Hewlett Packard Enterprise

Brochure

# SNAPSHOTS WITH HPE EZMERAL DATA FABRIC

## INTRODUCTION

The ability to create and manage snapshots is an essential feature expected from enterprise-grade storage systems. This capability is increasingly seen as critical with Big Data systems as well. A snapshot captures the state of the storage system at an exact point in time and is used for recovering data that was lost or corrupted due to application error, user error, or a malicious event.

## SNAPSHOT BENEFITS WITH HPE EZMERAL DATA FABRIC

Snapshots within a Big Data context are useful in both storage and compute. HPE Ezmeral Data Fabric provides consistent snapshots that offer the following benefits:

**Recovery from errors**
Operational and analytical applications manipulate the data in the distributed file system on behalf of the user or the administrator. Application-level errors or even inadvertent user errors can mistakenly delete data or modify data in an unexpected way. In this case, snapshots can be used to restore data quickly and easily.

**Managing real-time data analysis**
By using snapshots, query engines like Apache Drill can produce precise SQL query results against data sources subject to constant updates, such as sensor data or social media streams. As new SQL queries are written, either to achieve more specific data results or to improve performance of existing queries, if the underlying data is constantly changing, it is difficult to assess query improvement. Snapshots allow new queries to run against a specific point-in-time image of data to make accurate comparisons against existing queries. Snapshots help ensure the data for both queries is constant, even in a scenario where data is continuously being ingested in real-time.

**Machine learning model training**
Machine learning frameworks, such as TensorFlow, can use snapshots to get a known, unchanging, point-in-time view of an otherwise continuously changing data set. These snapshots allow for an auditable model training process, where ongoing updates to machine learning models can be run against a static baseline data set. This helps to identify actual improvements in the model, since the snapshot data remains unchanged.

**Verifying data lineage**
Data lineage helps businesses track the data sources, transformations, and destinations as the data flows through HPE Ezmeral Data Fabric. It is often used to identify the cause of any irregularities in a data set or report. However, to investigate an observed irregularity, auditors have to view the history of the data as it was, when the data was created. Snapshots create an immutable view of the data to help ensure the lineage cannot be corrupted from modifications.

**Compliance auditing**
Organizations often have to demonstrate they were in compliance with required standards over a certain time period. In such a scenario, snapshots can help businesses show an immutable view of data at a certain point in time to prove they were in compliance.

**Multivariate and A/B testing, and data versioning**
Snapshots can be used to create immutable views of the results of multivariate and A/B testing. This means the results can be compared and analyzed while being safeguarded against inadvertent or even malicious modifications.

**Online backups**
With snapshots, you can create a baseline for consistent backups. Snapshots can be created when the system is running, thus allowing you to create consistent backups without having to halt data updates in the system. Snapshots also allow you to backup and restore online HPE Ezmeral Data Fabric tables (using the CopyTable functionality).

## SNAPSHOTS IMPLEMENTATION

In HPE Ezmeral Data Fabric, a snapshot is a read-only image of a volume at a specific point in time. Snapshots are created at the volume level, not the entire cluster level, which means you can isolate them to specific segments of your data sets. You can create a snapshot in an ad-hoc manner or automate the process with a schedule. A snapshot takes almost no time to create and uses very little incremental disk space because they are not standard copies of data. They are implemented with a method known as redirect-on-write, which is very space-efficient. You can take a snapshot of a 1 PB cluster in seconds with no additional data storage required.

The redirect-on-write method provides protection without duplicating the data. It uses pointers to keep track of data blocks. If a block needs modification, HPE Ezmeral Data Fabric simply redirects the pointer for that block to another block and writes the new data there (that is, it redirects on writes). HPE Ezmeral Data Fabric keeps track of all the blocks that comprise a given snapshot. If a process accesses data in a given snapshot, it simply uses these pointers to access those blocks.

Here's a brief explanation of how redirect-on-write works:

**Time T0.** File F1 is written. It uses data blocks A, B, and C.

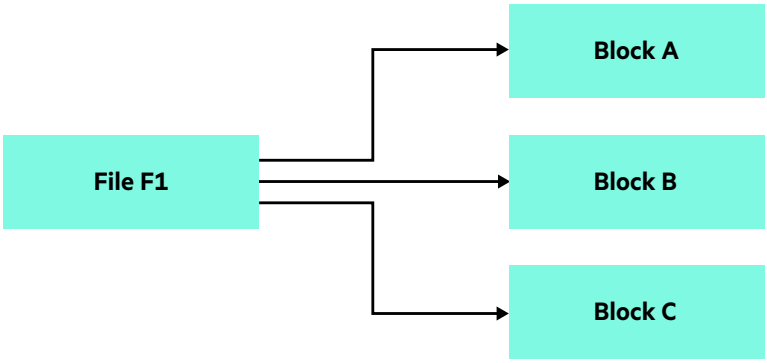


**FIGURE 1.** Data blocks A, B, and C are mutable

**Time T1.** We create snapshot S1. Data blocks A, B, and C are made immutable and S1 points to A, B, and C.

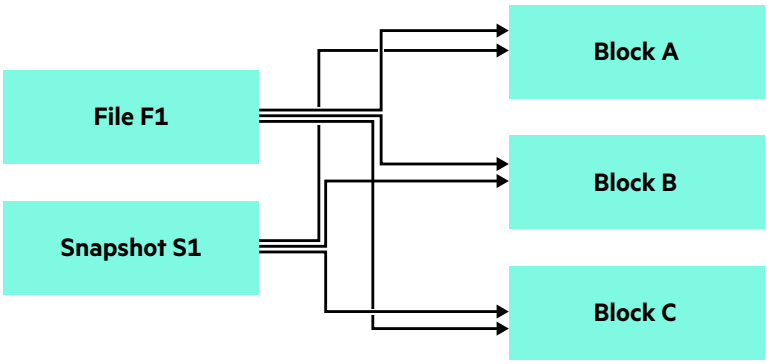


**FIGURE 2.** Data blocks A, B, and C are made immutable

**Time T2.** We need to make changes to F1. Since HPE Ezmeral Data Fabric supports full read-write capabilities, any part of the file can be updated. In this example, let's say the data in block C needs to be updated.

Since block C is immutable, we create a separate block C' and add the changes there. Snapshot S1 still points to data blocks A, B, and C. We can restore or view the contents of file F1 as they were at time T1. Current contents of F1 are held in data blocks A, B, and C'.

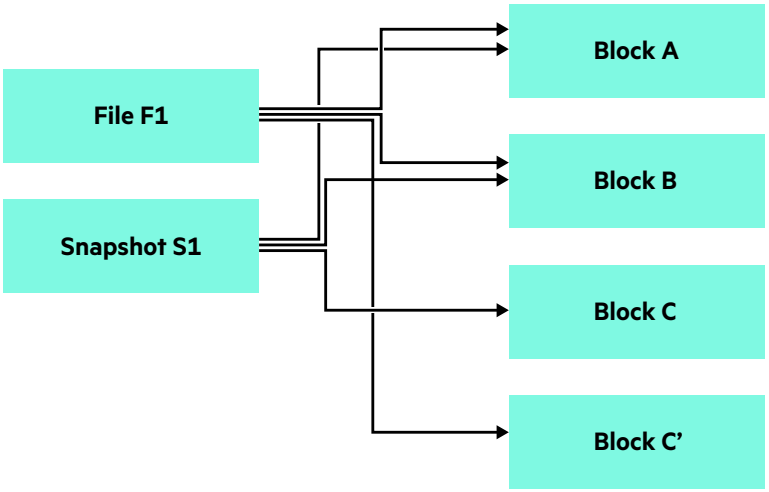


**FIGURE 3.** Data blocks A, B, and C are immutable; C' is mutable

**Time T3.** We create snapshot S2. Data block C' is made immutable, and snapshot S2 points to A, B, and C'.

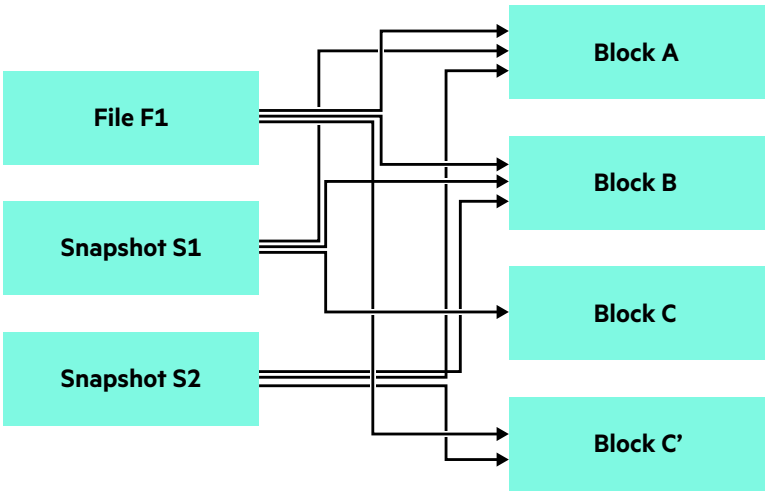


**FIGURE 4.** Data blocks A, B, C, and C' are immutable

We can now restore or view the contents of file F1 at timestamps T1 and T3 using snapshots S1 and S2, respectively. Note that there is no data duplication of blocks A and B between file F1 and snapshots S1 and S2. Only the data blocks that are changed after a snapshot is taken will require incremental storage space.

Another important fact is that the snapshots are implemented directly in the storage layer in an efficient and fast way. Any application that saves data in benefits from out-of-the-box snapshots. Moreover, since the snapshots are atomic and consistent, applications have the exact view of the data at the time that the snapshot was taken. This is not true on HDFS-based data platforms, as will be explained in the following section.

## WORKING WITH SNAPSHOTS

The following sections describe procedures associated with snapshots:

**Creating a snapshot:** You can create a snapshot in an ad-hoc manner or use a schedule to automate snapshot creation. Each snapshot created by a schedule has an expiration date that determines how long the snapshot is retained. When you schedule snapshots, the expiration date is determined by the retain parameter of the schedule.

**Viewing the contents of a snapshot:** At the top level of each volume is a directory called **.snapshot** containing all the snapshots for the volume. You can view the directory with Hadoop commands (for example, `hadoop fs -ls`) or by mounting the cluster with NFS to view the directory with operating system tools (such as the Linux® `ls` command).

**Viewing a list of snapshots:** You can view a list of snapshots for a volume with the volume snapshot list command or with the control system in the HPE Ezmeral Data Fabric.

**Removing a snapshot:** Each snapshot has an expiration date and time, when it is deleted automatically. You can remove a snapshot before its expiration, or you can preserve a snapshot to prevent it from expiring. You can remove a snapshot with the volume snapshot remove command or from the HPE Ezmeral Data Fabric's control system interface.

## HDFS-BASED DATA PLATFORMS AND SNAPSHOTS

The snapshots from HPE Ezmeral Data Fabric capture data in a precise and consistent state. HDFS and HBase snapshots, in contrast, do not provide consistency and lack many other important capabilities. This means that HDFS and HBase snapshots cannot be trusted for the scenarios described earlier in this document.

### HDFS snapshots

HDFS is an append-only file system, so intuitively it appears as an easy-to-implement snapshot. However, the separation of data and metadata in HDFS, combined with the NameNode being a bottleneck in the system, makes it difficult or impossible to implement consistent snapshots. As a result, HDFS snapshots took years to implement and are not consistent; hence, they do not work with applications that were not specifically designed to support HDFS snapshots and their limitations.

Applications must be made snapshot-aware by calling a new HDFS API that sends up-to-date file length information to the NameNode (SuperSync/SuperFlush). It is hard to design such an application to work correctly, since the use of SuperSync across a cluster can overwhelm the NameNode, causing the entire cluster to fail or causing other processes to come to a halt. Moreover, applications making use of snapshots must avoid modifying files during the creation of the snapshot.

HDFS snapshots apply only to the metadata (on the NameNode), so they do not work correctly while files are being written. This happens because the NameNode has difficulty handling even 1000 metadata updates per second. To avoid this, HDFS avoids sending the file length to the NameNode on every hsync/hflush, because that would overwhelm the NameNode. The effect of this implementation is that files that are being written continue to change inside the snapshot, meaning the snapshot is not actually a snapshot—it can contain data that was written after the snapshot was taken, or can fail to contain data that was written and flushed before the snapshot was taken.

### HBase snapshots

Because of the storage semantics, HBase snapshots cannot rely on the underlying HDFS snapshots and need to be built separately. This is unlike the default snapshots in the HPE Ezmeral Data Fabric, where there is a common snapshot capability that applies to all data in the cluster. However, HBase snapshots are also not consistent, as each region server snapshots its own data at different times. HBase snapshots exhibit the same problems as HDFS snapshots in terms of containing transactions committed after the snapshot and missing transactions committed before the snapshot.

The following table compares all the snapshots:

| | Snapshots with HPE Ezmeral Data Fabric | HDFS snapshots | HBase snapshots |
|---|---|---|---|
| **Correctness and completeness** | | | |
| Consistency | **Yes** | No | No |
| Supported applications | **All** | Custom (snapshot-aware, SuperSync API should be called) | HBase only; correct results require application cooperation |
| Data loss issues | **No** | Recent updates before the snapshot are lost | Snapshots lose data when regions are merged |
| Enabled by default | **Yes** | No | No |
| Files, streams, and tables in one snapshot | **Yes** | No | No |
| Multiple tables in one snapshot | **Yes** | N/A | No |
| Access data in snapshot directly without recovery process | **Yes** | Yes | No (need to clone table) |
| Can be used with non-Hadoop applications | **Yes** | No | No |
| **Scalability and performance implications** | | | |
| Memory overhead | **O(1)** | O(M) M = Modifications | O(1) |
| NameNode pressure | **No** | Yes (NameNode eventually dies) | Yes (NameNode eventually dies) |
| **Management** | | | |
| Scheduling policies | **Yes** | No | No |
| Retention policies | **Yes** | No | No |
| Volumes | **Yes** | No | No |

## SUMMARY

Enterprise data storage solutions have offered the consistent snapshot capability for years, and HPE Ezmeral Data Fabric offers the same for Big Data. The snapshots as are prioritized on the platform, enabling any kind of application to benefit from it, out of the box and with no special application modifications.

## LEARN MORE AT

hpe.com/info/data-fabric



a50001680ENW, June 2020, Rev. 1