options

- SAN with tier N storage

Several Options for Hardware

Here are some of your options

- SAN with tier N storage
- NFS filers

Several Options for Hardware

Here are some of your options

- SAN with tier N storage
- NFS filers
- iSER/RDS Infiniband storage servers

SAN

- Proven stability in years
- Well supported by several vendors
- Not very cost effective for high tiers
- Moderate performance

NFS

- Easy to configure
- Ready to be used by multiple databases
- Poorer performance
- Cost effective

iSER/RDS

- Relatively new technologies
- Not available on all platforms
- Initial setup/learning cost
- Highest Performance
- Best performance/price ratio

Two Simple ASM diskgroups

```
CREATE DISKGROUP DATA EXTERNAL DISK '/dev/rdsk/emcpower01',
'/dev/rdsk/emcpower03',
'/dev/rdsk/emcpower05';
```

Two Simple ASM diskgroups

```
CREATE DISKGROUP DATA EXTERNAL DISK '/dev/rdsk/emcpower01',
'/dev/rdsk/emcpower03',
'/dev/rdsk/emcpower05';

CREATE DISKGROUP FRA EXTERNAL DISK '/dev/rdsk/emcpower02',
'/dev/rdsk/emcpower04',
'/dev/rdsk/emcpower06';
```

Under the Hoods

- What we care usually are LUNs as DBAs
- Storage guys usually say *Do not bother friend! We have RAID X in this box*
- Why RAID 5, RAID 1, or anything else can not protect you in some cases?

Case Study

- Prepare a RAID 5 7+1 raid group by using 8x300 GB disks

Case Study

- Prepare a RAID 5 7+1 raid group by using 8x300 GB disks
- You heard about the bug in 10.2 forcing us to use ASM disk size < 2 TB

Case Study

- Prepare a RAID 5 7+1 raid group by using 8x300 GB disks
- You heard about the bug in 10.2 forcing us to use ASM disk size $<$ 2 TB
- So you split each raid group into two and give them to your DBA as

Case Study

- Prepare a RAID 5 7+1 raid group by using 8x300 GB disks
- You heard about the bug in 10.2 forcing us to use ASM disk size < 2 TB
- So you split each raid group into two and give them to your DBA as

RAID GROUP	LUN 1	LUN 2
RG01	/dev/rdsk/emcpower01	/dev/rdsk/emcpower02
RG02	/dev/rdsk/emcpower03	/dev/rdsk/emcpower04
RG03	/dev/rdsk/emcpower05	/dev/rdsk/emcpower06

Case Study

- Assume that you have lost RG01 due to an incorrectable problem.

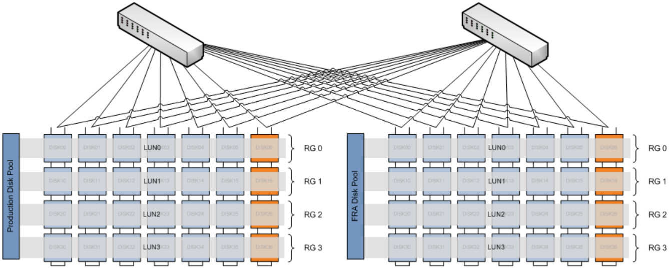
Case Study

- Assume that you have lost RG01 due to an incorrectable problem.
- Although you did everythin correct with IUB,

Case Study

- Assume that you have lost RG01 due to an incorrectable problem.
- Although you did everythin correct with IUB,
- Since you lost `/dev/rdsk/emcpower02`, you also lost FRA

A sample SAN redundant at each level



RMAN Backupset Set Optimization Techniques

- Uninitialized block optimization since 9i
- Empty block optimization since 10g
- Undo optimization by 11g Release 1

RMAN Compression Enhancement Techniques

- Pre-Compression Block Processing a is technique to increase the redundancy for the unused parts of database blocks by injecting pseudo redundancy to increase the effectiveness of compression (filling with 0s).
- To Enable

  ```
  CONFIGURE COMPRESSION ALGORITHM ’LOW’ AS OF RELEASE ’11.2.0.0.0’
  OPTIMIZE FOR LOAD FALSE;
  ```

RMAN Compression Enhancement Techniques

- Pre-Compression Block Processing a is technique to increase the redundancy for the unused parts of database blocks by injecting pseudo redundancy to increase the effectiveness of compression (filling with 0s).
- To Enable

  ```
  CONFIGURE COMPRESSION ALGORITHM ’LOW’ AS OF RELEASE ’11.2.0.0.0’
  OPTIMIZE FOR LOAD FALSE;
  ```
- To Disable (Default)

  ```
  CONFIGURE COMPRESSION ALGORITHM ’LOW’ AS OF RELEASE ’11.2.0.0.0’
  OPTIMIZE FOR LOAD TRUE;
  ```

RMAN Compression Enhancement Techniques

- Pre-Compression Block Processing a is technique to increase the redundancy for the unused parts of database blocks by injecting pseudo redundancy to increase the effectiveness of compression (filling with 0s).
- To Enable

  ```
  CONFIGURE COMPRESSION ALGORITHM ’LOW’ AS OF RELEASE ’11.2.0.0.0’
  OPTIMIZE FOR LOAD FALSE;
  ```
- To Disable (Default)

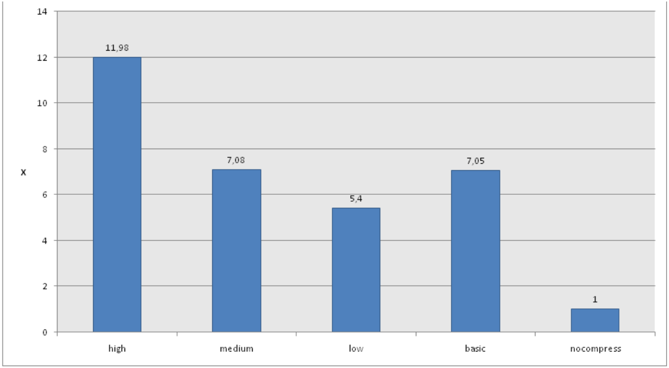
  ```
  CONFIGURE COMPRESSION ALGORITHM ’LOW’ AS OF RELEASE ’11.2.0.0.0’
  OPTIMIZE FOR LOAD TRUE;
  ```
- Use it with data files having sparse blocks because it is not priceless.

RMAN Binary Compression

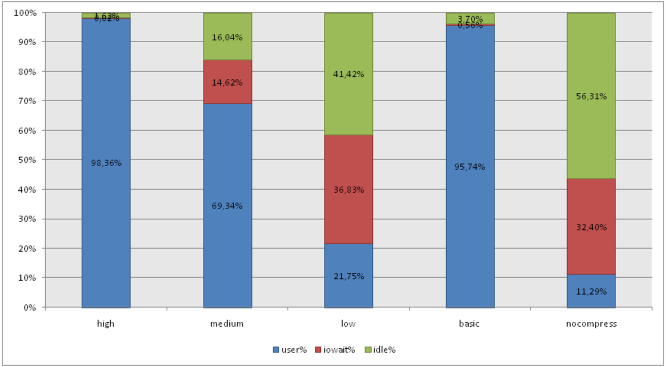
```
select * from v$rman_compression_algorithm;

ALGO_N   INIT_REL   ALGO_DESC                                  ALGO_COMP   IS_VAL   REQ_ACO
BASIC    10.0.0.0.0 good compression ratio                     11.2.0.0.0  YES      NO
LOW      11.2.0.0.0 maximum possible compression speed         11.2.0.0.0  YES      YES
MEDIUM   11.2.0.0.0 balance between speed and compression ratio 11.2.0.0.0 YES      YES
HIGH     11.2.0.0.0 maximum possible compression ratio         11.2.0.0.0  YES      YES
```

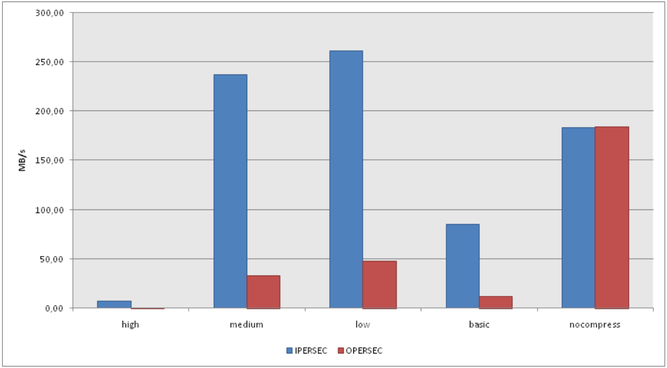
How much do they compress ?



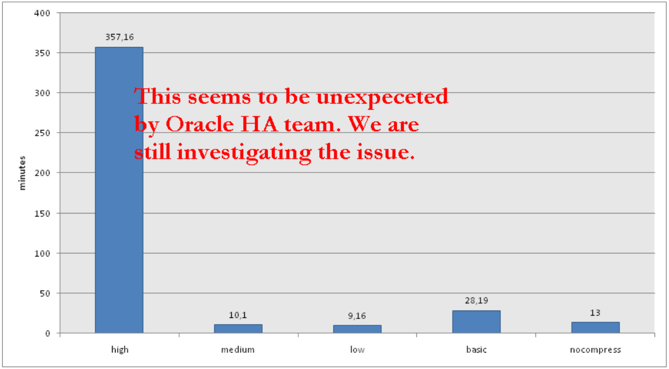
How much resource do they consume ?



How about I/O throughput ?



How much time do they require ?



This seems to be unexpeceted by Oracle HA team. We are still investigating the issue.

How to align those algorithms with our IUB strategy ?

Here is my rules of thumb on binary compression as far as IUB is concerned

- In 10g, disable backupset compression.
- In 11g Release 1, enable `ZLIB` compression.
- In 11g Release 2, enable `MEDIUM` or `LOW` level of advance compression option.

Final Remark on Binary Compression for IUB

- For all practical purposes, RMAN's compression algorithms are very effective for backupset compression as we compare them with hardware level solutions (tape compression,etc)

Final Remark on Binary Compression for IUB

- For all practical purposes, RMAN's compression algorithms are very effective for backupset compression as we compare them with hardware level solutions (tape compression,etc)
- That fundamentally because they know what is inside an Oracle block

Final Remark on Binary Compression for IUB

- For all practical purposes, RMAN's compression algorithms are very effective for backupset compression as we compare them with hardware level solutions (tape compression,etc)
- That fundamentally because they know what is inside an Oracle block
- But the major problem is that they don't allow us to compress image copies which dominate the FRA usage

Those are the ones you may already know

Those are my baselines:

number of RMAN channels How many of them ?

- For image copy (Day 1), number of channels should be set to number of physical disk available in FRA pool.
- For incremental backup sets (Day 2+), number of channels should be set to number of physical cores (or cool threads) available of the host.

Those are the ones you may already know

Those are my baselines:

number of RMAN channels How many of them ?

- For image copy (Day 1), number of channels should be set to number of physical disk available in FRA pool.
- For incremental backup sets (Day 2+), number of channels should be set to number of physical cores (or cool threads) available of the host.

`filesperset` should be set to 1 for SAME

Those are the ones you may already know

Those are my baselines:

number of RMAN channels How many of them ?

- For image copy (Day 1), number of channels should be set to number of physical disk available in FRA pool.
- For incremental backup sets (Day 2+), number of channels should be set to number of physical cores (or cool threads) available of the host.

`filesperset` should be set to 1 for SAME

`check logical` option should be enabled during incremental backups especially in 10g.

Those are the ones you may already know

Those are my baselines:

number of RMAN channels How many of them ?

- For image copy (Day 1), number of channels should be set to number of physical disk available in FRA pool.
- For incremental backup sets (Day 2+), number of channels should be set to number of physical cores (or cool threads) available of the host.

`filesperset` should be set to 1 for SAME

`check logical` option should be enabled during incremental backups especially in 10g.

change tracking file should be enabled for fast incremental backups.

Those are the ones you may already know

Those are my baselines:

number of RMAN channels How many of them ?

- For image copy (Day 1), number of channels should be set to number of physical disk available in FRA pool.
- For incremental backup sets (Day 2+), number of channels should be set to number of physical cores (or cool threads) available of the host.

`filesperset` should be set to 1 for SAME

`check logical` option should be enabled during incremental backups especially in 10g.

change tracking file should be enabled for fast incremental backups.

move FRA to tape Best way to do this is to use `BACKUP RECOVERY AREA`

Those are the ones you may already know

Those are my baselines:

number of RMAN channels How many of them ?

- For image copy (Day 1), number of channels should be set to number of physical disk available in FRA pool.
- For incremental backup sets (Day 2+), number of channels should be set to number of physical cores (or cool threads) available of the host.

`filesperset` should be set to 1 for SAME

`check logical` option should be enabled during incremental backups especially in 10g.

change tracking file should be enabled for fast incremental backups.

move FRA to tape Best way to do this is to use `BACKUP RECOVERY AREA`

online verification of image copies `backup check logical validate datafilecopy all` allows you to read all your image copies by performing a detailed diagnostic on blocks.

Requirements Reloaded

A Backup & Recovery solution is good if

- satisfied If you can perform full database recovery *fast*.

Requirements Reloaded

A Backup & Recovery solution is good if

- satisfied If you can perform full database recovery *fast*.
- satisfied If your backups are not pain on the neck of your database.

Requirements Reloaded

A Backup & Recovery solution is good if

satisfied If you can perform full database recovery *fast*.

satisfied If your backups are not pain on the neck of your database.

partially If you can ensure your backup(s) health *quickly* before a crash.

Requirements Reloaded

A Backup & Recovery solution is good if

- satisfied If you can perform full database recovery *fast*.
- satisfied If your backups are not pain on the neck of your database.
- partially If you can ensure your backup(s) health *quickly* before a crash.
- partially If you keep the cost of backup recovery minimum without sacrifying anything above.

Let's Start with Bad News

If you accept to choose ZFS as your FRA target instead of ASM

- You will loose dynamic rebalance capability for dropping disks. You are not allowed to drop a disk from ZFS pool.

Let's Start with Bad News

If you accept to choose ZFS as your FRA target instead of ASM

- You will loose dynamic rebalance capability for dropping disks. You are not allowed to drop a disk from ZFS pool.
- I/O balancing will not be as good as ASM because ZFS stripes only new data files on new members of ZFS pool

Let's Start with Bad News

If you accept to choose ZFS as your FRA target instead of ASM

- You will loose dynamic rebalance capability for dropping disks. You are not allowed to drop a disk from ZFS pool.
- I/O balancing will not be as good as ASM because ZFS stripes only new data files on new members of ZFS pool
- If you are not using one of below you will not be able to use ZFS:
    - Solaris 10 Update 6+
    - Open Solaris Build 27+
    - Linux with FUSE

But there are Good Ones

If you accept to choose ZFS as your FRA target instead of ASM

- You can reduce the FRA size requirements almost by half (even for highly compressed DWH) with a minimum CPU cost

But there are Good Ones

If you accept to choose ZFS as your FRA target instead of ASM

- You can reduce the FRA size requirements almost by half (even for highly compressed DWH) with a minimum CPU cost
- You can open your database on another server in 5 minutes in order to verify backup health without harming your FRA pool

But there are Good Ones

If you accept to choose ZFS as your FRA target instead of ASM

- You can reduce the FRA size requirements almost by half (even for highly compressed DWH) with a minimum CPU cost
- You can open your database on another server in 5 minutes in order to verify backup health without harming your FRA pool
- You can have multiple versions of your image copied databases with minimum space requirement

Create your ZFS Pool and FS

```
time zpool create -f tank /dev/sdd /dev/sdf

real    0m0.846s
user    0m0.007s
sys     0m0.034s
```

Create your ZFS Pool and FS

```
time zpool create -f tank /dev/sdd /dev/sdf
```

```
real    0m0.846s
user    0m0.007s
sys     0m0.034s
```

```
time zfs create tank/fraOnZFS
```

```
real    0m0.641s
user    0m0.005s
sys     0m0.027s
```

Enable LZJB Compression

```
zfs set compression=lzjb tank/fra0nZFS
```

Enable LZJB Compression

```
zfs set compression=lzjb tank/fraOnZFS
```

Other option is to use different levels of gzip compression (gzip-[1-9]) with better compression ratios with the cost of CPU power

Allow oracle user to write ZFS and alter FRA destination

```
cd /tank/fraOnZFS/
```

Allow oracle user to write ZFS and alter FRA destination

```
cd /tank/fraOnZFS/
```

```
chown oracle:dba .
```

Allow oracle user to write ZFS and alter FRA destination

```
cd /tank/fraOnZFS/
```

```
chown oracle:dba .
```

```
alter system set db_recovery_file_dest = ’/tank/fraOnZFS/’;
```

Comparing Results

Solution	Image Copy Duration	Size
ASM	10:14	1733.5 MB
ZFS	06:52 (x1.5)	561 MB (x3.09)

Create a Snapshot

```
zfs snapshot tank/fraOnZFS@test
```

Create a Snapshot

```
zfs snapshot tank/fraOnZFS@test
```

```
zfs list
```

```
NAME USED AVAIL REFER MOUNTPOINT
tank 561M 3.36G 19K /tank
tank/fraOnZFS 561M 3.36G 561M /tank/fraOnZFS
tank/fraOnZFS@test 17K - 561M -
```

Create a writeable clone from the snapshot

```
 zfs clone tank/fraOnZFS@test tank/testclone
 ls -la /tank/testclone/PGROUND/datafile/* |  wc -l
4
```

Create a writeable clone from the snapshot

```
zfs clone tank/fraOnZFS@test tank/testclone
ls -la /tank/testclone/PGROUND/datafile/* | wc -l
4
```

```
rm -f /tank/testclone/PGROUND/datafile/o1_mf_users_5kbwcv40_.dbf
ls -la /tank/testclone/PGROUND/datafile/* | wc -l
3
```

Create a writeable clone from the snapshot

```
zfs clone tank/fraOnZFS@test tank/testclone
ls -la /tank/testclone/PGROUND/datafile/* |  wc -l
4
```

```
rm -f /tank/testclone/PGROUND/datafile/o1_mf_users_5kbwcv40_.dbf
ls -la /tank/testclone/PGROUND/datafile/* |  wc -l
3
```

```
ls -la /tank/fraOnZFS/PGROUND/datafile/* |  wc -l
4
```

Share the clone over NFS

```
zfs set sharenfs=on tank/testclone
```

Rest is simple

1. Mount the NFS share clone on a test server.
2. Mount DB using the control file in FRA
3. `switch database to copy;`
4. `recover database [until ...];`
5. `alter database open [resetlogs];`

ZFS Deduplication

- As you may all know Deduplication is the *colourful candy* for a few years.
- Latest version (not available for FUSE yet) of ZFS let you to enable deduplication at pool level.
- I have not test it yet but it seems promising.
- For more check ▸ Jeff Bonwick's Blog

ZFS is Promising

- Quotas for file systems in a zpool
- Different redundant configurations: mirrored, RAID-Z, double-parity RAID-Z
- Shorter release cycles with compared to ASM

Bottom Line

- Keep in mind that things chage.

Bottom Line

- Keep in mind that things chage.
- For enormous databases IUB seems to be the best solution for the time being

Bottom Line

- Keep in mind that things chage.
- For enormous databases IUB seems to be the best solution for the time being
- Keep your eyes on changes in storage technologies.

Bottom Line

- Keep in mind that things chage.
- For enormous databases IUB seems to be the best solution for the time being
- Keep your eyes on changes in storage technologies.
- Tape backup strategy is still crucial for archival purposes

Bottom Line

- Keep in mind that things chage.
- For enormous databases IUB seems to be the best solution for the time being
- Keep your eyes on changes in storage technologies.
- Tape backup strategy is still crucial for archival purposes
- ZFS might fine tune IUB for several platforms

Q&A