Hortonworks

HBase Backup and Restore

Vladimir Rodionov
Ted Yu

About the authors

- **Ted Yu:**
- **Been working on HBase for over 6 years**
- **HBase committer / PMC**
- **Senior Staff Engineer at Hortonworks**

- **Vladimir:**
- **active contributor to hbase (over 100 HBase JIRAs)**
- **completed most of the backup work based on IBM's initial contribution**
- **Senior Staff Engineer at Hortonworks**

HBase Backup – Why We Need It

- **Database needs disaster recovery tool**
- **Previously users can perform snapshot**
- **However, execution cost for snapshot may be high – flush across region servers is involved**
- **There was no incremental snapshot – whole dataset is captured by snapshot**
- **Incremental backup doesn't involve flushing, making continuous backup possible**

Brief History of Backup / Restore work

- **Started by engineers at IBM – see HBASE-7912**
- **Initial design included backup manifest**
- **Vladimir / Ted picked up the work last year**
- **Vladimir rendered many iterations of patches for phase 2 work (see HBASE-14123)**
- **Due to feedback from community, the design has gone thru major changes**
- **Mostly tested by developers and QA engineers so far**

HBase Backup Types

- **Full backup – foundation for incremental backups**
- **Incremental backup – can be periodic to capture changes over time**
- **Supports table level backup**

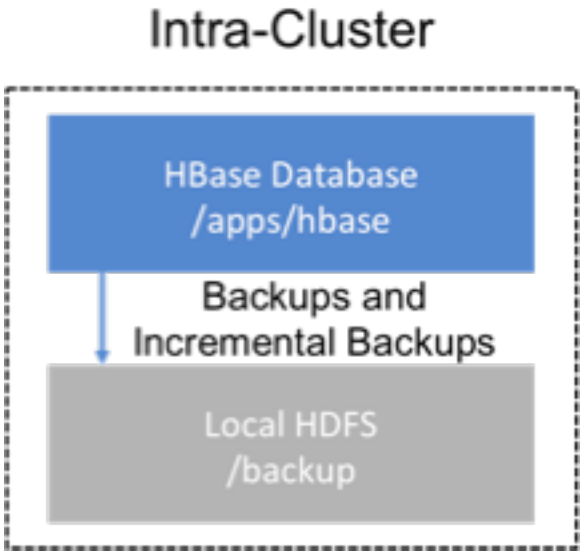
Required Configuration

- **Set hbase.backup.enable to true**
- **BackupLogCleaner for hbase.master.logcleaner.plugins**
- **LogRollMasterProcedureManager for hbase.procedure.master.classes**
- **LogRollRegionServerProcedureManager for hbase.procedure.regionserver.classes**

- **Backup may get stuck if not configured properly**

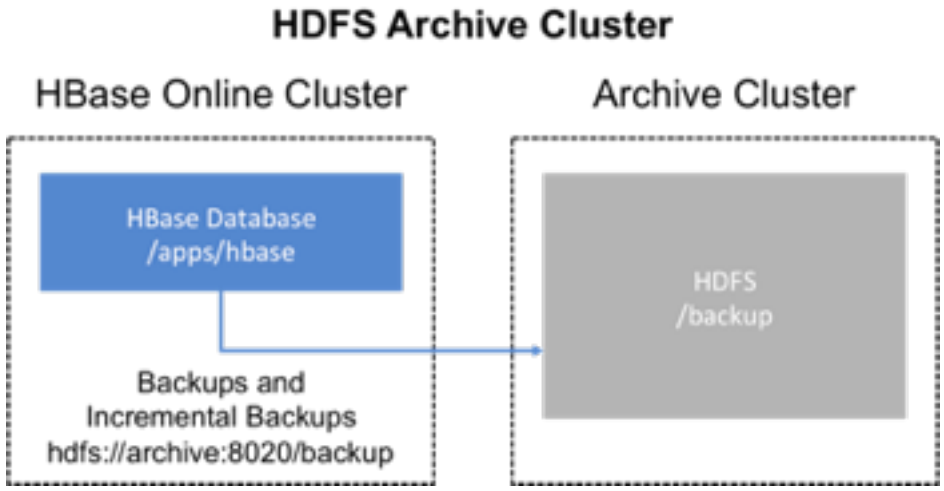
Backup Strategy

- **Intra-cluster backup is appropriate for testing**



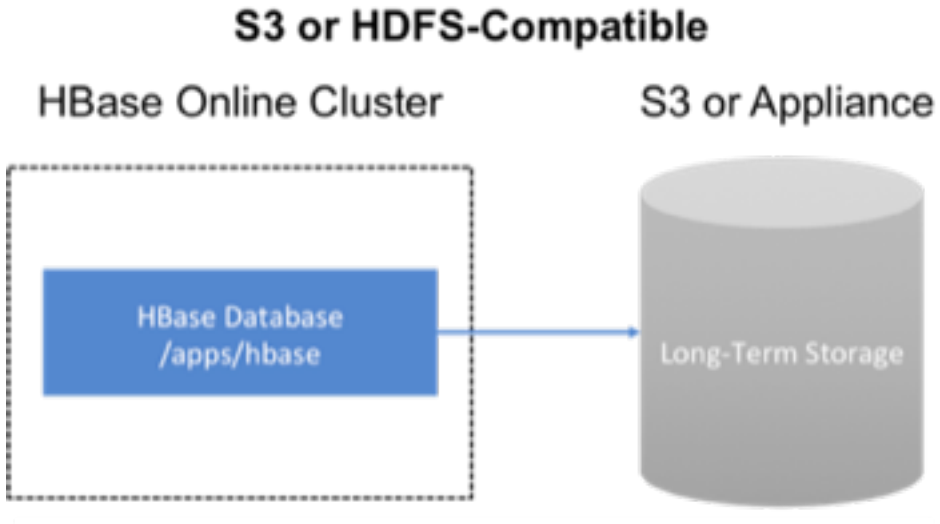
Backup Strategy: Dedicated HDFS Cluster

- **backup on a separate HDFS archive cluster**



Backup Strategy: Cloud or a Storage Vendor

- **vendor can be a public cloud provider or a storage vendor who uses a Hadoop compatible file system**



Best Practices for Backup-and-Restore

- **Secure a full backup image first**
- **Formulate a restore strategy and test it**
- **Define and use backup sets for groups of tables that are logical subsets of the entire dataset**
- **Document the backup-and-restore strategy, and ideally log information about each backup**

Creating/Maintaining Backup Image

- **Run the following command as hbase superuser:**

- **hbase backup create {{ full | incremental } {backup_root_path} {[-t tables] | [-set backup_set_name]}} [[-silent] | [-w number_of_workers] | [-b bandwidth_per_worker]]**

Using Backup Sets

- **Reduces the amount of repetitive input of table names.**
- **“hbase backup set add” command.**
- **You can have multiple backup sets**
- **Backup set can be used in the “hbase backup create” or “hbase backup restore” commands**

Restoring a Backup Image

- **You can only restore on a live HBase cluster**
- **Run the following command as hbase superuser**
- **hbase restore {[-set backup_set_name] | [backup_root_path] | [backupId] | [tables]} [[table_mapping] | [-overwrite] | [-check]]**

- **hbase restore /tmp/backup_incremental backupId_1467823988425 -t mytable1,mytable2 -overwrite**

Backup table

- **Backup table will keep track of all backup sessions**
  - Write/Read backup session state
  - Write/Read backup session progress (per region server).
  - Stores last backed up WAL file timestamp (per region server).
  - Stores list of all backed up WAL files (for BackupLogCleaner)
  - Stores backup sets
- **Must be backed up and restored separately from other tables**
- **Information needed for restore is on hdfs**

Incremental backups

- **Use Write Ahead Logs (WALs) to capture the data changes since the previous backup**
- **Log roll is executed across all RegionServers**
- **All the WAL files from incremental backups between the last full backup and the incremental backup are converted to HFiles**
- **A process similar to the DistCp tool is used to move the source backup files to the target file system**

Filter WALs on backup to only include relevant edits

- **Suppose incremental backup request is for table t, all the tables already registered in a backup system, T, are union'ed with t**
- **For every table K in the union:**

1. **Convert new WAL files into HFile applying table filter for K**
2. **Move these HFile(s) to backup destination**

Restore

- **The full backup is restored from the full backup image.**
- **HFileSplitter job will collect all HFile(s), split them into new region boundaries**
- **HBase Bulk Load utility is invoked by restore to import the HFiles as restored data in the table.**

Backup Manifest

- **Backup image has the following:**
- Backup Id, Backup Type, Backup Rootdir, Table List, start timestamp, completion timestamp
- Mapping between region server and last recorded WAL timestamp
- **Backup image keeps lineage of all previously created backup images (ancestors)**
- **When backup image list covers the image being considered, it is removed from restore**
- **See message BackupImage in Backup.proto**

Bulk load support

- **Bulk loaded Hfiles are recorded in backup table at the end of bulk load, thru preCommitStoreFile() hook**
- **During incremental backup, these Hfiles are copied to backup destination**
- **During restore, these Hfiles are loaded into target table**

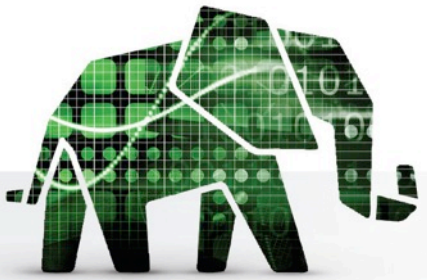
Limitations of the Backup-Restore

- **Only one active backup session is supported.**
- **Both backup and restore can't be canceled while in progress. (HBASE-15997,15998)**
- **Single backup destination only is supported. HBASE-15476**
- **There is no merge for incremental images (HBASE-14135)**
- **Only superuser (hbase) is allowed to perform backup/restore**

Credit

- **Richard Ding**
- **Vladimir Rodionov**

Q/A



Thank you.