Backup And Journal Management
For Efficient Database Recovery
Goal

• Present a reliable backup and journal management strategy to repair broken database.

• Starting point to analyze your own strategy.
Agenda

• Basics on database recovery
• Backing-up database
• Restoring database
• Managing journal files
• Recovering database
Basics On Database Recovery

• To repair a broken database you will need:
  – A database backup to restore
  – All journal files from your backup to recover until the latest committed transaction

• Database backups and journal files should be on different devices than the database files
Backup: The Rdb Offering

• Rdb allows any combination of the following type of backup:
  - Full or incremental backup
  - Complete or partial (by area) backup
  - Online or offline backup
  - Quiet point or noquiet point backup
  - With or without checksum optimization
  - And more …
Backup: Why Do We Do It?

• Because I’m using a computer! 😊
• To be able to restore my database
  – As easily as possible
    ☑ Number of files involved
    ☑ Number of commands involved
  – As fast as possible
    ☑ Restore only what is mandatory
    ☑ Restore quickly

• The goal of backup is to restore and remember that:
  – During backup the application is running
  – During restore the application is down
Backup: Keep It Simple

• Full and Complete backup does provide a single backup file usable for all type of restore

• /CHECKSUM will verify the checksum of database pages during backup, avoiding backing up corrupted pages
  – It’s the default
  – Do not use /NOCHECKSUM

$ RMU/BACKUP/LOG DB:FOO BCK:FOO_BCK.RBF
Backup: Quiet_Point vs NoQuiet_Point

- NoQuiet_Point will not wait the end of active transactions before starting
  - Transactions may spawn over several backup files
  - This will be handled by Journal backup
  - It is not the default

$ RMU/BACKUP/LOG/NOQUIET DB:FOO BCK:FOO_BCK.RBF
Backup: Online vs NoOnline

• Depend if your production allow application downtime
• Online is just another database user doing a read only transaction
  – That transaction may be a long one
  – Watch snapshot records if you use snapshot in row cache
  – It is not the default

$ RMU/BACKUP/LOG/NOQUIET/ONLINE  DB:FOO BCK:FOO_BCK.RBF
Backup: Where To Backup?

- **Tapes**:
  - less subject to incident
  - require human intervention

- **Disks**
  - Are online, ready to be use
  - On-disk Rdb backups can be VMS backed up to tape
  - Parallel backup to disk is possible since 7.1
Backup: Getting Information

- From database
  - $ RMU/DUMP/HEAD=BACKUP
  - Display backup information store in the root

- From backup file
  - $ RMU/DUMP/BACKUP/OPTION=ROOT
  - Display the equivalent of rmu/dump/header
Incremental Backup Optimization

• By default Incremental Backup Scan Optimization is turn on
  – It speeds up Incremental backup
  – It has a cost, information is maintain in SPAM pages for that feature
  – It may results in high contention for SPAM pages, impacting you application performances

• If you don’t use incremental backup disable that feature
Restore: How Much?

• Just what is mandatory:
  – Few pages if just few pages are corrupt
  – Few storage areas if just few storage areas are lost
  – Whole database if needed

• Page and area restore can be done online

• Avoid restoring just the root
  – You will have to play with TSN and CSN
  – You may end up with a corrupt database and having to do a full restore
Restore: From What?

- Restore from your latest full, complete backup
  - Which mean you have to rely on it’s quality
- Regularly restore your backup file to another set of disks
  - It’s the only way to be sure that your backup file is reliable
    - allows to restore a useable database
  - Do it from a specialize account with the needed privileges and quotas
Restore: By Page

• Corrupted pages are shown by:
  – $ RMU/SHOW CORRUPT <database>

• You can restore and recover just those pages and it could be done online without stopping the application with:
  – $ RMU/RESTORE/JUST_CORRUPT/ONLINE –
    <backup-filename>
Restore: By Storage Area

• Corrupted storage areas are shown by:
  - $ RMU/SHOW CORRUPT <database>
  - $RMU/VERIFY/ROOT <database>

• You can restore and recover just those storage areas and it could be done online without stopping the application with:
  - $ RMU/RESTORE/AREA/ONLINE –
    <backup-filename> <Rdb-area-name>
Restore: Relocate Database Files

• RMU Backup can generate an option file which can be used by the restore command:
  – `RMU/BACKUP/RESTORE_OPTION=foo.opt`

• The option file can be edited to relocate the database file(s)
  – If you lost some database disks for example

• You restore with:
  – `$ RMU/RESTORE/OPTION=foo.opt`

• To relocate the root file you have to use:
  – `$ RMU/RESTORE/ROOT=file-spec`
Restore: Option File

• Option file can be use to modify storage area parameters
  – Including page size
• Restore command itself can be use to modify some of the database parameters like:
  – Open mode
  – Max number of nodes
  – Local/Global buffer parameters
  – …
Restore: Be Careful

• Avoid dropping the database before restoring it
  – We never know!
Journaling

• AIJ files record most of the database changes
• They will allow to roll forward database changes from the last backup to the latest committed transaction
• Some DML statements are not journaled
• On DML statements when Rdb warns you to do a database backup it means it, so trust it and do it
Journaling: Setup

• Use multiple circular journals distributed over several disks
  – It avoids single point of failure
  – If AIJ files are not available the database will shutdown

• Turn on ALS, it’s faster
  – A single process write to the AIJ file

• Reserve twice the number of slots as needed
  – To recreate, online, all journals if they get lost
  – To allow the ALS to create emergency AIJ if needed
Journaling: Setup Example

```
alter data file db:foo
reserve 10 journals
add journal foo_aij1 file jnl1:foo_aij1.ajj
add journal foo_aij2 file jnl2:foo_aij2.ajj
add journal foo_aij3 file jnl1:foo_aij3.ajj
add journal foo_aij4 file jnl2:foo_aij4.ajj
journal is enabled
  (backup server is manual,
   log server is automatic);
  disc all;
```
Journaling: Option File

• An option file, describing your journaling setup, can be created with:
  – $RMU/SHOW AFTER/OUT=aij_opt_filespec

• That option file can be edited to modify your journaling setup when restoring the database with:
  – $ RMU/RESTORE/AIJ_OPT=aij_opt_filespec
Journaling: Emergency Journals

• In the same directory as the root file by default
  – Not that good!

• Place them on different device than the other AIJs
  – They may have been created because AIJ device has been lost

• Specify the location with the system logical name RDM$BIND_AIJ_EMERGENCY_DIR
Journaling: Backup

• Size AIJ files to backup AIJ at the same time as the database
  – All AIJs between database backups are available online, ready for automatic recovery

• Do quiet_point AIJ backup just before the database backup
  – Avoid transactions spawning over several AIJ backup
  – Guarantee a single startup point of recovery
Journaling: Backup (cont.)

- Recovery is mainly based on Sequence numbers
- AIJ backup allows you to include the sequence number in the AIJ backup file name

```
SQL> alter data file db:foo journal is enabled
SQL> (backup file bck:fooaibck (edit string is '_ ' + sequence));

$ rmu/backup/after db:foo ""
$ rmu/backup/noquiet/online db:foo bck:foo_bck.rbf
$ dir bck:*.*
FOOAIBCK_0.AIJ;1  FOO_BCK.RBF;1
```
Journaling: Information

- **RMU/Show Stat/screen=“AIJ journal information” db:foo**
  - Allow you to Zoom for more information on a given journal

- **RMU/Dump/Head=journal db:foo**

- **RMU/Dump/After/Nodata/Start=1/End=1 JNL1:FOO_AIJ1.AIJ**
Journaling: Information
(cont.)

Journaling: enabled  Shutdown:  60  Notify: disabled  State: Accessible
ALS: Running  ABS: disabled  ACE: disabled  FC: disabled  CTJ: disabled
ARB.Count:  300  ARB.Avail:  300  SwtchSched:  0  NxtSwtch:
FOO_AIJ1   0  *BACKUP NEEDED* Written Accessible
FOO_AIJ2   1  *BACKUP NEEDED* Written Accessible
FOO_AIJ3   2    527     319  Current Accessible
FOO_AIJ4                        Unused      512   Empty Latent Accessible

Available AIJ slot 1
Available AIJ slot 2

___AIJ Journal Information: FOO_AIJ1__________________________________________
___
___AIJ Journal "FOO_AIJ1"
___Filename is "SUSTAINING:[VIGIER.PRJ.BCK.DB]FOO_AIJ1.AIJ;1"
___Default AIJ filename is "TDB:FOO_AIJ1.AIJ"
___Backup sequence number is 0
___Allocation is defaulted to 512 blocks
___Extension is defaulted to 512 blocks
___Backup filename is "SUSTAINING:[VIGIER.PRJ.BCK.BCK]FOOAJBCK.AIJ;"
___Backup filename edits are defaulted to enabled
Journaling: Optimized AIJ

- “Optimize” an AIJ backup to improve recovery performance
- Eliminates rolled back transactions
- Updates are sorted by physical DBKEY
  - Sequential access through storage areas
  - Duplicates removed
  - Any database page only updated once
- Two sorting/recovery strategies
Journaling: Optimized A/J Sequential

• `/RECOVERY_METHOD = SEQUENTIAL`
  – by AREA:PAGE:LINE

• Recovery
  – Process one storage area at a time
  – Limited to the speed of single disk drive

• Avoids I/O contention between areas/devices
Journaling: Optimized AIJ Scatter

• `/RECOVERY_METHOD = SCATTER`
  – by `PAGE:AREA:-LINE`

• Recovery
  – Process all areas “at once”
  – Limited to throughput of all disk drives with areas that contain modified rows
  – Could “thrash” if many active areas on few disks

• Sorted by descending line number to cause LDX to be extended only once

• Performance improvements biased towards distribution of page number ranges among areas
Journaling: Using Optimized AIJ

- Optimize right after AIJ backups
  - Don’t wait till you need it
- Use optimized AIJ as your first choice
- Scatter may be better with many areas & drives
- Sequential may be better if you have few drives
- Keep original AIJ online as well
  - Some operations not possible with optimized AIJ as:
    - By page recover
    - By area recover
    - /Until recovery
Journaling: Automatic Recovery

• When backup was done with journaling on and all journals are available then restore will automatically start the database recovery

• From a single backup file and a single restore command, the database can be recovered until the latest committed transaction

• This is the goal of our backup strategy with the single startup point of recovery
Journaling: Automatic Recovery Example

$ RMU/RESTORE/NOCDD/LOG TBCK:FOO_BCK.RBF
%RMU-I-AIJRSTBEG, restoring after-image journal "state" information
  :
%RMU-I-AIJRSTEND, after-image journal "state" restoration complete
%RMU-I-RESTXT_00, Restored root file SUSTAINING:[VIGIER.PRJ.BCK.DB]FOO.RDB;1
  :
%RMU-I-LOGRECDDB, recovering database file SUSTAINING:[VIGIER.PRJ.BCK.DB]FOO.RDB;1
%RMU-I-AIJAUTOREC, starting automatic after-image journal recovery
  :
%RMU-I-AIJALLDONE, after-image journal roll-forward operations completed
%RMU-I-LOGSUMMARY, total 303 transactions committed
%RMU-I-LOGSUMMARY, total 0 transactions rolled back
%RMU-I-LOGSUMMARY, total 0 transactions ignored
%RMU-I-AIJSUCCES, database recovery completed successfully
  :
Journaling: Automatic Recovery And Journals

• After recovering from the current journals a new OpenVMS version of the journals is created
  – Poor man backup technic

• Keep the previous one safe, they have not been backed-up!

$ dir tdb:* .aij

FOO_AIJ1.AIJ;2  FOO_AIJ1.AIJ;1
FOO_AIJ2.AIJ;2  FOO_AIJ2.AIJ;1
FOO_AIJ3.AIJ;2  FOO_AIJ3.AIJ;1
FOO_AIJ4.AIJ;1
Journaling: Manual Recovery

- When restoring from database backup older than the latest one
- If database backup fail on database corruption
  - Corruption detected by /Checksum default qualifier
  - AIJs are already backed-up and not available online
- Restore will tell you which AIJ sequence number is expected
  - Here’s the advantage of having sequence number in the AIJ backup filename
$ rmu/backup/after tdb:foo ""

$ rmu/backup/noquiet/online tdb:foo tbck:foo_bck_3.rbf

%RMU-E-CORPAGPRES, Corrupt or inconsistent pages are present in area
SUSTAINING:[VIGIER.PRJ.BCK.DB]FOO_DAREA1.RDA;1

%RMU-F-FATALERR, fatal error on BACKUP

%RMU-F-FTL_BCK, Fatal error for BACKUP operation at 16-NOV-2004 07:28:05.88

$ rmu/restore/just_corrupt/log tbck:FOO_BCK_0.RBF

: %RMU-I-AIJRECFUL, Recovery of the entire database starts with AlJ file sequence 3

: $ rmu/recover/just_corrupt/log tbck:FOOAIJBCK_3.AIJ

:
Journaling: Recovery Performance

- Increase the number of recovery buffer to speed up the recovery buffers
- For automatic recovery use:
  - $ RMU/RESTORE/RECOVERY=AIJ_BUFFERS=<n>
- For manual recovery use:
  - $ RMU/RECOVER/AIJ_BUFFERS=<n>
- RMU/DUMP/AFTER/OPTION=STAT gives some recommendation which are good starting point for <n>
Journaling: Running Out Of Journal Space

• When you run out of journal space
  – AIJ disks lost
  – Unexpected database workload

• You can perform a NoQuiet_Point AIJ backup to avoid database shutdown

• Perform your usual backup procedure to get back to the single startup point of recovery
Key Note

• Get trained to recover your database
• Best training is to actually test your database recovery procedures regularly
For More Information

• Oracle Rdb
  – Guide to Database Maintenance
  – Release notes

• Oracle RMU Reference Manual