

Report: Snappy

Aishna Agrawal | Janvi Thakkar | Mihir Jain | Priyam Tongia

- **Introduction**

- In computer systems, a snapshot is the state of a system at a particular point in time. Snapshot is a copy of your files stored periodically.
- Snapshot and backup are different from each other.
- Backup can be stored in another location, same server, or even the same drive. Snapshots can be stored only in the same location where original data is present.

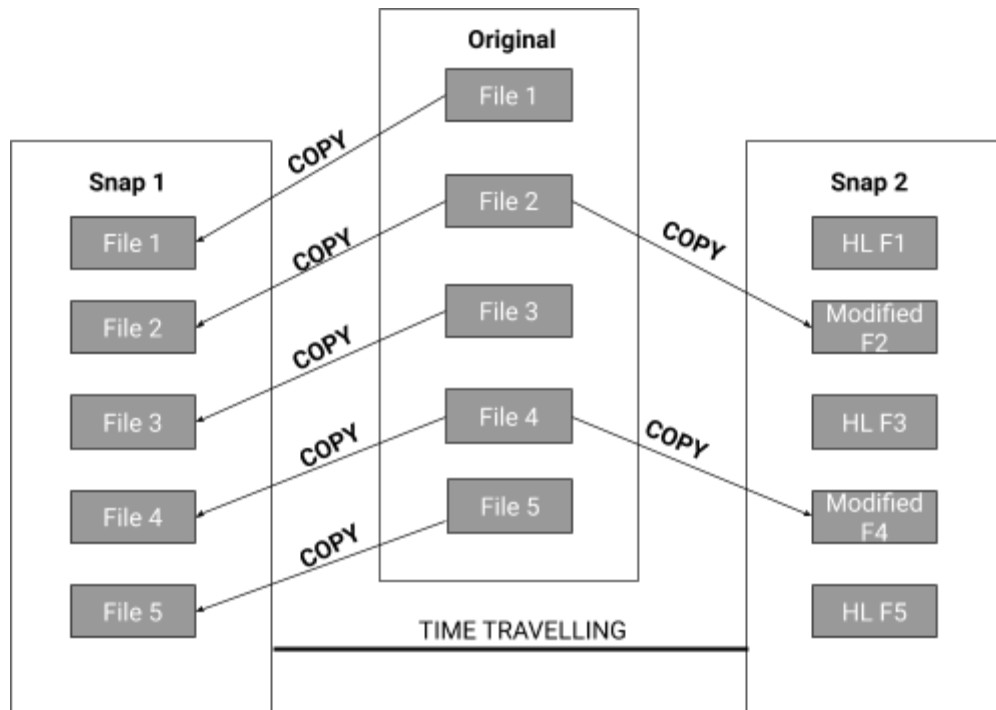
- **Need of the project / Motivation :**

- At times we lose our work progress due to human error or data corruption, and there the snapshot plays an important role in retrieving our data.
- Snapshots are generally created for data protection, but they can also be used for testing application software and data warehousing.

- **Our method / intuition / logic of work :**

- There are several methods to implement snapshot management
 - Copy-on-write → The first snapshot only saves meta-data about the files/directories. Before any write on the blocks containing these files/directories, the original block data gets copied to a designated position.
 - Redirect-on-write → similar to copy-on-write, but removes the double writing process, in this process the changed data is stored as a snapshot.
 - Log-Structured → Keeps track (log) of all changes made to a file. In reverting, the transactions of the log file are run in reverse order.
 - Split mirror → creates an identical copy of the file every time changes are made.
- And according to our needs, we decided to move forward with the split mirror method, with some improvisation (as it is seemingly feasible and easy to implement).
- The split mirror creates an identical copy of the data.

- Basically we would create a copy of our source directory for the first time and the subsequent snapshots will be the hard links to the previous versions, a copy will only be created if there are any changes found in the source.
- Reverting to a snapshot will involve overwriting the source directory with the contents of that snapshot.



- **Code Structure :**

- Storage folder: Contains snapshots of all the entries
 - Root directory created with the name given by hashing which contains all the snapshots of the directory with 0 based indexing
 - Config.csv: Contains mapping of a natural number to the path
-

- **Basic commands implemented :**

- help - gives details about all the command in our project
- create - create snapshot of the input directory, optional flag -d to add description
- revert (undo change) - revert to some older version of snapshot

- cleanup - remove all the snapshots and store the snapshot of the latest version
 - compare - lists the difference between the contents of folder/files
 - listsnap - list out the snapshots taken of the directory
 - logsnap - displays the changes done to the directory
 - delete - delete a specific snapshot of a directory
 - status - lists out the modified or added file in 2 folders
 - desc - attach a description to a snapshot
- **Analysis / limitations / extension**
 - Snapshot creation for files does not make sense, because everytime we make changes a copy of the file will be created (which will take a lot of space).
 - Our current method of implementation (split mirror technique) is not very efficient in terms of storage as compared to other techniques, and thus can be improvised.
 - This project is in working condition for linux and thus can be extended for different operating systems.
 - **Related Work**
 - Several companies such as amazon, ibm, vmware have their own snapshot management.
 - One of the open source available is snapper.io (<https://github.com/openSUSE/snapper>)

Our Github repository: <https://github.com/jvt3112/Snappy>
