Backup/Recovery Strategy and Impact on Applications

Jacek Wojcieszuk, CERN IT

Database Deployment and Persistancy Workshop

October, 2005

Outline

- Backup and recovery overwiew
- Current backup strategy overview
- Future improvements
- Exemplary recovery scenarios
- Consequences for applications

Types of failures

- Instance Failure
  - Usually connected with an Oracle process failure
- Media Failure
  - Disk failure, storage array controller failure etc.
- Block Corruption
  - Usually caused by bugs in Oracle software
- Human error
  - In most cases accidentally deleted/updated data
  - Database user or DBA
- Disaster
  - Fire, flood, earthquake, plane crash etc.

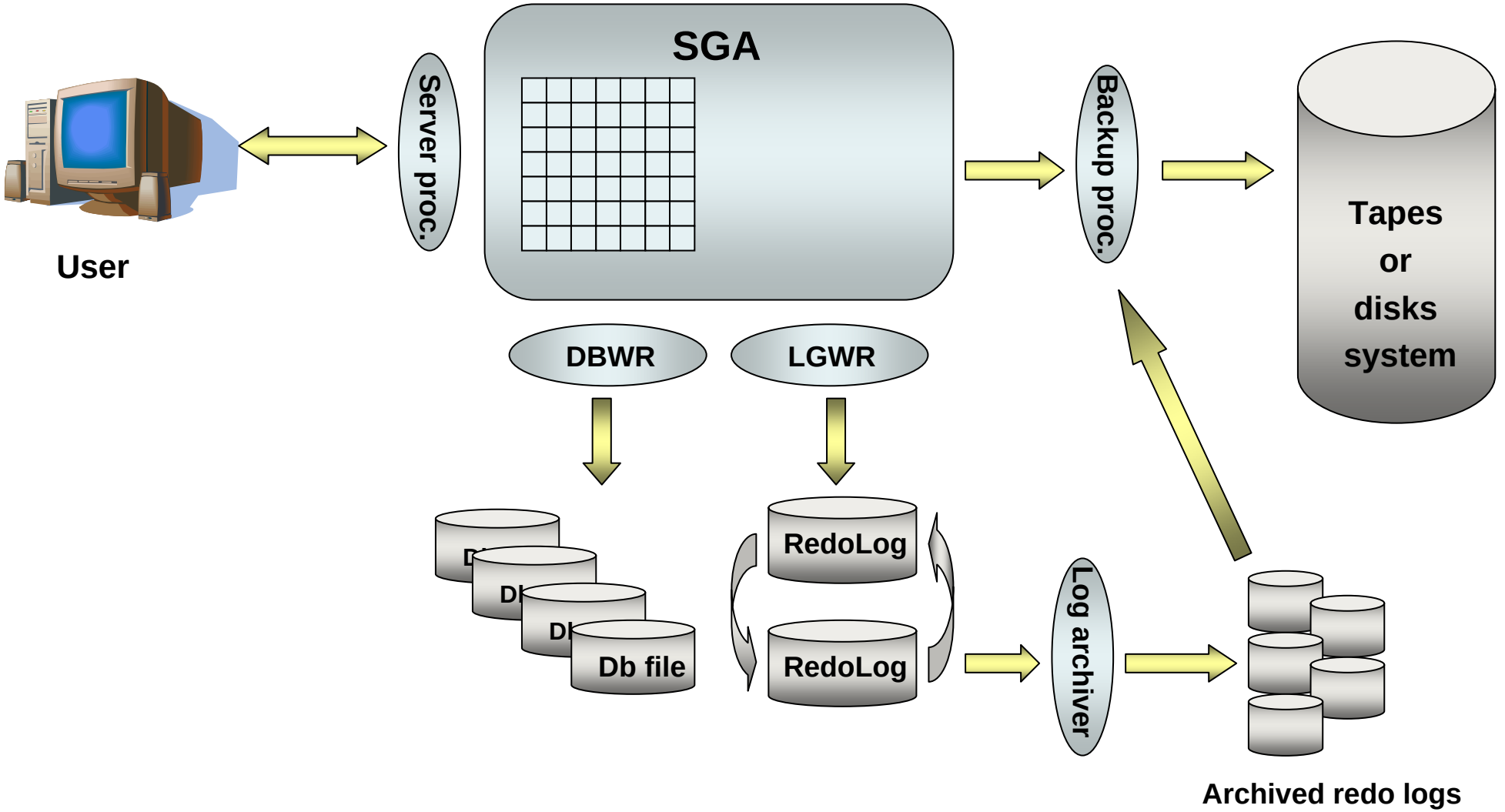
Backup options and tools

- Physical backups
  - Cold (off-line) backups
    - Full database only
    - Require downtime
    - Do not provide flexibility for point in time recovery (PITR)
  - Hot (on-line) backups
    - Different types of backups: full, incremental (cumulative, differential), archivelogs
    - Different scopes: full database, tablespace(s) or datafile(s)
    - Do not require database downtime
    - Can be used to recover full database, single/multiple tablespace(s)/datafile(s) or a corrupted block
    - Database can be recovered to any point in time within assumed backup retention period
  - All backups mentioned above
    - Can be performed with Oracle Recovery Manager (RMAN)
    - Can be send to disk or tapes

- Logical backups
  - Logical copy of data in the database
  - Support for different granularity
  - Can be taken either with legacy Export/Import tools or with Data Pump (10g)

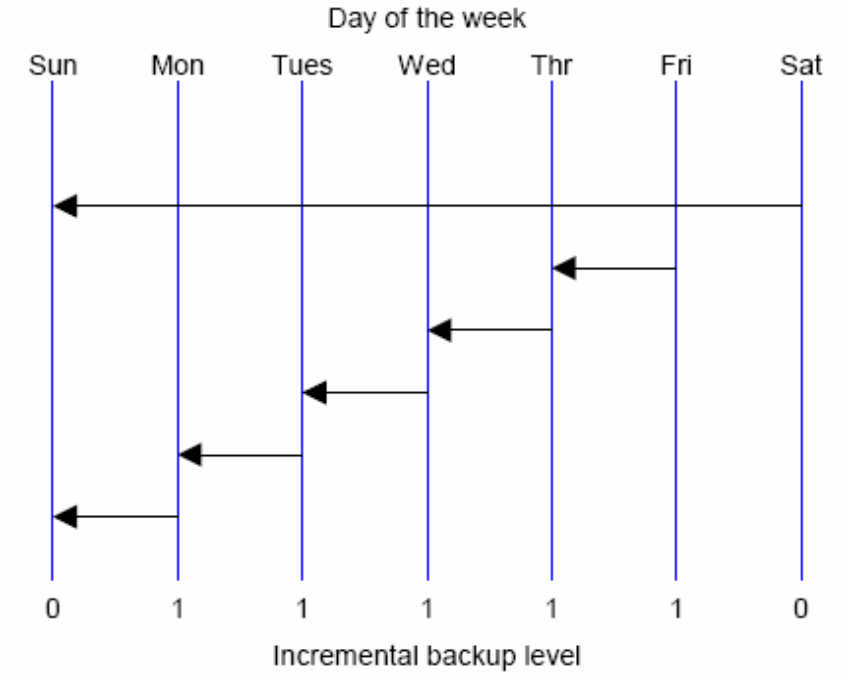
- Standby database (Data Guard)
  - Physical or logical

How hot backups are possible

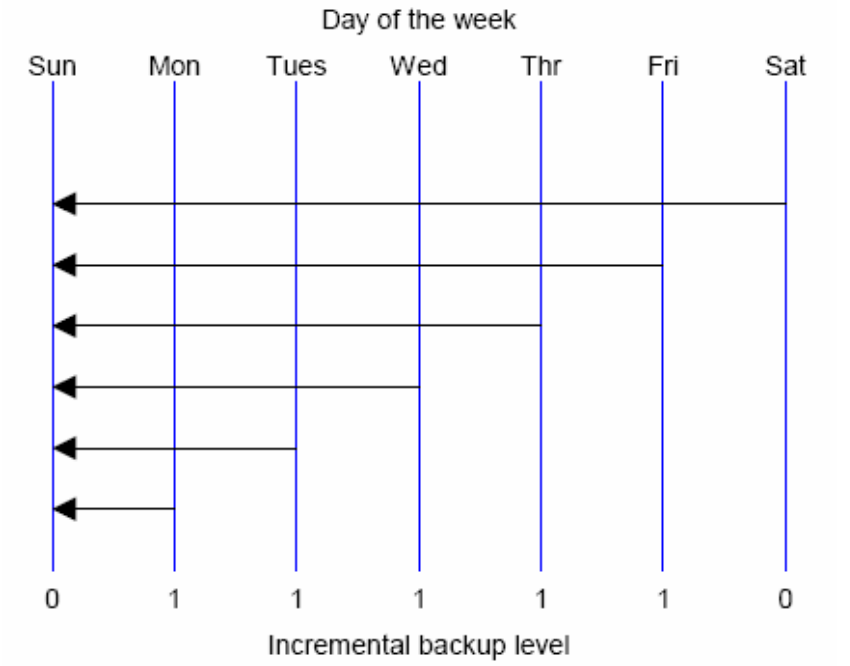


Different types of RMAN hot backups

- Full database backup
- Incremental backups (different levels 0-4)
  - Cumulative, differential
  - Can be used to update a full backup
- Archivelog backups
- Tablespace(s), datafile(s) backups



Differential incremental



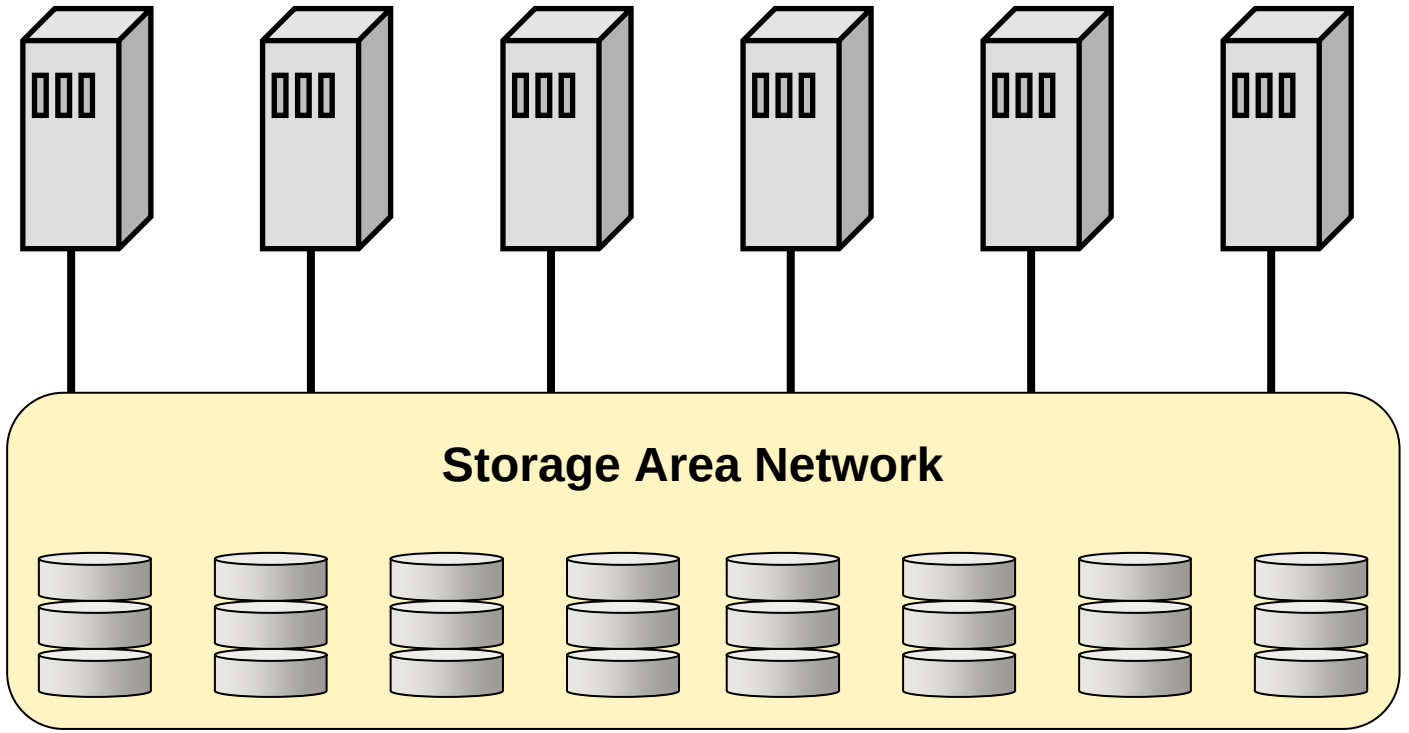
Cumulative incremental

Currently used backup strategy

- Strategy
  - Backups to tapes only
  - Full backup monthly or weekly depending on the database
  - Daily incremental differential backups
  - Backups of archivelogs every 10, 20 or 60 minutes (depending on the database)
  - 2 most recent full backups kept on tapes
  - Backups validation on regular basis
- Advantages
  - Relatively simple solution
  - Even in case of disaster only up to 60 minutes of transactions can be lost
- Disadvantages
  - Tapes are much slower then disks so database recovery time can be an issue
  - Tapes are less reliable then disks

Future improvements – disk backups (1)

- Are rather difficult to implement in case of databases running on single-node disk servers
- Nowadays are more and more popular thanks to getting cheaper disk space – make better use of available resources
- Much easier to maintain with Oracle 10g thanks to the Flash Recovery Area feature



Future improvements – disk backups (2)

- Backups to disk:
  - One full database backup at the beginning of database existence
  - Block change tracking
  - Daily incremental backups used to update the full database backup
  - Archived redo logs
- Additionally backups to tapes for disaster recovery:
  - Monthly full backups
  - Weekly incremental cumulative
  - Daily incremental differential
  - Hourly archivelog backups
  - 2 most recent full backups + subsequent incremental and archivelog backups kept to allow point in time recovery
  - Regular, automated test recoveries
- Elements of backup policy that should be agreed with experiments:
  - Backup frequency
  - Backup retention

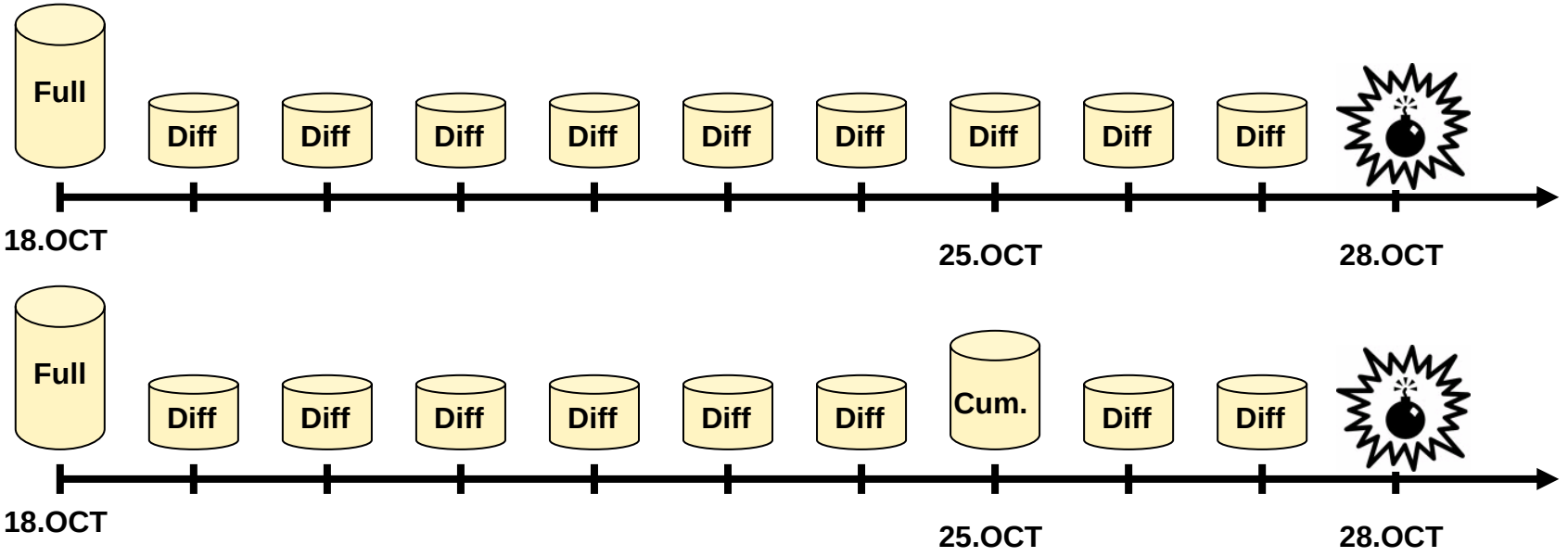
B&R improvements – pros and cons

- Advantages
  - Much shorter recovery time then in case of tape backups
    - Transfers between disks are usually an order of magnitude faster then transfers from tapes to disks
    - In case of emergency database can be started almost immediately directly from the disk backup (after recovery of recent transactions from archivelogs)
  - Tape backups necessary only for disaster and point in time recovery
  - Load on the tape system will decrease due to less often archivelog backups
  - Block change tracking will decrease database load

But: disk space consumption will double

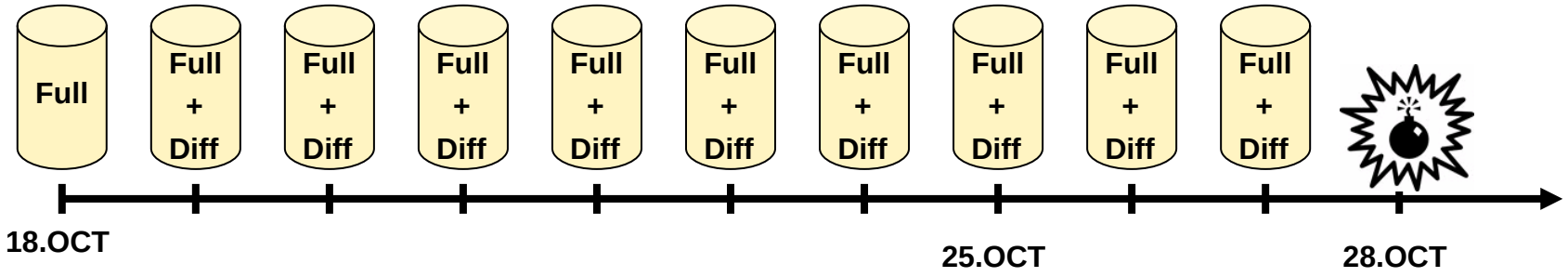
Failure and recovery scenarios (1)

- Broken disk
  - No transaction loss and no downtime thanks to disk mirroring
- Broken controller in a storage array
  - No transaction loss and no downtime thanks to mirroring between 2 independent storage arrays
- Disaster
  - Service downtime
  - Recovery of the database from tape backups with average speed of 50 GB/hour
  - Up to 60 minutes of transactions can be lost (depending on the database)



Failure and recovery scenarios (2)

- Failure of both storage arrays used by a database
  - Service interruption
  - Currently used tape backups:
    - Recovery time proportional to database size, age of the most recent full backup and number of transactions
    - Backups needed: full backup + n incremental backups + m archivelogs
    - Up to 60 minutes of transactions lost (depending on the database)
  - Disk backups:
    - Recovery time proportional to database size and number of transactions (usually 5-10 times shorter then in case of tape backups)
    - Backups needed full: backup + m archivelogs)
    - **For really big and critical databases it is possible to recover recent transactions from archived redo logs and startup of the database from backup copy (database can be back in productions in less then 1 hour)**
    - Up to 60 minutes of transactions can be lost



Handling human errors

- Erroneously deleted/updated data
  - Flashback query e.g:

```
INSERT INTO employee
(SELECT * FROM employee AS OF TIMESTAMP
to_timestamp('18-OCT-05 08:50','DD-MON-YY HH24:MI'));
```

  - Flashback transaction query e.g:

```
SELECT operation, undo_sql
FROM flashback_transaction_query
WHERE table_name = 'EMPLOYEES'
  AND start_timestamp >= to_timestamp('18-OCT-05 08:50','DD-
  MON-YY HH24:MI');
```

  - More information: https://edms.cern.ch/document/673184

- Accidentally dropped table:
  - Point in time recovery
  - Recovery from a logical backup

B&R strategies – impact on applications (1)

- Downtime and data loss
  - Mirroring and B&R strategies minimize probability of unscheduled downtime and data loss
  - Unfortunately they are still possible and appropriate handling on applications level is vital
    - E.g. applications shouldn't corrupt part of data they are able to access in case of unavailability of the other part
    - Applications should be able to deal with transaction loss in case of disaster
      - The only possible workaround would be deployment of a standby database which means doubling hardware and maintenance effort and decreased performance
- Human errors
  - Applications should minimize probability of human errors
  - Users should react quickly to their mistakes
    - Recovery of table erroneously cleared 3 hour ago is usually much simpler and much faster that of one cleared last week

B&R strategies – impact on applications (2)

- Backup frequency:
  - Affects amount of data that can be lost during disaster (e.g. more frequent archivelog backups == smaller data loss)
  - Has impact on recovery time (more frequent full/incremental backups == shorter time to recovery)
  - Frequent backups are most expensive in terms of resources (CPU, disk/tape space)
  - **We need feedback from application managers regarding desired backup frequency**

- Backup retention
  - Affects data safety
  - Affects possibility to perform point in time recovery
  - Affects resource (tape space) consumption
  - **We need feedback from application managers regarding desired backup retention**

- **Read-only data**
  - **All the data that is not going to change in the future should be placed in a separate tablespace and declared as read-only**
  - **Read-only tablespaces can be ommited during subsequent full backups**
  - **Declaring data as read only decreases probability of accidental deletes/updates**

Questions & Answers