DATA PROTECTION

David López
v 2.2.2
Updated Spring 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Storage tiers (not an standard classification)

- Tier I: Mission-critical data
- Tier II: Vital data
- Tier III: Sensitive data
- Tier IV: Non-critical

Sometimes access frequency is also took into account for tiers

Defined by the enterprise but also by two important parameters:

- RPO (Recovery Point Objective)
- RTO (Recovery Time Objective)

- A Service Continuity must be defined for major incidents

RPO & RTO

When an incident occurs, data must be recovered

- One disk can be lost (and can be recovered from RAID 1, 5, 6, …)
- Selected old information must be recovered (backup, snapshots, …)
- Most, or the whole system must be recovered
  - It requires an off-site backup

Recovery Point Objective: measures the maximum time of data that can be lost (not amount, but time)

Recovery Time Objective: measures the time it takes to recover the service

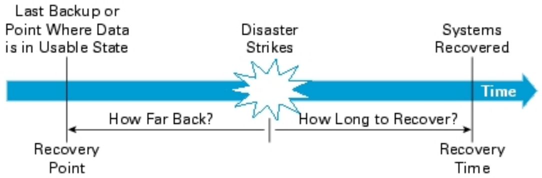


Image from essential-net.co.uk

Data tiers

	RTO	Solution?
Mission critical	immediate	Local mirror
Vital	seconds	External mirror, snapshot
Sensitive	minutes	RAID reconstruction, Tape
Non-critical	Hours?	Tape, historical archive

Business metrics

- RPO & RTO are "technical" concepts
- If you deal with business people, they use other metrics:
- **Risk Analysis (RA)**: what could possibly go wrong?
  - A disk fails, two disks fails, three disks fails…
  - The network is down
  - A terrorist attack
  - Earthquakes, hurricanes
  - What have I do to prevent / recover from these disasters?
- **Business Impact Analysis (BIA)**: How much will it cost?
  - The main costs are usually due to business discontinuity
  - Calculate the cost of downtime
  - The Risk Analysis can give you an idea of how much can cost to prevent or reduce this downtime
    - So, invest in security is a question of trade-offs

A real example

Amazon.com
- Sales 4th quarter 2010: $13,000,000,000
  - Source: http://www.auctionbytes.com/cab/cab/abn/y11/m01/i28/s01
- 1 quarter = 2,190 hours
- Downtime cost = 6,770,833 $/hour
  - … but this is assuming an equally probable sales distribution
  - … the days previous to Christmas can triplicate this approximation

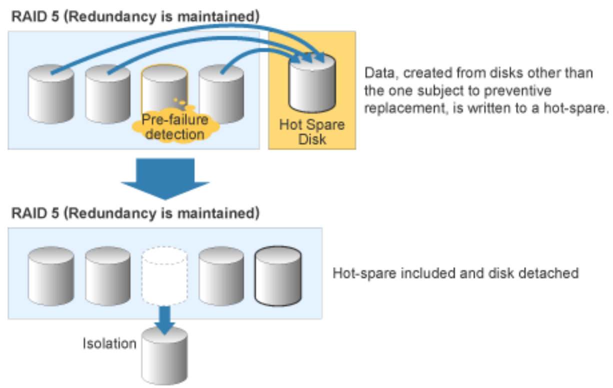
Cost of downtime

Application	Cost of downtime per hour	Annual losses with downtime of 1% (87.6 hrs/yr)	0.5% (43.8 hrs/yr)	0.1% (8.8 hrs/yr)
Brokerage operations	$6,450,000	$565,000,000	$283,000,000	$56,500,000
Credit card authorization	$2,600,000	$228,000,000	$114,000,000	$22,800,000
Package shipping services	$150,000	$13,000,000	$6,600,000	$1,300,000
Home shopping channel	$113,000	$9,900,000	$4,900,000	$1,000,000
Catalog sales center	$90,000	$7,900,000	$3,900,000	$800,000
Airline reservation center	$89,000	$7,900,000	$3,900,000	$800,000
Cellular service activation	$41,000	$3,600,000	$1,800,000	$400,000
Online network fees	$25,000	$2,200,000	$1,100,000	$200,000
ATM service fees	$14,000	$1,200,000	$600,000	$100,000

Table borrowed from Hennessy & Patterson CA:AQA 5th Edition

Hot Spare Disk

- Redundant disk
- Pre-failure detection
  - Warning to maintain team
  - Reconstruction (the new one will be the new hot spare)



SSD cache

- Specially for peak access to slow disk (HDD)
- There are two different caches, one cache for reads, and one cache for writes

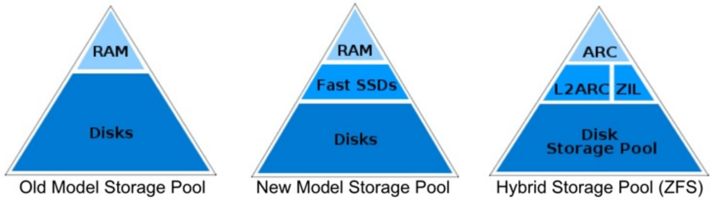


Image borrowed from thestoragearchitect.com

Legal needs

Some data need to be stored for long time, but it is seldom accessed

- WORN (Write Once Read Never)
- WORO (Write Once Read Occasionally)

Sometimes is just historical data sorted for legal reasons

- Medical records usually 5 years (more than 50 years by country or regulation)
- Research & development 10 years
- Manufacturing Quality Assurance 15 years
- Drug research 30 years
- Broadcasting content (volunteer) 50 years

An example to think about: UPC historical records

Data corruption: causes

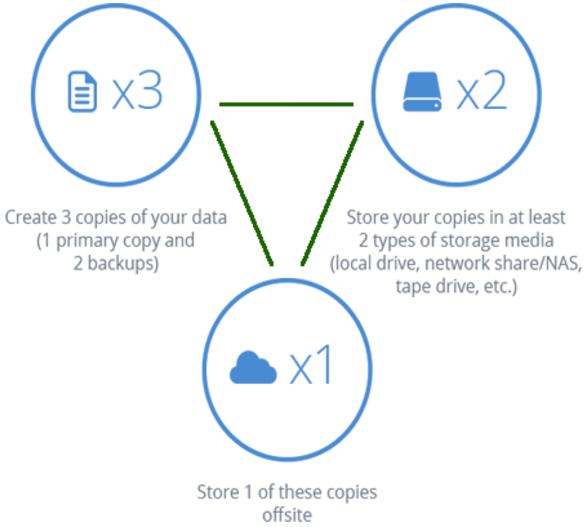
- Hardware failures
  - Complete/ partial disk failure
  - Spikes in power, erratic arm movements, scratches in media (bit rot)
- Software failures
  - Firmware bugs (modern drives contains hundreds of thousands of lines of firmware code)
  - "Wild writes" (bad location) and "phantom writes" (reported as done, but it never reached the disk)
  - Memory corruption (e.g. cosmic rays)
- Big disasters
  - E.g. natural disasters, malicious attacks
- Human errors
  - E.g. remove the wrong file

- The biggest problems is "silent corruption": an undetected data corruption
  - CERN study 1 in 10^6 bits

Data corruption: solutions

- Checksums
  - Stored or updated on disk during write operations and read back to verify the block during reads
- Redundancy
  - In on-disk structures, like RAID (but RAID only works if you know which one is the wrong block)
  - Remote mirroring
- Backup
  - On-site and off-site
  - High cost: it requires a complete analysis
- Snapshots and other techniques

Backup concepts the 3-2-1 rule of thumb



Backup concepts

Full backup
- Makes a copy of all data (but not the redundant data)
- Can be "real" or "synthetic" (we will see more in a few slides)

Incremental backup
- Idea: to Identify and record all files that have changed since the last full backup
- Smaller and quicker than full backup
- Two (main) types:
- Differential Backup
  - All the differences since the last incremental backup are stored
  - A full restoration will be slower, since all increments will be restored
  - If one copy fails, the restoration fails
- Cumulative backup
  - Stores all the differences since the last full backup (sometimes also the last cumulative backup)
  - Requires more resources (and grows in size)
  - Faster recovery

A full backup is required

Restoration after a disaster starts with the last full backup
- If it is too old, much time is required to fully restore (differential)
- Incremental is too big (cumulative)

Having a recent full backup is important
- But it cost resources… a trade-off is required
- One solution is *synthetic backup*
  - A new full backup is generated based on the last full backup plus incremental backups
  - Is made out-of-the-server, so it does not interfere in normal operations (but a dedicated disk server is required when the full backup is generated)

An off-site backup is required! (At least, a full backup… depending on your recovery point objective – RPO)
- Sometimes (physically) moving tapes
- Sometimes using the net

Backup problems: frozen data

Backups require a frozen image of the data while the application remains on line and produces new data
- Usually called the **quiesce** operation: pausing or altering the running processes of a computer, in order to guarantee a consistent and usable backup

There are several solutions:
- Cold backup: data are locked and not available to users
  - Good for full backups in some business
- Replication using a Business Continuity Volumes (BCV) /Shadow copy
- Snapshots
  - Be careful: there are several (and very different techniques) behind the "snapshot" concept
- Continuous Data Protection

Business Continuity Volume (BCV) aka Shadow Copy

Main idea: you have a copy of disks just for full backup

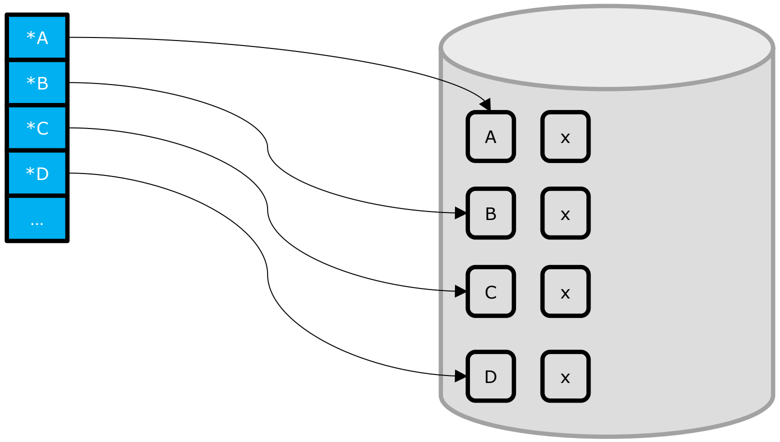
- But you don't need to duplicate your disks!
  - For instance, imagine you have RAID 51 (aka "the RAID for the truly paranoid") with 10 2TB disks
- You can copy ALL data with a maximum of 4 2TB data
  - You don't need RAID levels for the copy!
  - Can be an external disk / service
- You start to synchronize disks (all writes in both systems, while copying the rest): this is the establish operation
- Once the disk is synchronized, you froze the copy (split operation)
- Advantages?
  - You have a full in-site backup in disks (fast recover)
  - Once splitted, you can perform a tape / offsite copy of the backup
  - Some kind of incremental backup is required

Snapshots (Point in time copies) concept

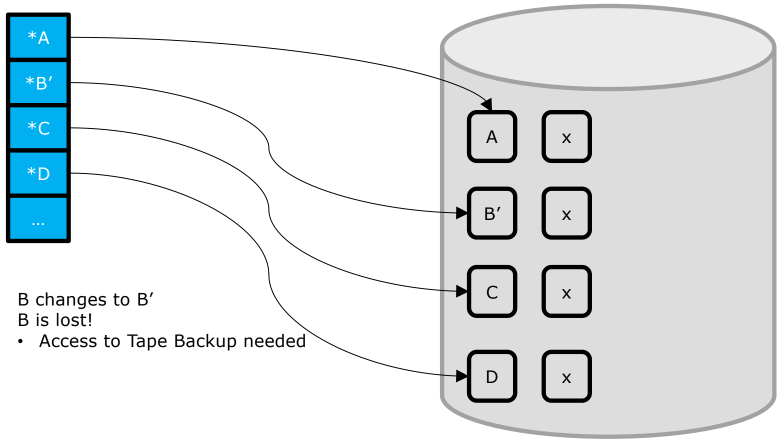
- A storage snapshot is a **set of reference markers**, or pointers, to data stored on a disk drive, on a tape, or in a storage area network (SAN)
- A snapshot is something like a detailed table of contents, but it is treated by the computer as a complete data backup
- Snapshots streamline access to stored data and can speed up the process of data recovery
- Use copy-on-write techniques

- Advantages: snapshots are small and fast

Snapshots

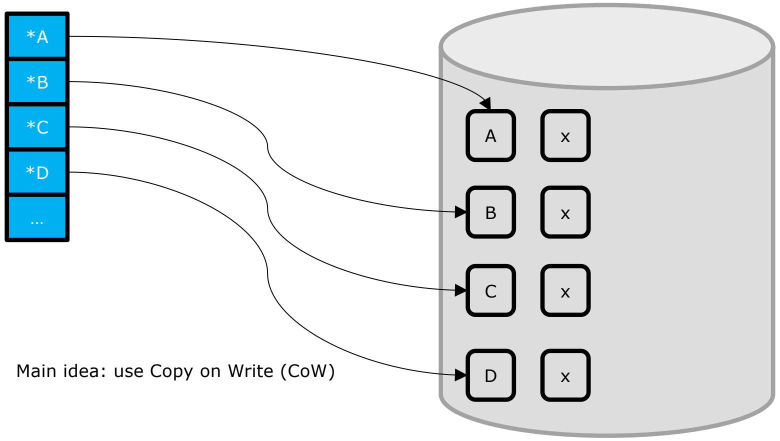


Snapshots



B changes to B'
B is lost!
- Access to Tape Backup needed

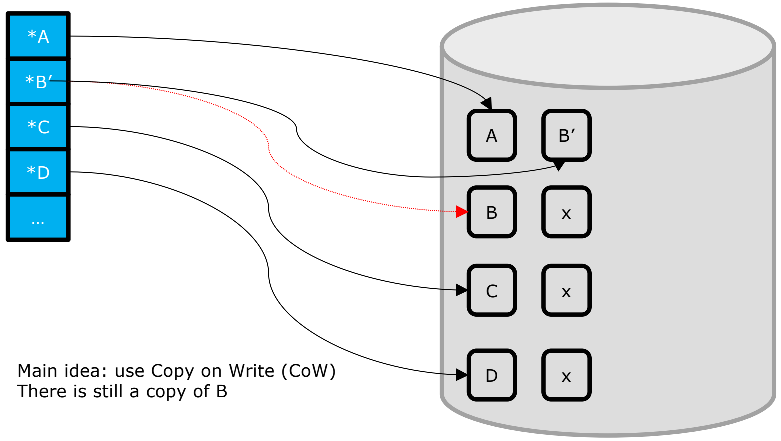
Snapshots



Main idea: use Copy on Write (CoW)

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

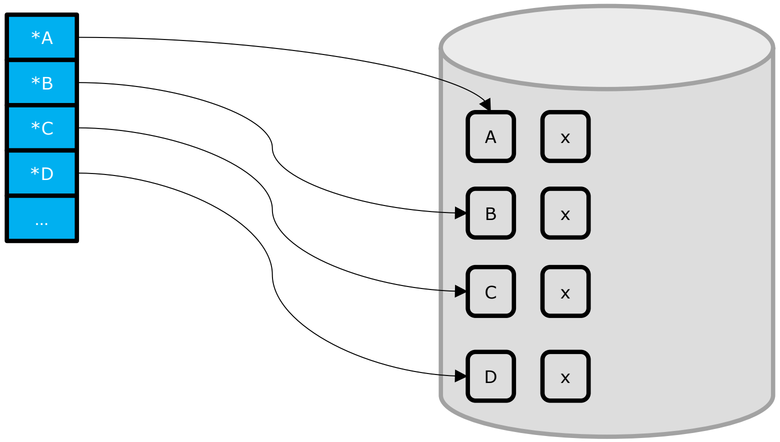
Snapshots



Main idea: use Copy on Write (CoW)
There is still a copy of B

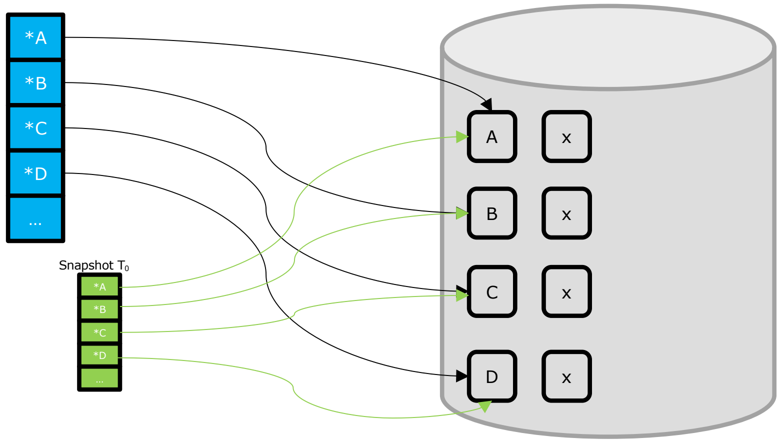
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Snapshots Example: After a tape full backup



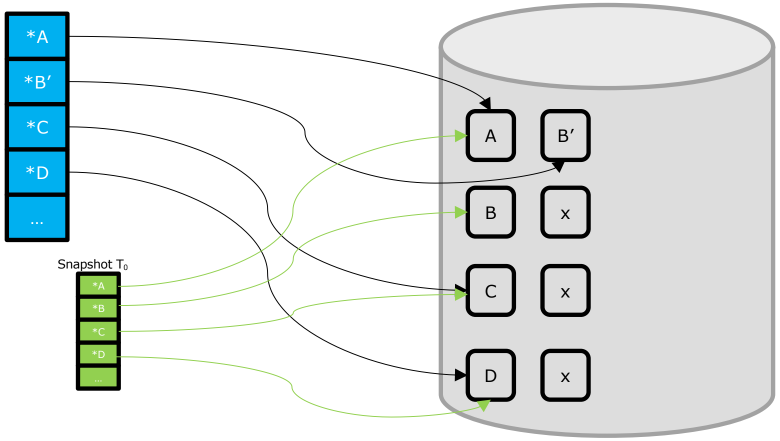
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Snapshots Example: Time T0: a copy of the block pointers is made



UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Snapshots Example: Write Block B (becomes B')



Snapshot Limitations

Number of available free blocks

- Blocks Three states: busy, free, CoW
- 10% free needed

Helps fast recovery (no need for tape access)

Does not substitute other kind of Backup

Continuous Data Protection (aka CDP)

Also know as **Real-time backup**

- Every change is automatically saved (asynchronously written) to a separate storage location, usually another computer over the network
  - Can be encrypted, in case of an external, rented backup center
- Adds overhead on every write
- CDP creates an electronic journal of complete storage *snapshots*, one storage snapshot for every instant in time that data modification occurs
  - Be careful: this is not the "snapshot" concept, thus it stores data blocks too
- The record of the changes is available for users, so they can recover a previous version of the file
- Some restrictions:
  - Usually not all files, but the specified ones (e.g. you can exclude temporal files)
  - You also need a complete backup for big disasters
- Differs from mirroring in that it enables a roll-back of the log, and thus restoration of old image data

Checksums for data integrity

- Using checksums for on-disk blocks
  - It can use a Fletcher-based checksum or a SHA-256 hash thorough the file system tree
- Checksums are kept separate from the corresponding blocks by storing them in the parent blocks
  - Uses a generic block pointer structure
  - This block contains the checksum of the block it references
  - Before using a block, the system calculates its checksum and verifies it against the stored checksum in the block pointer
    - If it fails, it can be reconstructed using RAID, mirroring, ..
  - The checksum hierarchy forms a self-validating Merkle-tree
- This technique permits to detect data corruption such as bit rot, phantom writes and misdirects reads and writes

Replication for data recovery

- Besides using RAID (we will see it later), some systems maintain replicas for some “important” on-disk blocks
  - For instance, the checksums previously explained
- By default, they store multiple copies of metadata, and single copies of data
  - Each block pointer contains pointers to up to three copies of the block been referenced (*ditto blocks*)
  - When a corruption is detected, the redundant copies are recovered

STORAGE

David López

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH