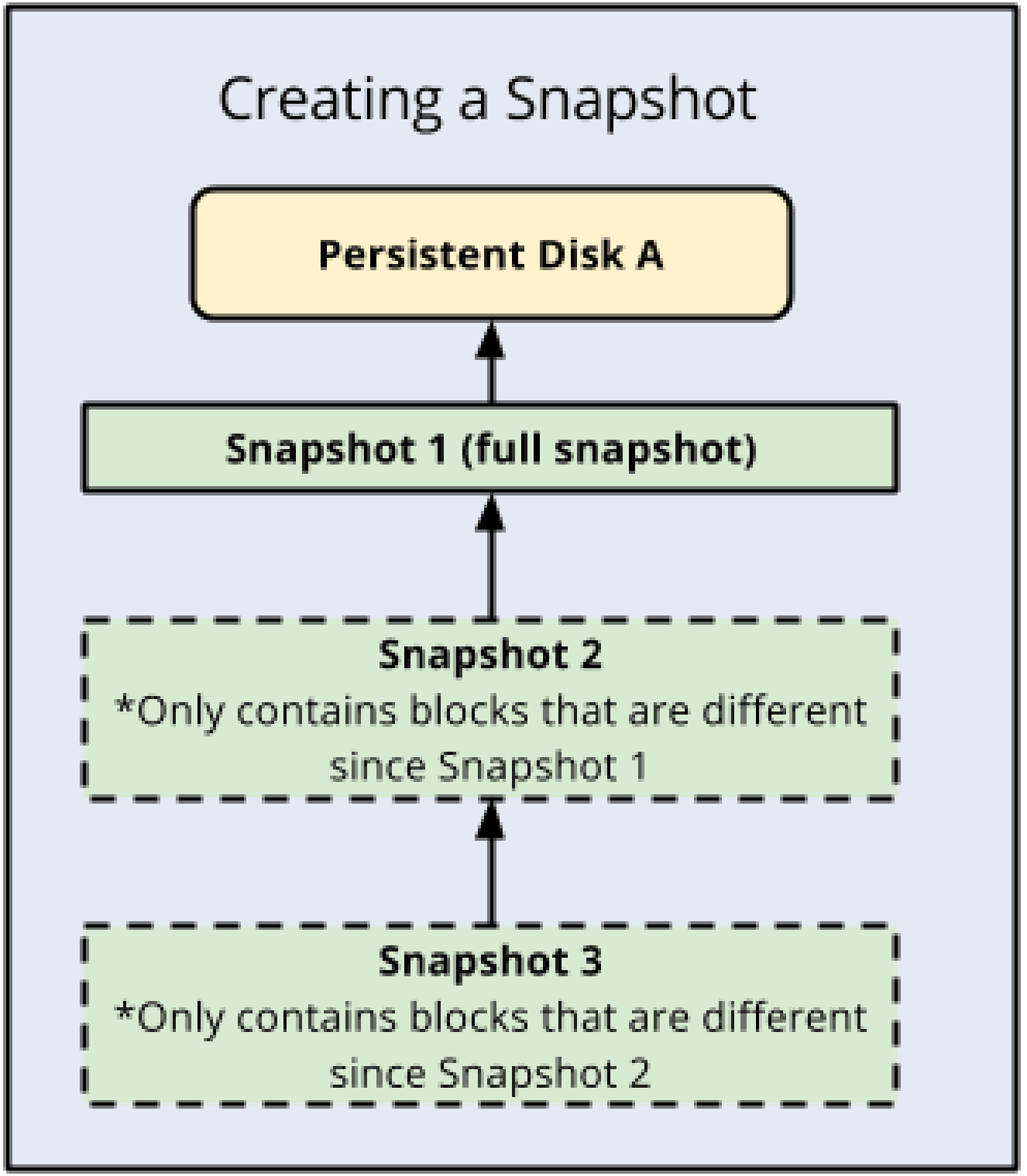
Snapshots to a File System

Definition:
A snapshot is a read-only, point-in-time copy of the filesystem state used for restoring data in the event of failure.

Snapshot vs Copying:
Snapshotting only requires as much disk space as the changed blocks require while copying requires the number of copies times the size of the file.


Creating a Snapshot
Persistent Disk A
Snapshot 1 (full snapshot)
Snapshot 2
*Only contains blocks that are different since Snapshot 1
Snapshot 3
*Only contains blocks that are different since Snapshot 2

Snapshot Technologies

	Copy-on-write	Redirect-on-write	Clone/ split mirror	COW w/back-ground copy	Incremental
Snapshot is tightly coupled to original data	Yes	Yes	No	Yes, until background copy finishes	Depends on how original snapshot is generated
Space efficient	Yes	Yes	No	No	No
Original data system IO and CPU resource overhead	High	Medium	Low	Low	Low
Write overhead on orig. data copy	High	None	None	High	High
Protects against logical data errors by rolling back to orig. copy	Yes	Yes	Yes	Yes	Yes
Protects against physical media failures of orig. copy	No	No	Yes	After background copy completes	Depends on underlying snapshot tech.

Challenges

The challenges that can occur with snapshots,
-- use the right kind of snapshots,
-- get their data in the right state prior to taking the snapshot, and
-- have something to control, manage, configure and report on the snapshot process.

Requirements

- Take snapshot of current filesystem ~ 10-20 GB of data.
- Add - add new files to the filesystem at various positions, for various users.
- Delete – delete few of the files from filesystem, for various users.
- Modify - modify some files entirely, replace a word completely in a file, change/delete few lines in a file.
- Copy a file from one directory to another.
- All the above made changes should reflect in your snapshot.
- Languages allowed: C/C++. Use of STL is allowed.
- You are not supposed to use ‘system()’ library functions and system commands like ls, cp, mv, mkdir etc. You have to implement your own versions of these commands.
- Consider all the edge cases and justify your approach for solving problems.