

# Backup & Recovery Enhancements in Oracle 10g

Radoslav Rusinov  
TechnoLogica Ltd.

- Availability is the most important characteristic of the business data

# Types of Failures and Oracle's Solutions

- **Human Errors**
  - Export, Data Pump, Log Miner, TSPITR
  - **Flashback Features**
- **Block Corruptions**
  - RMAN
- **Media Failures**
  - RMAN
- **Computer and Hardware Failures**
  - RAC, Data Guard, Replication, Streams
- **Disaster**
  - Data Guard, Replication, Streams

# Target Audience

- Oracle DBAs –  
Production  
Environments
- Oracle DBAs –  
Development  
Environments
- Oracle Developers



# Test Environment

- Used Test Environment:
  - Oracle Database Version: 10.2.0.2
  - Operating System: Windows XP SP2
  - Database Log Mode: ARCHIVELOG
- Assertions in this presentation are in result of tests performed on the above environment
- **Ensure that every feature works in the expected way in your specific environment**

# Agenda

- **Flash Recovery Area**
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion

# Automatic Disk-Based Backup & Recovery with **Flash Recovery Area**

- 95% of the recovery operations need no more than 3 days-old backups
- What is Flash Recovery Area?
  - Unified disk storage location
  - Default location for disk-based backup files
  - Defines the max available space for disk-based backup files
  - Can use file system or ASM disk group

# Flash Recovery Area (FRA)

- Automatically names the files by using OMF
- Has its own directory structure
  - `..\flash_recovery_area\ORCL102\`



Database Area



Flash Recovery Area

- ARCHIVELOG
- AUTOBACKUP
- BACKUPSET
- CONTROLFILE
- DATAFILE
- FLASHBACK
- ONLINELOG

# Flash Recovery Area (FRA)

- FRA is not a place for the data files
- Data files reside in the Database Area
- FRA must be located on a different physical storage than Database Area
- Configured with two parameters:
  - *DB\_RECOVERY\_FILE\_DEST* - destination
  - *DB\_RECOVERY\_FILE\_DEST\_SIZE* – max size

# Flash Recovery Area (FRA)

- You can create your own additional directories and files in FRA
- To configure retention policy:
  - *RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY | RECOVERY WINDOW OF*
- New RMAN commands to backup FRA
  - *RMAN> BACKUP RECOVERY AREA* – only to SBT device
  - *RMAN> BACKUP RECOVERY FILES* – only to SBT device

# Automated Space Management

- Oracle does not delete the “obsolete” files. It retain them until space pressure appears
- Behaves like a cache for backup files
- Oracle issues alert when FRA is **85%** full
- Critical alert when FRA is **3%** full
- At **100%** a message in the alert.log:

...

```
ORA-19815: WARNING: db_recovery_file_dest_size of ...  
bytes is 100.00% used
```

...

# Space Pressure

- Two file types reside in FRA
- Permanent files:
  - Multiplexed copies of control file
  - Multiplexed copies of online redo logs
- Transient files:
  - Archived redo logs
  - Datafile and controlfile copies
  - Controlfile autobackups
  - Backup sets

# Space Pressure

- Archived logs in FRA that are in the recovery window and that are not yet backed up are treated in a **similar way as permanent** files
- Backup sets in FRA that are in the recovery window and are not yet available on tape are treated in a **similar way as permanent** files
- Space pressure over FRA causes automatic deleting of transient files
- Oracle never deletes permanent files

# Space Pressure

- Backup sets treated as permanent will not permit creation of new backups and will not be deleted
- Default Oracle installation uses FRA as default and only one location for archived logs: *LOG\_ARCHIVE\_DEST\_10 = USE\_DB\_RECOVERY\_FILE\_DEST*
- This along with the default value of: *LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST=1* can make the archivelog destination in FRA mandatory and to cause space pressure

# Space Pressure – Recommendations

- Set smaller and reasonable recovery window
- Backup archived logs regularly
- Configure at least two archive log destinations
- Give more physical space to FRA
- Configure the FRA size to be at most 90% of the real physical available space
- For Disk-to-Disk-to-Tape strategies: backup backupsets from FRA to tape regularly
- For Disk-to-Tape strategies: perform database backups to tape regularly

# Space Monitoring

- V\$RECOVERY\_FILE\_DEST

```
SQL> SELECT * FROM v$recovery_file_dest;
```

NAME	SPACE_LIMIT	SPACE_USED	SPACE_RECLAIMABLE	NUMBER_OF_FILES
E:\oracle\ flash_reco very_area	6442450944	1299748352	0	27

- V\$FLASH\_RECOVERY\_AREA\_USAGE (10g R2):

```
SQL> SELECT * FROM v$flash_recovery_area_usage;
```

FILE_TYPE	PERCENT_SPACE_USED	PERCENT_SPACE_RECLAIMABLE	NUMBER_OF_FILES
CONTROLFILE	0	0	0
ONLINELOG	0	0	0
ARCHIVELOG	.69	0	3
BACKUPPIECE	18.99	0	17
IMAGECOPY	0	0	0
FLASHBACKLOG	.44	.38	6

# Monitoring FRA Metrics with EM

Database Instance: [orcl102.ad.tlogica.com](#) > [All Metrics](#) > [Recovery Area Free Space \(%\)](#) > Recovery Area RECOVERY AREA

## Recovery Area Free Space (%): Recovery Area RECOVERY AREA

Latest Data Collected From Target **Apr 9, 2006 4:22:15 PM**

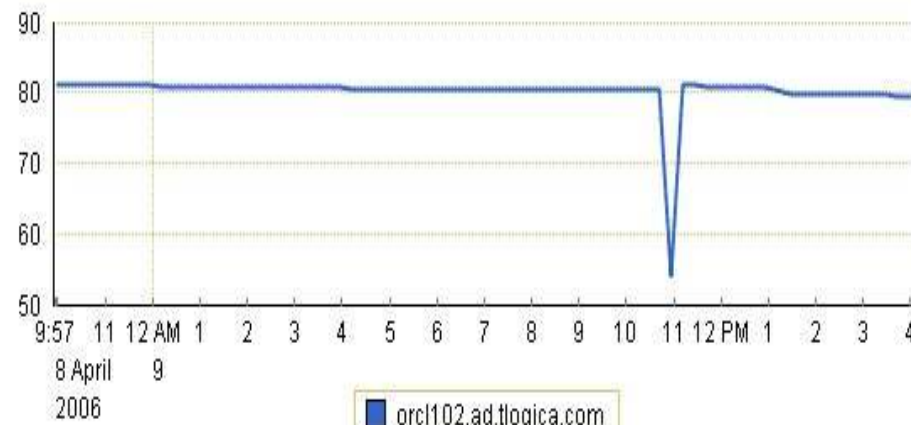
View Data  

Recovery Area **RECOVERY AREA**

### Statistics for Last 24 Hours

Last Known Value **79.64**  
Average Value **80.33**  
High Value **81.26**  
Low Value **54.07**  
Warning Threshold **Not Defined**  
Critical Threshold **Not Defined**  
Threshold Occurrences **No data**

### Metric Value



# Flash Recovery Area - Summary

- Unified place for your disk-based backups
- Automatically managed files
- Simplified administration tasks
- Faster Backup
- Faster Restore

# Agenda

- Flash Recovery Area
- **Flashback Technology Enhancements**
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- **Flashback DATABASE**
- Flashback DROP  
Flashback TABLE
- Flashback QUERY
- Flashback VERSIONS  
QUERY
- Flashback  
TRANSACTION  
QUERY

# FLASHBACK DATABASE – Faster Point-In-Time Recovery

## Human Error

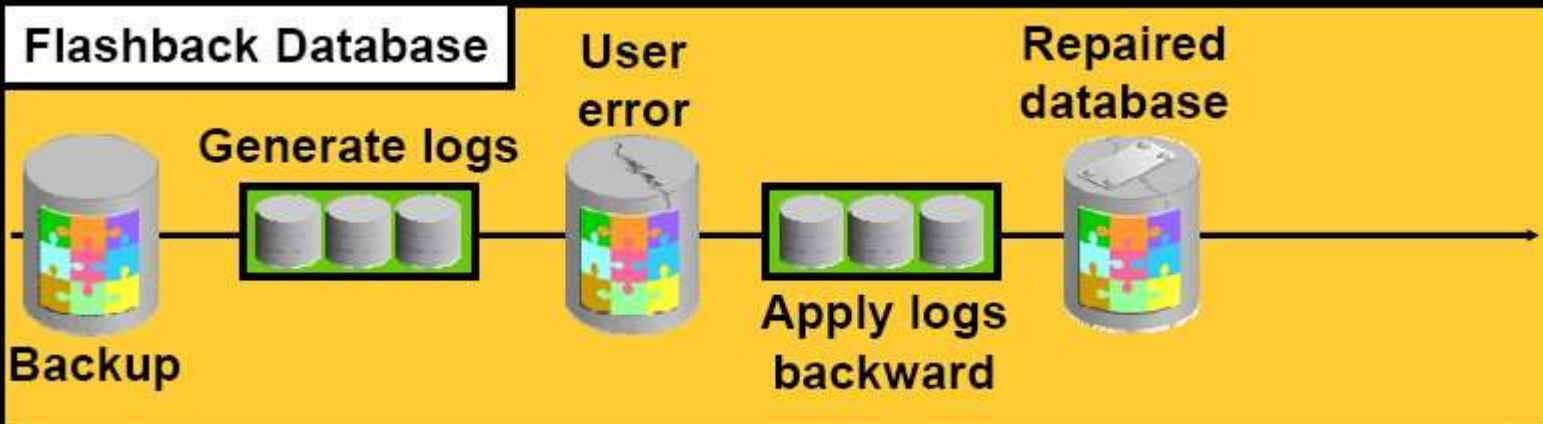
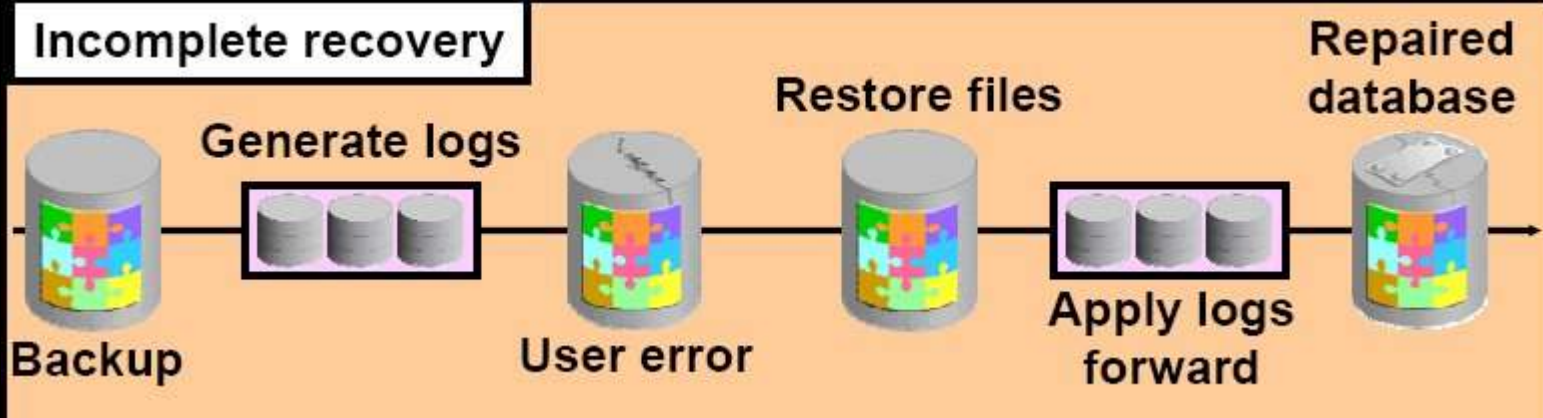
**Single Biggest Cause  
of Downtime**



# Flashback Database

- Reverse the whole database to a point in time
- Faster recovery from user errors
- Uses own logging mechanism
- Flashback Logs - stored in FRA
- New process: Recovery Writer (RVWR)
- New SGA structure: Flashback Buffer

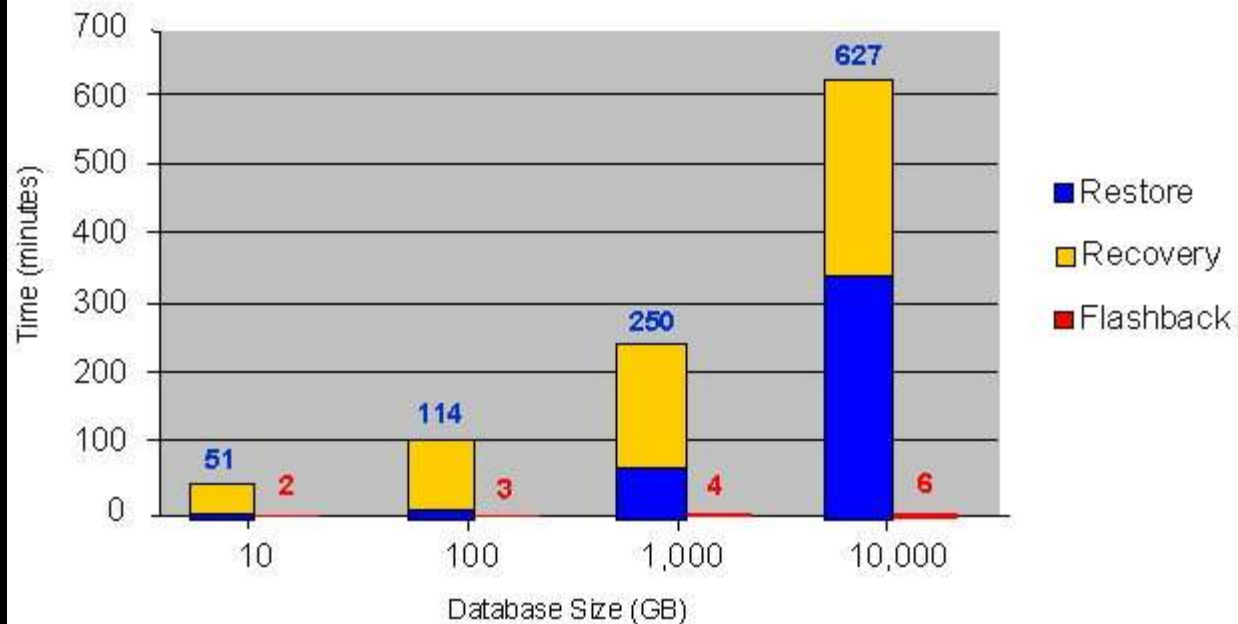
# Incomplete Recovery vs. Flashback Database



# Advantages

## Flashback Database versus Classic Point-In-Time Recovery

*Flashback Database is 100 Times Faster*



# Flashback Database

- The availability of the archived logs within the flashback recovery window is important
- The recovery depends on the size of changes
- Does not depend on the database size
- Prerequisites:
  - The database **MUST** be in ARCHIVELOG
  - FRA **MUST** be configured
  - The Flashback Logs can reside only in FRA

# Configuration with EM

## Flash Recovery

Flash Recovery Area is enabled for this database. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to a tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location

Flash Recovery Area Size

Flash Recovery Area Size must be set when the location is set

Reclaimable Flash Recovery Area (B) **0**

Free Flash Recovery Area (GB) **4.77**

Enable Flashback Database - flashback logging can be used for fast database point-in-time recovery\*

The flash recovery area must be set to enable flashback logging. When using flashback logs, you may recover your entire database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate.

Specify how far back you wish to flash the database in the future

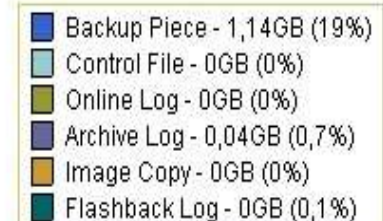
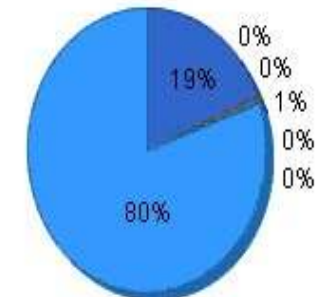
Flashback Retention Time

Current size of the flashback logs(MB) **49.688**

Lowest SCN in the flashback data **4121849534**

Flashback Time **Apr 9, 2006 11:34:07 AM**

## Flash Recovery Area Usage



# Manual Configuration

- SQL> STARTUP MOUNT EXCLUSIVE;
- SQL> ALTER DATABASE **FLASHBACK ON**;
- SQL> ALTER SYSTEM SET  
**DB\_FLASHBACK\_RETENTION\_TARGET** = 1440  
SCOPE=BOTH;
- **Flashback can be stopped at tablespace level:**
  - SQL> ALTER TABLESPACE USERS **FLASHBACK OFF**;
  - Excluded datafiles must be offlined if the database needs to be flashbacked and opened
  - After flashbacking a media recovery for the offlined datafiles must be performed

# Flashback OFF for SYSTEM Tablespace

- Oracle allows the flashback for logging for SYSTEM to be disabled
- This leads to non-working flashback database feature

```
SYSP@ORCL102> flashback database to timestamp to_date('10.04.2006 20:34','dd.mm.yyyy hh24:mi');
flashback database to timestamp to_date('10.04.2006 20:34','dd.mm.yyyy hh24:mi')
*
ERROR at line 1:
ORA-38753: Cannot flashback data file 1; no flashback log data.
ORA-01110: data file 1: 'E:\ORACLE\ORADATA\ORCL102\SYSTEM01.DBF'
```

- Don't exclude flashbacking for SYSTEM tablespace

# Space Issues

- *DB\_FLASHBACK\_RETENTION\_TARGET* is a **TARGET** parameter
- It does not guarantee your recovery window
- Flashback logs are not archived and are reused
- Flashback logs can cause space pressure over FRA

# Space Pressure

- If FRA is Full -> Recovery Writer (RVWR) will start to reuse flashback logs
- If FRA is Full -> transient files will be deleted
- Flashback logs are treated as **transient** files in FRA
- They can be automatically deleted in response of space pressure

# Space Pressure - Recommendations

- Set Flashback retention target to a reasonable value. No more than few days.
- Do not configure flashback retention target bigger than RMAN recovery window
- Configure the FRA size to be at most 90% of the real physical available space
- To delete immediately all flashback logs:

```
SQL> ALTER DATABASE FLASHBACK OFF;
```

# Current Flashback Window

- V\$FLASHBACK\_DATABASE\_LOG

```
SYSPORCL102> SELECT oldest_flashback_scn oldest_scn,  
2  to_char(oldest_flashback_time,'dd.mm.yyyy hh24:mi') oldest_time,  
3  round(flashback_size/1024/1024)||' MB' current_size,  
4  round(estimated_flashback_size/1024/1024)||' MB' estimated_size  
5  FROM v$flashback_database_log;
```

OLDEST_SCN	OLDEST_TIME	CURRENT_SIZE	ESTIMATED_SIZE
4121849534	09.04.2006 11:34	65 MB	185 MB

# Restore Points (10g R2)

- They are not exactly a feature of the Flashback Database
- Can be used with FLASHBACK TABLE
- They are a map between an SCN, TIME and DATABASE INCARNATION
- Rely on archived redo logs from around the time of their creation

# Restore Points (10g R2)

- “The most common cause for downtime is change”
- Use them before performing of any significant change within the database:
  - Application upgrades
  - Batch jobs
  - Starting of flashback database operations

# Normal Restore Points (10g R2)

- `SQL> CREATE RESTORE POINT  
before_app_upgrade;`
- Can be created when Flashback Database is OFF
- Automatically deleted if:
  - More than 2048 in count
  - Older than  
`CONTROL_FILE_RECORD_KEEP_TIME`

# Guaranteed Restore Points (10g R2)

- `SQL> CREATE RESTORE POINT  
before_app_upgrade GUARANTEE FLASHBACK  
DATABASE;`
- Flashback Database can be OFF
- Using Flashback Logs
- Could generate a lot of flashback logs

# Guaranteed Restore Points (10g R2)

- Not dropped automatically
- Must be explicitly dropped
- Usable for short periods of time
- Message in the *alert.log* after their creation
- They can cause the Database to HANG

# Using Flashback Database

- SQL> STARUP MOUNT
  - SQL> **FLASHBACK DATABASE TO TIMESTAMP**  
to\_timestamp('10.04.2006  
20:50','dd.mm.yyyy hh24:mi');
  - SQL> **FLASHBACK DATABASE TO RESTORE  
POINT** before\_app\_upgrade;
- SQL> ALTER DATABASE OPEN RESETLOGS;
- (10g R2) Keeps flashback logs after  
RESETLOGS
- (10g R2) SQL> FLASHBACK DATABASE TO **BEFORE  
RESETLOGS**;

# Monitoring with EM

- In Home Page

High Availability	
Instance Recovery Time (sec)	17
Last Backup	Apr 10, 2006 9:47:19 PM
Usable Flash Recovery Area (%)	76.17
Flashback Time	Apr 10, 2006 9:31:30 PM

- In Recovery Settings

Specify how far back you wish to flash the database in the future

Flashback Retention Time

Current size of the flashback logs(MB) 84.297

Lowest SCN in the flashback data 4121978661

Flashback Time Apr 10, 2006 9:31:30 PM

# Flashback Database - Summary

- Very powerful for faster point-in-time recovery
- Generate much less flash redo than online redo logs
- Can be used with restore points
- Space consumer
- I/O consumer

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- Flashback DATABASE
- Flashback DROP
- Flashback TABLE
- Flashback QUERY
- Flashback VERSIONS QUERY
- Flashback TRANSACTION QUERY

# FLASHBACK DROP - Undo a Dropped Table

- Reverses the effect of a DROP TABLE operation
  - SQL> **DROP TABLE** my\_table;
- Uses logical container called Recycle Bin
- Table remains in the Recycle Bin
- Can be queried

# Flashback DROP

- Using of Recycle Bin is ON by default
- The objects in the Recycle Bin are **included** in backups
- Can be disabled:
  - SQL> ALTER SESSION|SYSTEM **SET**  
**RECYCLEBIN=OFF;**
- To remove a dropped table from the Recycle bin:
  - SQL> **PURGE TABLE** my\_table;

# Flashback DROP

- To purge a tablespace:
  - SQL> **PURGE TABLESPACE** users;
- To purge a user-specific objects:
  - SQL> **PURGE TABLESPACE** users **USER** app\_user;
- To purge all dropped objects in the database:
  - SQL> **PURGE DBA\_RECYCLEBIN**;

# Space Pressure

- Objects in the Recycle Bin are not counted as used space
- Oracle drops objects in the Recycle Bin only in response of a space pressure
- The recycle bin objects are deleted before performing automatic extending of the datafile
- Be aware of user quotas

# Flashback DROP

- To restore a table from the Recycle bin:
  - SQL> **FLASHBACK TABLE** my\_table **TO BEFORE DROP**;
- Recover indexes
- Does NOT recover constraints
- Tables with defined security policies are NOT kept in the Recycle Bin

# FLASHBACK DROP - Summary

- For fastest table recovery after DROP TABLE
- Recovery success depends on the size of the underlying tablespace
- What is the difference:
  1. SQL> DROP TABLE my\_table;
  2. SQL> DROP TABLE my\_table **PURGE**;
- Most users are using the second statement already

# Agenda

- Flash Recovery Area
- **Flashback Technology Enhancements**
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- Flashback DATABASE
- Flashback DROP
- **Flashback TABLE**
- Flashback QUERY
- Flashback VERSIONS QUERY
- Flashback TRANSACTION QUERY

# Restore deleted table data with **FLASHBACK TABLE**

- Can recover table to a point in time in the past
- Uses information in the UNDO tablespace
- Indexes, constraints and triggers are restored automatically
- Different feature than the FLASHBACK DROP

# Flashback Table

- Enable row movement for the table:
  - SQL> **ALTER TABLE** my\_table **ENABLE ROW MOVEMENT** ;
- Recovers a table to a point in time:
  - SQL> **FLASHBACK TABLE** my\_table **TO TIMESTAMP** to\_timestamp('11.04.2006 20:50', 'dd.mm.yyyy hh24:mi');
- Can be used with restore points:
  - SQL> **FLASHBACK TABLE** my\_table **TO RESTORE POINT** before\_app\_upgrade;

# Flashback Table

- Cannot be used after DDL operations over the table
- Use Flashback Database to recover from:
  - SQL> **TRUNCATE TABLE** my\_table;
- The recovery window depends on the size of UNDO tablespace
- Can be configured with:  
*UNDO\_RETENTION* parameter

# Space Pressure

- It is a **TARGET** parameter
- In 10g *UNDO\_RETENTION* is automatically tuned
- In 10g the recovery window can be guaranteed:
  - `SQL> ALTER TABLESPACE undotbs RETENTION GUARANTEE ;`
- This can cause the database to HANG

# Agenda

- Flash Recovery Area
- **Flashback Technology Enhancements**
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- Flashback DATABASE
- Flashback DROP
- Flashback TABLE
- **Flashback QUERY**
- Flashback VERSIONS QUERY
- Flashback TRANSACTION QUERY

# FLASHBACK QUERY – Recovery at Row Level

- Allows a tables to be queried to the time in the past
- Available since Oracle9i
- Uses UNDO tablespace
- Querying old state of *my\_table*:

```
- SQL> SELECT * FROM my_table AS OF  
TIMESTAMP TO_TIMESTAMP('11.04.2006  
21:40', 'DD.MM.YYYY HH24:MI');
```

# Flashback Query

- Configure recovery window with *UNDO\_RETENTION* parameter
- Cannot be used after DDL operations
- Recovery of PL/SQL package after wrong compilation:

```
SQL> SELECT text FROM dba_source AS OF  
TIMESTAMP TO_TIMESTAMP('11.04.2006  
21:40', 'DD.MM.YYYY HH24:MI')  
WHERE owner='APP_USER1' AND  
NAME='LOST_PKG';
```

# Agenda

- Flash Recovery Area
- **Flashback Technology Enhancements**
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- Flashback DATABASE
- Flashback DROP
- Flashback TABLE
- Flashback QUERY
- **Flashback VERSIONS QUERY**
- Flashback TRANSACTION QUERY

# FLASHBACK VERSIONS QUERY – Audit Rows Changes

- Retrieve all versions of the rows between two points of time
- Uses UNDO tablespace
  - SQL> SELECT versions\_xid, name FROM my\_table **VERSIONS BETWEEN TIMESTAMP t1 AND t2;**
- Very useful for auditing of user actions

# Agenda

- Flash Recovery Area
- **Flashback Technology Enhancements**
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- Flashback DATABASE
- Flashback DROP
- Flashback TABLE
- Flashback QUERY
- Flashback VERSIONS QUERY
- **Flashback TRANSACTION QUERY**

# FLASHBACK TRANSACTION QUERY – Audit Transaction Changes

- Displays which transaction were responsible for certain changes for a given interval of time:
  - SQL> SELECT start\_scn, commit\_scn, operation, logon\_user, undo\_sql FROM FLASHBACK\_TRANSACTION\_QUERY;
- Uses UNDO tablespace
- Returns the exact undo statement to undo the transaction
- Cannot be used after DDL operation over the table

# Recovery with Flashback Features - Summary

- Very powerful
- Faster recovery
- Faster response of human errors
- Needs a space
- It is only an addition to your backup strategy, not a replacement for it
- There is no full protection from human errors - the DBA is a human too

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- **Space Pressure & Database Hangs**
- Physical Backup & Recovery Enhancements
- Conclusion

# Space Pressure & Database Hangs

- An UNDO tablespace with *RETENTION GUARANTEE* can cause the database to HANG
- A SYSTEM tablespace that cannot extend can cause the database to HANG

# Space Pressure & Database Hangs

- The Database cannot HANG if the FRA is Full
- Only two exceptions of the above:
  1. Archived logs that are treated as **permanent** files in FRA and space pressure from ARC process
  2. Flashback logs that are treated as **permanent** files. Caused by Guaranteed Restore Points

# Database Hangs - Recommendations

- Set smaller and reasonable recovery window
- Backup archived logs regularly
- Configure at least two archive log destinations
- Give more physical space to FRA
- Configure the FRA size to be at most 90% of the real physical available space
- Delete guaranteed restore points as soon as possible
- Monitor FRA to ensure that there is no space pressure

# ARC Process & Database Hangs

- Not started ARC process after putting the database in ARCHIVELOG mode can cause a hang
- ARC process is started automatically in 10g after putting the database in ARCHIVELOG mode

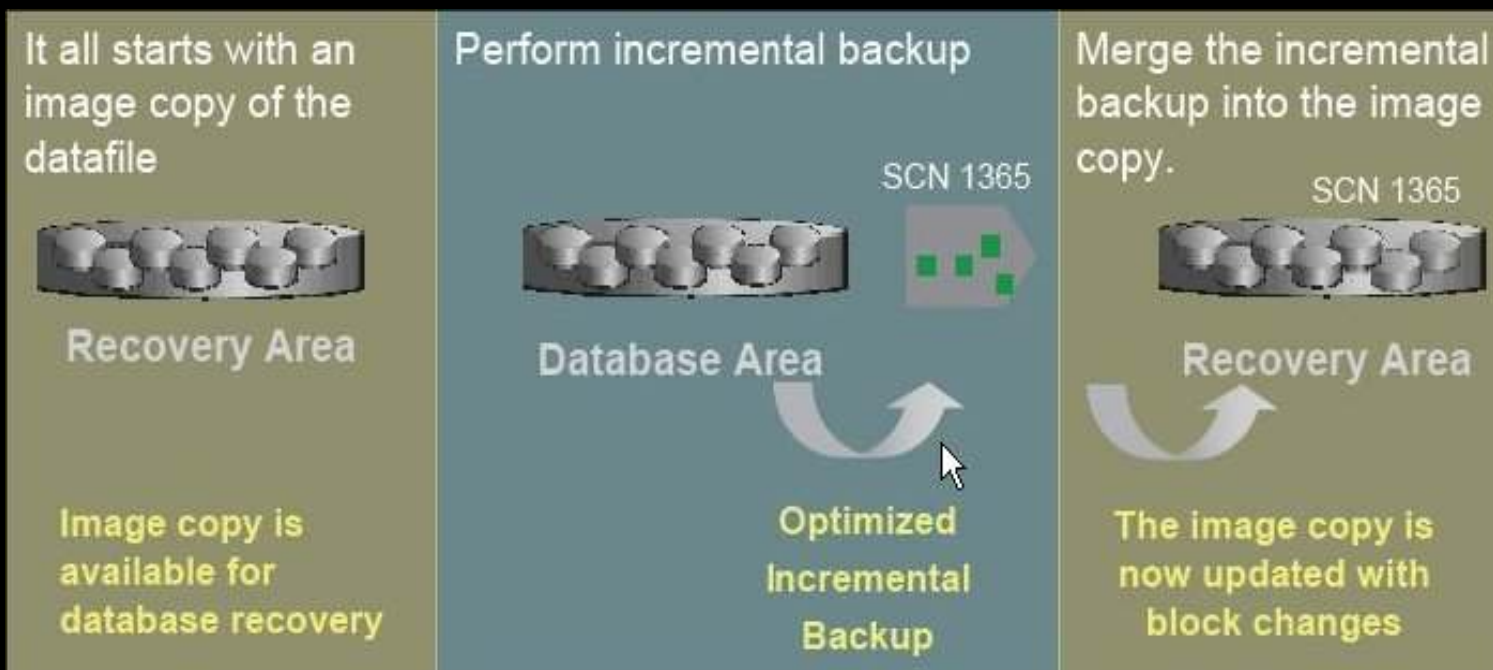
# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- **Incrementally Updated Backups**
- Change Tracking File
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- Oracle Secure Backup
- Miscellaneous - RMAN

# Recovery with Incrementally Updated Backups

- Incremental backups capture only block changes since the last backup
- In 10g there is ability to recover by using incrementally updated backup files
- Incremental backups are applied to the datafile image copies (level 0)

# Incrementally Updated Backups



# Incrementally Updated Backups

- RMAN> RUN {  
**RECOVER COPY OF DATABASE WITH TAG**  
**'incr\_updated\_baclup';**  
**BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY**  
**WITH TAG 'incr\_updated\_backup'**  
**DATABASE;**  
}
- Provides smaller Recovery Time Objective
- In case of real recovery image copies will be updated with last incremental backup plus needed small amount of archived logs

# Incrementally Updated Backups

- In case of media recovery RMAN will choose the incremental backups over the available archived logs
- Then only a small amount of archived logs will be applied
- It is much faster approach
- Ensures much faster media recovery

# Oracle-Suggested Backup Strategy

- ```
RUN {  
  ALLOCATE CHANNEL oem_disk backup DEVICE TYPE DISK;  
  RECOVER COPY OF DATABASE WITH TAG 'ORA$OEM_LEVEL_0';  
  BACKUP INCREMENTAL LEVEL 1 CUMULATIVE COPIES=1 FOR  
  RECOVER OF COPY WITH TAG 'ORA$OEM_LEVEL_0' DATABASE;  
}
```
- This strategy is for 1 day recovery window only
- Shouldn't be used if there is configured recovery window bigger than 1 day
- Instead this use 'UNTIL TIME' clause in RECOVER COPY statement

# Oracle-Suggested Backup Strategy

## Schedule Oracle-Suggested Backup: Destination

Database **orcl102.ad.tlogica.com**  
Backup Strategy **Oracle-Suggested Backup**

Cancel Step 1 of 4 Next

Select the destination media for this backup.

Disk

Use disk as the only storage for backups

Tape

Use tape as the only storage for backups

Both Disk and Tape

Use disk to store the most recent database backup and archive logs for fast complete recovery. Use tape to store older backups for extended recovery window.

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- **Change Tracking File**
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- Oracle Secure Backup
- Miscellaneous - RMAN

# Fast Incremental Backups with **Change Tracking File**

- Usually Incremental backups are slower
- All blocks in a datafile are scanned
- This can result in bigger backup window
- In 10g can be used a Change Tracking File to perform faster incremental backups
- Tracks the physical location of all database changes

# Change Tracking File

- RMAN uses it to find the location of changed blocks since the last incremental backup
- Only these blocks will be scanned and included in the backup
- Improves significantly incremental backup strategies
- Can make an incremental backup up to 20 times faster

# Change Tracking File

- Change Tracking is disabled by default
- Has minimal performance overhead
- Enabling:
  - `SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;`
- Uses the Database Area by default:
  - `DB_CREATE_FILE_DEST`
- The size of the Change Tracking File is around 20 MB

# Starwood Hotels – Results

Oracle Open World 2005 - Session Id: 1181

- **Before**
  - **8 TB** Data Warehouse
  - Backup takes **19 hours** to complete
  - Incremental backups have size **500 GB**
- **Strategy**
  - Incrementally updated backups
  - Change Tacking File
- **After**
  - Backup takes **2 hours** to complete
  - No need to perform full backup every Sunday
  - Faster recovery

# Fast Recovery with Disk-Based Strategy

- Recovery strategy in case of media failure:
  - Backup:
    - Incrementally updated backups in FRA
  - Recovery
    - RMAN> **SWITCH DATABASE TO COPY;**
    - RMAN> RECOVER DATABASE;
  - Start the database by using its image copy, multiplexed control file and online redo logs from FRA
  - Take backup immediately after the recovery operation

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- **Compressed Backups**
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- Oracle Secure Backup
- Miscellaneous - RMAN

# Less Space for Backups with **Compressed Backups**

- In 10g backups can be compressed with RMAN
  - RMAN> BACKUP **AS COMPRESSED** BACKUPSET DATABASE;
- Only for backupsets, not for image copies
- 1/5 of the size of the original files
- RMAN automatically uncompress during recovery operations
- The speed of backup & recovery operations depends on the CPU resources

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- Compressed Backups
- **Simplified RESETLOGS**
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- Oracle Secure Backup
- Miscellaneous - RMAN

# Simplified Recovery Through RESETLOGS

- Before 10g - after every RESETLOGS a full backup must be performed immediately. Oldest archived logs was not usable
- In 10g there is ability to recover to an oldest incarnation. No need to perform full backup after every RESETLOGS
- The most important change is the new archived logs format:

```
LOG_ARCHIVE_FORMAT="log%t_%s_%r.arc"
```

# Simplified Recovery Through RESETLOGS

First archived log from Incarnation 17: **ARC00001\_0587681349.001**

Last archived log from Incarnation 16: **ARC00281\_0587468099.001**

```

RMAN> run {
2> allocate channel oem_disk_backup1 device type disk;
3> set until time='13.04.2006 19:56:00';
4> restore database;
5> recover database;
6> alter database open resetlogs;
7> }

released channel: ORA_DISK_1
released channel: ORA_SBT_TAPE_1
allocated channel: oem_disk_backup1
channel oem_disk_backup1: sid=156 devtype=DISK

executing command: SET until clause
released channel: oem_disk_backup1
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of set command at 04/13/2006 20:38:30
RMAN-20207: UNTIL TIME or RECOVERY WINDOW is before RESETLOGS time
RMAN> reset database to incarnation 16;

database reset to incarnation 16
```

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- Physical Backup & Recovery Enhancements
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- Oracle Secure Backup
- Miscellaneous - RMAN

# Preview and Validate Backups

- To preview backups and commands that will be used in a subsequent restore operation:
  - RMAN> RESTORE DATABASE **PREVIEW**;
  - RMAN> RESTORE ARCHIVELOG FROM TIME 'SYSDATE-7' **PREVIEW**;
- To test backups in a hypothetical recovery situation:
  - RMAN> **RESTORE** CONTROLFILE **VALIDATE**;
  - RMAN> **RESTORE** DATABASE **VALIDATE**;
  - RMAN> **RESTORE** ARCHIVELOG FROM TIME 'SYSDATE-7' **VALIDATE**;
  - RMAN> **RESTORE** DATABASE ARCHIVELOG FROM TIME 'SYSDATE-7' **VALIDATE**;

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- **New RMAN Views**
- Easy Tablespace Transport
- Oracle Secure Backup
- Miscellaneous - RMAN

# New Views for RMAN (10g R2)

## **V\$RMAN\_BACKUP\_JOB\_DETAILS**

**V\$BACKUP\_ARCHIVELOG\_DETAILS**

**V\$BACKUP\_ARCHIVELOG\_SUMMARY**

**V\$BACKUP\_CONTROLFILE\_DETAILS**

**V\$BACKUP\_CONTROLFILE\_SUMMARY**

**V\$BACKUP\_COPY\_DETAILS**

**V\$BACKUP\_COPY\_SUMMARY**

**V\$BACKUP\_DATAFILE\_DETAILS**

**V\$BACKUP\_DATAFILES\_SUMMARY**

**V\$BACKUP\_PIECE\_DETAILS**

**V\$BACKUP\_SET\_DETAILS**

**V\$BACKUP\_SET\_SUMMARY**

**V\$BACKUP\_SPFILE\_DETAILS**

**V\$BACKUP\_SPFILE\_SUMMARY**

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- **Easy Tablespace Transport**
- Oracle Secure Backup
- Miscellaneous - RMAN

# Cross-Platform Transportable Tablespaces

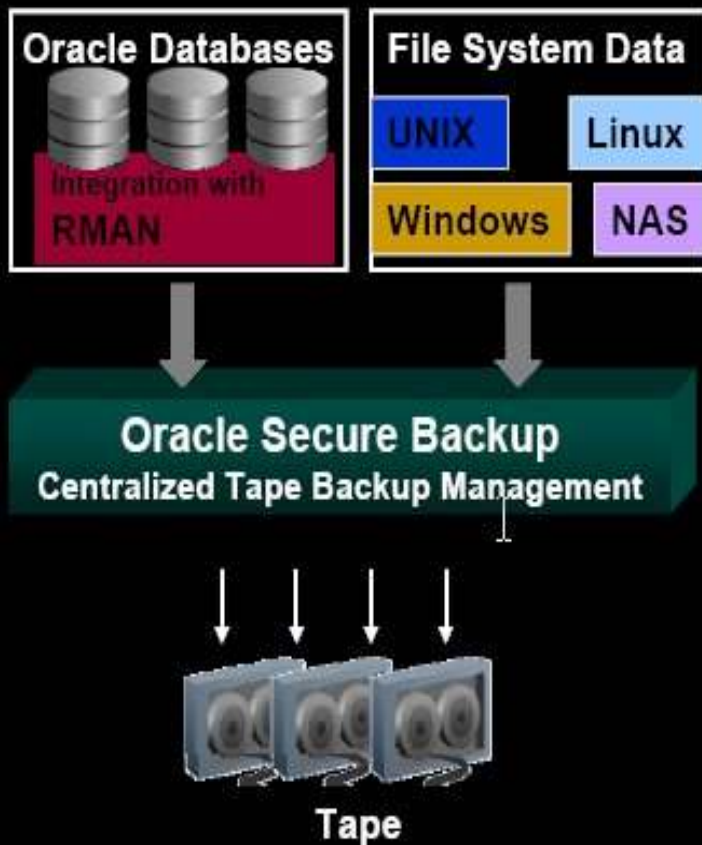
- Easily move tablespaces between “Little Endian” and “Big Endian” OS platforms:
- RMAN> **CONVERT** TABLESPACE USERS  
**TO PLATFORM** 'Solaris[tm] OE (32-bit)'  
DB\_FILE\_NAME\_CONVERT  
'E:\oracle\oradata\orcl102', 'E:\oracle\oradata\';
- *V\$TRANSPORTABLE\_PLATFORM*



# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- **Oracle Secure Backup**
- Miscellaneous - RMAN

# Oracle Secure Backup (10g R2)



- Media Management Software coming from Oracle
- Centralized backup management
- Security policy based
- Supports most of tape devices
- Encryption to tapes
- Expected in May 2006

# Agenda

- Flash Recovery Area
- Flashback Technology Enhancements
- Space Pressure & Database Hangs
- **Physical Backup & Recovery Enhancements**
- Conclusion
- Incrementally Updated Backups
- Change Tracking File
- Compressed Backups
- Simplified RESETLOGS
- Preview and Validate Backups
- New RMAN Views
- Easy Tablespace Transport
- Oracle Secure Backup
- **Miscellaneous - RMAN**

# Miscellaneous - RMAN

- (10g R2 EE) - Dynamic Channel Allocation over the instances in RAC
- (10g R2) - RMAN Channel Failover between instances in RAC
- (10g R2 EE) - RMAN encrypted backups
  - RMAN> **CONFIGURE ENCRYPTION ALGORITHM**  
`'algorithm_name'`;
- Backup database to image copies:
  - RMAN> **BACKUP AS COPY** DATABASE;

# Miscellaneous - RMAN

- (10g R2) - Transportable Tablespace Sets from Backups
  - Before 10g R2 tablespaces can be transported only if they are in read-only mode during tablespace transport process
  - In 10g R2 there is capability tablespaces to be transported from backups instead of data files
  - Can be very useful for OLTP systems

## Miscellaneous - RMAN

- (10g R2) - RMAN excludes **empty data blocks** from backups. It can do this even in MOUNT stage
- (10g R2) - Automatic Tempfile Creation after Media Recovery with RMAN
- Automatic Recovery of files that have never been backed up:
  - RMAN> **RESTORE DATABASE;**
  - Needs all archived redo logs from the time of the data file creation till restore point

# Miscellaneous - RMAN

- Drop the database with simple command:
  - RMAN> **DROP DATABASE;**
  - RMAN> **DROP DATABASE INCLUDING BACKUPS;**
- Cataloging whole directories with RMAN
  - RMAN> **CATALOG START WITH**  
**'D:\my\_backups';**
- Specifying Limits for Backup Duration:
  - RMAN> **BACKUP DATABASE DURATION 07:00;**
  - RMAN> **BACKUP DATABASE DURATION 07:00**  
**MINIMUM LOAD|TIME;**

# Miscellaneous - RMAN

- Cross-Platform Transportable Databases
  - A Whole Database can be transferred between different OS platforms
  - RMAN command: **CONVERT DATABASE**
  - Restriction: Target and Destination Platform must be with same endian format

# User-Managed Backups

- In 10g the whole database can be put in Online Backup Mode:
  - SQL> ALTER DATABASE **BEGIN BACKUP;**
  - SQL> ALTER DATABASE **END BACKUP;**

# Conclusion

- Unified Storage for a Disk-Based Backup Strategy
- Fast recovery from Human Errors with Powerful Flashback Features
- Faster Physical Backups
- Faster Recovery from Media Failures
- Integration with Enterprise Manager

- A Backup is only as good as the Recovery it can provide

# Questions & Answers

URL: <http://dba-blog.blogspot.com>

Email: [Radoslav.Rusinov@gmail.com](mailto:Radoslav.Rusinov@gmail.com)