Předpoklady – architektura – důležité části

- log files
- paměťové struktury
  - database buffer cache
  - redo log buffer
- synchonizace
  - log sequence number
  - checkpoint number
- procesy
  - log writer
  - checkpoint
  - archiver
  - database writer
- módy provozu databáze
  - nearchivní
  - archivni
- parameter file

Zpracování DML příkazu

Uživatelský proces pošle žádost o zpracování DML příkazu (například UPDATE). Server proces jej zpracuje v těchto krocích:

1. Data, která se mají měnit je třeba dostat do **Database buffer cache**. Buď tam jsou z předchozího zpracování, nebo je nahraje z datových souborů. Při tomto procesu může být nastartován **database writer**, aby se uvolnila cache.
2. Na data, která mají být měněna je aplikován **výlučný zámek** (implicitně se zamyká na úrovni řádky).
3. Do UNDO bloků se zkopírují stará data (jsou označena číslem transakce, která je mění).
4. Do **redo log bufferu** se vygeneruje **změnový vektor**, který obsahuje stará i nová data a ID transakce.

Po skončení DML operace

se změněná data stále nachází v paměti v database bufferu (v datových a undo blocích) a v redo log bufferu.

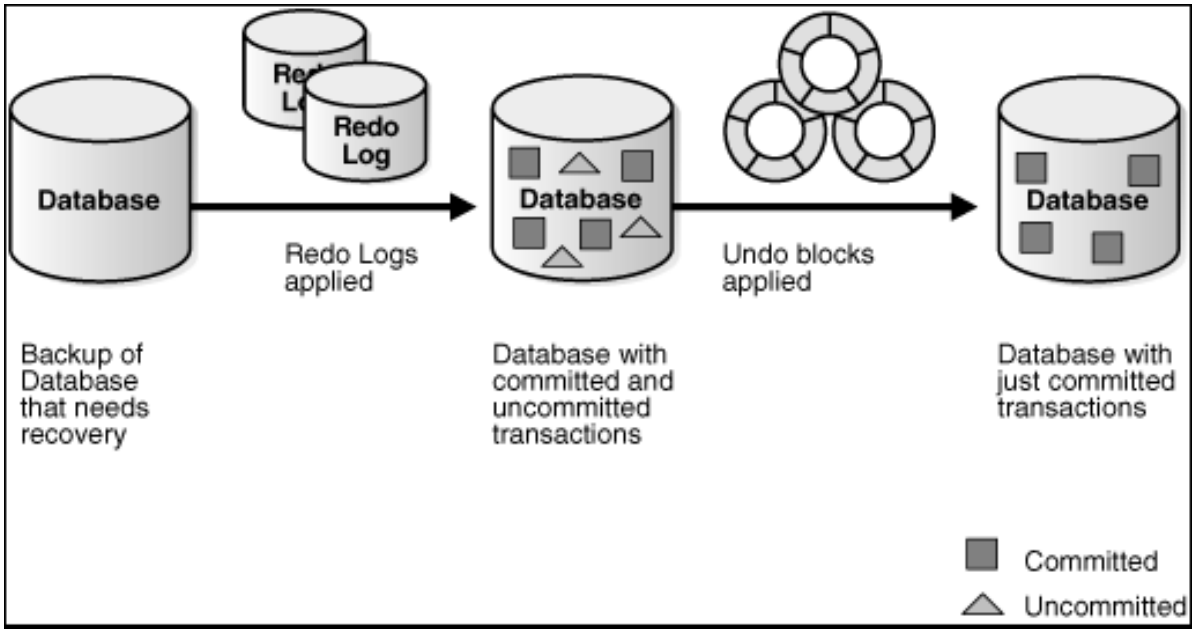
Zpracování příkazu COMMIT

1. Do redo log bufferu je vytvořen zápis, že příslušná transakce zavolala COMMIT.
2. Log writer zapíše na disk celý obsah redo log bufferu (jsou tam informace o ukončených i neukončených transakcích, protože do redo log bufferu se zapisuje sekvenčně a transakce jsou souběžné).
3. Když je obsah red log bufferu na disku (v log souboru), dostane uživatel hlášení “transaction complete”.
4. Bloky v UNDO segmentech, které obsahovaly stará data z ukončené transakce jsou uvolněny pro přepsání.
5. Jsou uvolněny zamčené záznamy v datových blocích (v database buffer cache).

Dojde-li v této chvíli k pádu instance.

Provede se po dalším restartu automaticky **instance recovery**. Úspěšně ukončené transakce budou obnoveny z redo log souboru.

Instance Recovery Picture



Instance Recovery Process

1. Data files out of sync
   Při (re)startu instance se zjistí, že synchonizační známky (SCN) v hlavičkách DB souborů a řídících souborů. Další analýza potom rozhodne, zda pro recovery proces budou stačit online redo log soubory (**instance recovery**) nebo je třeba začít s archivovanými žurnály (**media recovery**)
2. Roll forward (redo)
   Začne se od nejstaršího SCN (v hlavičkách DB souborů) a postupně se přehraje celá transakční aktivita.
3. Committed and noncommitted data in files
   Došlo k obnově datových i undo bloků do stavu těsně před pádem.
4. Roll back (undo)
   Na transakce, které nebyly v době pádu instance ukunčené, je třeba aplikovat standardní rollback operaci.
5. Committed data in files
   Databáze je synchronizovaná. Požadavky **Atomicity** a trvalosti (**Durability**) z **ACID** jsou splněny.

Categories of database failures

- Statement failure
- User process failure
- Network failure
- User error
- Instance failure
- Media failure

Zálohování – klasifikace

- fyzická záloha (data files, control files, log files)
  - příkaz copy (cp) z OS
  - pomocí nástroje Recovery Manager (RMAN)
    - image copy
    - backup piece
- “logická” záloha
  Pomocí utilit exp a imp (resp. datových pump - data pumps).
  Zálohovat lze na úrovni jednotlivých tabulek, uživatelských schémat a celé databáze.

Fyzickou zálohu lze provádět buď za běhu databáze (**hot backup**) nebo při skončené (a synchronizované) databázi (**cold backup**).

Přepnutí databáze do archivního módu

Zabrání přepisu redo log souboru před zazálohováním.
Automacká archivace (archiver) může a nemusí běžet.

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT EXCLUSIVE;
SQL> ALTER DATABASE ARCHIVELOG
SQL> ALTER DATABASE OPEN;
```

Po přepnutí do archivního módu by mělo následovat shutdown a **full (cold) backup**.
Příkaz **archive log list** v sqlplus informuje o stavu módu databáze a automatické archivaci.

Fyzická záloha pomocí příkazu copy

Cold backup

1. uzavření databáze
   (`shutdown normal | transactional | immediate`)
2. záloha všech souborů (řídící, datové, žurnály)
3. startup
4. Je dobré zálohovat také parameter file a password file.

Hot backup

Databáze musí být v achivním módu.
Záloha se provádí postupně pro tablespaces.

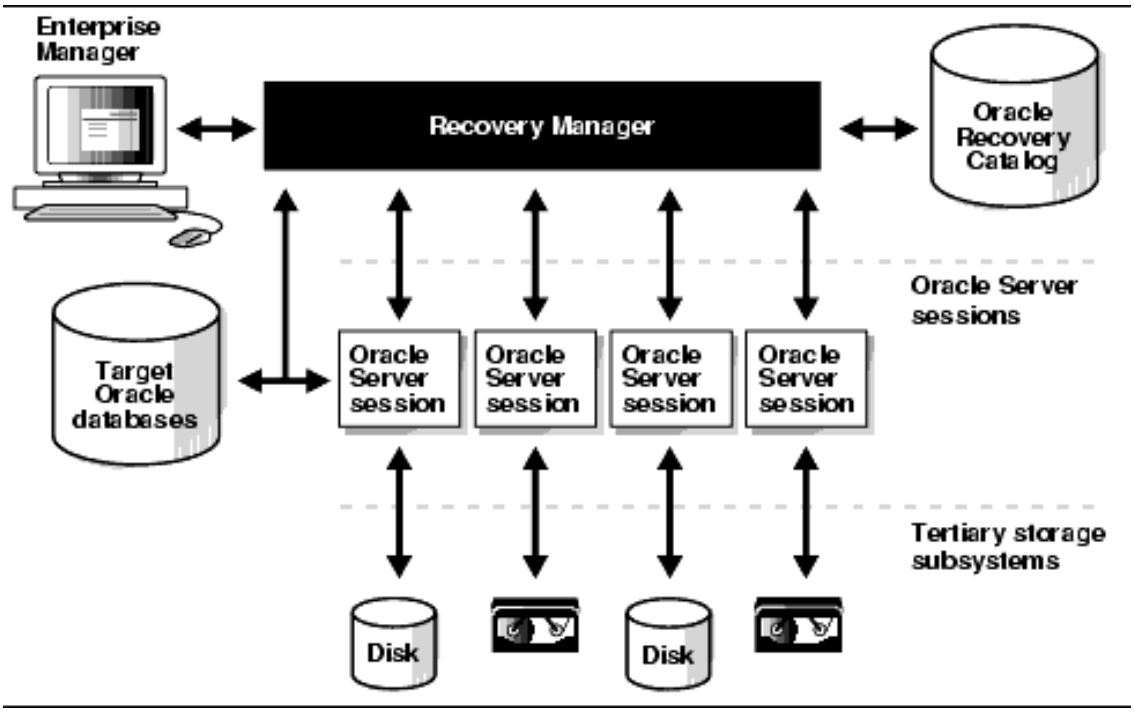
1. `alter tablespace tblsp_name begin backup;`
2. záloha souborů patřících k tablespace na úrovni OS.
3. `alter tablespace tblsp_name end backup;`

V průběhu zálohy se zvýší množství dat zapisovaných do žurnálu.

Recovery Manager

- Komplexní nástroj pro zálohování, katalogizaci záloh, vystavení souborů ze zálohy a obnovení databáze.
- Vlastní prostředí (rman) s vlastním skriptovacím jazykem pro zálohování a obnovu.
- Umožňuje interaktivní přístup i spouštění uložených skriptů.
- Zálohování – **image copy** nebo **backup piece**
- Podporuje inkrementální a kumulativní backup.
- Podporuje online (hot) backup.
- Při zálohování provádí kontrolu konzistence datových bloků.
- Operace `RESTORE` a `RECOVER`.
- Nepovinný katalog. Dotazy nad katalogem (`REPORT`, `LIST`).

Recovery Manager Figure



RMAN - ukázka obnovy tablespace

```
RMAN>RUN {
2>BACKUP AS BACKUPSET
3>FORMAT '/u01/db01/backup/%d_%s_%p'
4> DURATION 10:00 MINIMIZE LOAD
5>(DATABASE);
6>SQL 'alter system archive log current';
7>}
```

RMAN – katalog – LIST

```
RMAN> LIST BACKUP OF DATABASE;

RMAN> LIST BACKUP OF DATAFILE
2>   "/db01/ORADATA/u03/users01.dbf";

RMAN> LIST COPY OF TABLESPACE "SYSTEM";

RMAN> LIST COPY OF DATABASE ARCHIVELOG
2> FROM TIME='SYSDATE-7';
```


RMAN – katalog – REPORT

```
RMAN> REPORT NEED BACKUP incremental 3;

RMAN> REPORT NEED BACKUP days 3;

RMAN> REPORT NEED BACKUP redundancy 2;

RMAN> REPORT NEED BACKUP
2>    recovery window of 3 days;
```

RMAN - ukázka obnovy tablespace

```
run{
sql "ALTER TABLESPACE inv_tbs OFFLINE IMMEDIATE";
RESTORE TABLESPACE inv_tbs;
RECOVER TABLESPACE inv_tbs DELETE ARCHIVELOG;
sql "ALTER TABLESPACE inv_tbs ONLINE";
   }
```


RMAN – incomplete recovery

```
RMAN> RUN {
2> SET UNTIL TIME = '2005-11-28:11:44:00';
3> RESTORE DATABASE;
4> RECOVER DATABASE;
5> ALTER DATABASE OPEN RESETLOGS; }
```

Flashback – charakteristika

Jedná se o “doplněk” k možnostem “klasického” zálohování databáze. Přináší celkem “nedatabázové rysy”:

- recycle bin
- nahlížení postupných změn záznamu přes několik transakcí

- “fyzický flashback”
  - technologie paralelní k žurnálům
  - náročné na další prostor
  - rychlejší a koncepčně snazší návrat zpět v čase (rewind)
- “logický flashback”
  - založen an `UNDO` segmentech
  - parametr `undo_retention`
  - obecně není tolik náročný na přídavný prostor

Logický flashback – ukázka

Jaká byla hodnota řádku v minulosti?

```
SELECT * FROM employees AS OF TIMESTAMP
TO_TIMESTAMP('2004-04-04 09:30:00',
             'YYYY-MM-DD HH:MI:SS')
WHERE last_name = 'Chung';
```

Logický flashback – ukázka

Jak se vyvíjela hodnota řádky během zpracování?

```
SELECT versions_startscn, versions_starttime,
versions_endscn, versions_endtime,
versions_xid, versions_operation,
name, salary
FROM employees
VERSIONS BETWEEN TIMESTAMP
TO_TIMESTAMP('2003-07-18 14:00:00',
             'YYYY-MM-DD HH24:MI:SS')
AND TO_TIMESTAMP('2003-07-18 17:00:00',
                 'YYYY-MM-DD HH24:MI:SS')
WHERE name = 'JOE';
```


Logický flashback – ukázka

Kdo může za změny a jak to vrátit zpátky?

```
SELECT xid, operation, start_scn,commit_scn,
       logon_user, undo_sql
FROM flashback_transaction_query
WHERE xid = HEXTORAW('00020003000002D');
```

Recycle bin

Patří taktéž do kategorie “logický flashback”.

- příkaz DROP TABLE ve skutečnosti tabulku pouze přejmenuje.
- návrat je možný pomocí příkazu:
  `FLASHBACK TABLE <table_name>`
  `TO BEFORE DROP [RENAME TO <new_name>];`
- vysypat koš lze příkazem:
  `PURGE [USER_|DBA_]RECYCLEBIN`
- prohlížení koše je možné přes pohledy:
  `DBA_RECYCLEBIN` nebo `USER_RECYCLEBIN`
- příkaz DROP je rozšířen o klauzuli PURGE
  `DROP TABLE <table_name> [PURGE];`

Fyzický flashback – nastavení

Je velmi náročné na diskový prostor.
Doporučuje se dvojnásobem velikosti DB.

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT EXCLUSIVE;
SQL> ALTER SYSTEM SET
2 DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE OPEN;
```

Fyzickýflashback – ukázka

Převinutí databáze zpět.
Rychlejší alternativa k PITR.

```
RMAN> FLASHBACK DATABASE TO TIME =
2> "TO_DATE('2004-05-27 16:00:00',
3> 'YYYY-MM-DD HH24:MI:SS')";

RMAN> FLASHBACK DATABASE TO SCN=23565;

RMAN> FLASHBACK DATABASE
 2> TO SEQUENCE=223 THREAD=1;
```