

HFDS Snapshots

Table of contents

1 Overview.....	2
1.1 Snapshottable Directories.....	2
1.2 Snapshot Paths.....	2
2 Snapshot Operations.....	3
2.1 Administrator Operations.....	3
2.2 User Operations.....	3

1 Overview

HDFS Snapshots are read-only point-in-time copies of the file system. Snapshots can be taken on a subtree of the file system or the entire file system. Some common use cases of snapshots are data backup, protection against user errors and disaster recovery.

The implementation of HDFS Snapshots is efficient:

- Snapshot creation is instantaneous: the cost is $O(1)$ excluding the inode lookup time.
- Additional memory is used only when modifications are made relative to a snapshot: memory usage is $O(M)$, where M is the number of modified files/directories.
- Blocks in datanodes are not copied: the snapshot files record the block list and the file size. There is no data copying.
- Snapshots do not adversely affect regular HDFS operations: modifications are recorded in reverse chronological order so that the current data can be accessed directly. The snapshot data is computed by subtracting the modifications from the current data.

1.1 Snapshottable Directories

Snapshots can be taken on any directory once the directory has been set as *snapshottable*. A snapshottable directory is able to accommodate 65,536 simultaneous snapshots. There is no limit on the number of snapshottable directories. Administrators may set any directory to be snapshottable. If there are snapshots in a snapshottable directory, the directory can be neither deleted nor renamed before all the snapshots are deleted.

1.2 Snapshot Paths

For a snapshottable directory, the path component *".snapshot"* is used for accessing its snapshots. Suppose `/foo` is a snapshottable directory, `/foo/bar` is a file/directory in `/foo`, and `/foo` has a snapshot `s0`. Then, the path

```
/foo/.snapshot/s0/bar
```

refers to the snapshot copy of `/foo/bar`. The usual API and CLI can work with the *".snapshot"* paths. The following are some examples.

- Listing all the snapshots under a snapshottable directory:

```
hdfs dfs -ls /foo/.snapshot
```

- Listing the files in snapshot `s0`:

```
hdfs dfs -ls /foo/.snapshot/s0
```

- Copying a file from snapshot `s0`:

```
hdfs dfs -cp /foo/.snapshot/s0/bar /tmp
```

The name ".snapshot" is now a reserved file name in HDFS so that users cannot create a file/directory with ".snapshot" as the name. If ".snapshot" is used in a previous version of HDFS, it must be renamed before upgrade; otherwise, upgrade will fail.

2 Snapshot Operations

2.1 Administrator Operations

The operations described in this section require superuser privilege.

2.1.1 Allow Snapshots

Allowing snapshots of a directory to be created. If the operation completes successfully, the directory becomes snapshottable.

- Command:

```
hdfs dfsadmin -allowSnapshot <path>
```

- Arguments:

path	The path of the snapshottable directory.
------	--

See also the corresponding Java API `void allowSnapshot(Path path)` in `HdfsAdmin`.

2.1.2 Disallow Snapshots

Disallowing snapshots of a directory to be created. All snapshots of the directory must be deleted before disallowing snapshots.

- Command:

```
hdfs dfsadmin -disallowSnapshot <path>
```

- Arguments:

path	The path of the snapshottable directory.
------	--

See also the corresponding Java API `void disallowSnapshot(Path path)` in `HdfsAdmin`.

2.2 User Operations

The section describes user operations. Note that HDFS superuser can perform all the operations without satisfying the permission requirement in the individual operations.

2.2.1 Create Snapshots

Create a snapshot of a snapshottable directory. This operation requires owner privilege of the snapshottable directory.

- Command:

```
hdfs dfs -createSnapshot <path> [<snapshotName>]
```

- Arguments:

path	The path of the snapshottable directory.
snapshotName	The snapshot name, which is an optional argument. When it is omitted, a default name is generated using a timestamp with the format " 's' yyyyMMdd-HH:mm:ss.SSS", e.g. "s20130412-151029.033".

See also the corresponding Java API [Path createSnapshot\(Path path\)](#) and [Path createSnapshot\(Path path, String snapshotName\)](#) in [FileSystem](#). The snapshot path is returned in these methods.

2.2.2 Delete Snapshots

Delete a snapshot of from a snapshottable directory. This operation requires owner privilege of the snapshottable directory.

- Command:

```
hdfs dfs -deleteSnapshot <path> <snapshotName>
```

- Arguments:

path	The path of the snapshottable directory.
snapshotName	The snapshot name.

See also the corresponding Java API [void deleteSnapshot\(Path path, String snapshotName\)](#) in [FileSystem](#).

2.2.3 Rename Snapshots

Rename a snapshot. This operation requires owner privilege of the snapshottable directory.

- Command:

```
hdfs dfs -renameSnapshot <path> <oldName> <newName>
```

- Arguments:

path	The path of the snapshottable directory.
------	--

oldName	The old snapshot name.
newName	The new snapshot name.

See also the corresponding Java API [void renameSnapshot\(Path path, String oldName, String newName\)](#) in [FileSystem](#).

2.2.4 Get Snapshottable Directory Listing

Get all the snapshottable directories where the current user has permission to take snapshots.

- Command:

```
hdfs lsSnapshottableDir
```

- Arguments: none

See also the corresponding Java API `SnapshottableDirectoryStatus[] getSnapshottableDirectoryListing()` in `DistributedFileSystem`.

2.2.5 Get Snapshots Difference Report

Get the differences between two snapshots. This operation requires read access privilege for all files/directories in both snapshots.

- Command:

```
hdfs snapshotDiff <path> <fromSnapshot> <toSnapshot>
```

- Arguments:

path	The path of the snapshottable directory.
fromSnapshot	The name of the starting snapshot.
toSnapshot	The name of the ending snapshot.

See also the corresponding Java API `SnapshotDiffReport getSnapshotDiffReport(Path path, String fromSnapshot, String toSnapshot)` in `DistributedFileSystem`.