Hadoop Upgrade

Manual upgrade procedure for Hadoop clusters.

Upgrade is an important part of the lifecycle of any software system, especially a distributed multi-component system like Hadoop. This is a step-by-step procedure a Hadoop cluster administrator should follow in order to safely transition the cluster to a newer software version. This is a general procedure, for particular version specific instructions please additionally refer to the release notes and version change descriptions.

The purpose of the procedure is to minimize damage to the data stored in Hadoop during upgrades, which could be a result of the following three types of errors:

1. **Hardware failure** is considered normal for the operation of the system, and should be handled by the software. 2. **Software errors**, and 3. **Human mistakes**

can lead to partial or complete data loss.

In our experience the worst damage to the system is incurred when as a result of a software or human mistake the name node decides that some blocks /files are redundant and issues a command for data nodes to remove the blocks. Although a lot has been done to prevent this behavior the scenario is still possible.

Common assumptions:

- Newer versions should provide automatic support and conversion of the older versions data structures.
- *Downgrades are not supported. In some cases, when e.g. data structure layouts are not affected by particular version the downgrade may be possible. In general, Hadoop does not provide tools to convert data from newer versions to older ones.*
- Different Hadoop components should be upgraded simultaneously.
- *Inter-version compatibility is not supported. In some cases when e.g. communication protocols remain unchanged different versions of different components may be compatible. For example, Job{{`Tracker v.0.4.0 can communicate with Name}}`Node v.0.3.2. In general, Hadoop does not guarantee compatibility of components of different versions.*

Instructions:

1. Stop map-reduce cluster(s)
   ```
   bin/stop-mapred.sh
   ```
   and all client applications running on the DFS cluster. 2. Run `fsck` command:
   ```
   bin/hadoop fsck / -files -blocks -locations > dfs-v-old-fsck-1.log
   ```
   Fix DFS to the point there are no errors. The resulting file will contain complete block map of the file system.

   Note. Redirecting the `fsck` output is recommend for large clusters in order to avoid time consuming output to stdout. 3. Run `lsr` command:
   ```
   bin/hadoop dfs -lsr / > dfs-v-old-lsr-1.log
   ```
   The resulting file will contain complete namespace of the file system. 4. Run `report` command to create a list of data nodes participating in the cluster.

   `bin/hadoop dfsadmin -report > dfs-v-old-report-1.log` 5. Optionally, copy all or unrecoverable only data stored in DFS to a local file system or a backup instance of DFS. 6. Optionally, stop and restart DFS cluster, in order to create an up-to-date namespace checkpoint of the old version.
   ```
   bin/stop-dfs.sh
   ```
   `bin/start-dfs.sh` 7. Optionally, repeat 3, 4, 5, and compare the results with the previous run to ensure the state of the file system remained unchanged. 8. Copy the following checkpoint files into a backup directory:
   ```
   dfs.name.dir/edits
   ```
   `dfs.name.dir/image/fsimage` 9. Stop DFS cluster.
   ```
   bin/stop-dfs.sh
   ```
   Verify that DFS has really stopped, and there are no DataNode processes running on any nodes. 10. Install new version of Hadoop software. See GettingStartedWithHadoop and HowToConfigure for details. 11. Optionally, update the `conf/slaves` file before starting, to reflect the current set of active nodes. 12. Optionally, change the configuration of the name node's and the job tracker's port numbers, to ignore unreachable nodes that are running the old version, preventing them from connecting and disrupting system operation.
   ```
   fs.default.name
   ```
   `mapred.job.tracker` 13. Optionally, start name node only.
   ```
   bin/hadoop-daemon.sh start namenode -upgrade
   ```
   This should convert the checkpoint to the new version format. 14. Optionally, run `lsr` command:
   ```
   bin/hadoop dfs -lsr / > dfs-v-new-lsr-0.log
   ```
   and compare with `dfs-v-old-lsr-1.log` 15. Start DFS cluster.

   `bin/start-dfs.sh` 16. Run report command:
   ```
   bin/hadoop dfsadmin -report > dfs-v-new-report-1.log
   ```
   and compare with `dfs-v-old-report-1.log` to ensure all data nodes previously belonging to the cluster are up and running. 17. Run `lsr` command:
   ```
   bin/hadoop dfs -lsr / > dfs-v-new-lsr-1.log
   ```
   and compare with `dfs-v-old-lsr-1.log`. These files should be identical unless the format of `lsr` reporting or the data structures have changed in the new version. 18. Run `fsck` command:
   ```
   bin/hadoop fsck / -files -blocks -locations > dfs-v-new-fsck-1.log
   ```
   and compare with `dfs-v-old-fsck-1.log`. These files should be identical, unless the `fsck` reporting format has changed in the new version.

   19. Start map-reduce cluster
   ```
   bin/start-mapred.sh
   ```

In case of failure the administrator should have the checkpoint files in order to be able to repeat the procedure from the appropriate point or to restart the old version of Hadoop. The `*.log` files should help in investigating what went wrong during the upgrade.

Enhancements:

This is a list of enhancements intended to simplify the upgrade procedure and to make the upgrade safer in general.

1. A **shutdown** function is required for Hadoop that would cleanly shut down the cluster, merging edits into the image, avoiding the restart-DFS phase. 2. The **safe mode** implementation will further help to prevent name node from voluntary decisions on block deletion and replication. 3. A **fa ster fsck** is required. *Currently `fsck` processes 1-2 TB per minute.* 4. Hadoop should provide a **backup solution** as a stand alone application. 5. Introduce an explicit **-upgrade option** for DFS (See below) and a related 6. **finalize upgrade** command.

Shutdown command:

During the shutdown the name node performs the following actions.

- It locks the namespace for further modifications and waits for active leases to expire, and pending block replications and deletions to complete.
- Runs `fsck`, and optionally saves the result in a file provided.
- Checkpoints and replicates the namespace image.
- Sends shutdown command to all data nodes and verifies they actually turned themselves off by waiting for as long as 5 heartbeat intervals during which no heartbeats should be reported.
- Stops all running threads and terminates itself.

Upgrade option for DFS:

The main idea of upgrade is that each version that modifies data structures on disk has its own distinct working directory. For instance, we'd have a "v0.6" and a “v0.7” directory for the name node and for all data nodes. These version directories will be automatically created when a particular file system version is brought up for the first time. If DFS is started with the -upgrade option the new file system version will do the following:

- The name node will start in the read-only mode and will read in the old version checkpoint converting it to the new format.
- Create a new working directory corresponding to the new version and save the new image into it. The old checkpoint will remain untouched in the working directory corresponding to the old version.
- The name node will pass the upgrade request to the data nodes.
- Each data node will create a working directory corresponding to the new version. If there is metadata in side files it will be re-generated in the new working directory.
- Then the data node will hard link blocks from the old working directory to the new one. The existing blocks will remain untouched in their old directories.
- The data node will confirm the upgrade and send its new block report to the name node.
- Once the name node received the upgrade confirmations from all data nodes it will run the `fsck` and then switch to the normal mode when it’s ready to serve clients’ requests.

This ensures that a snapshot of the old data is preserved until the new version is validated and tested to function properly. Following the upgrade the file system can be run for a week or so to gain confidence. It can be rolled back to the old snapshot if it breaks, or the upgrade can be “finalized” by admin using the “finalize upgrade” command, which would remove old version working directories.

Care must be taken to deal with data nodes that are missing during the upgrade stage. In order to deal with such nodes the name node should store the list of data nodes that have completed the upgrade, and reject data nodes that did not confirm the upgrade.

When DFS will allow modification of blocks, this will require copying blocks into the current version working directory before modifying them.

Linking allows the data from several versions of Hadoop to coexist and even evolve on the same hardware without duplicating common parts.

Finalize Upgrade:

When the Hadoop administrator is convinced that the new version works properly he/she/it can issue a “finalize upgrade” request.

- The finalize request is first passed to the data nodes so that they could remove their previous version working directories with all block files. This does not necessarily lead to physical removal of the blocks as long as they still are referenced from the new version.
- When the name node receives confirmation from all data nodes that current upgrade is finalized it will remove its own old version directory and the checkpoint in it thus completing the upgrade and making it permanent.

The finalize upgrade procedure can run in the background without disrupting the cluster performance. Being in finalize mode the name node will periodically verify confirmations from the data nodes and finalize itself when the load is light.

Simplified Upgrade Procedure:

The new utilities will substantially simplify the upgrade procedure:

1. Stop map-reduce cluster(s) and all client applications running on the DFS cluster. 2. Stop DFS using the shutdown command. 3. Install new version of Hadoop software. 4. Start DFS cluster with -upgrade option. 5. Start map-reduce cluster. 6. Verify the components run properly and finalize the upgrade when convinced. This is done using the -finalizeUpgrade option to the hadoop dfsadmin command.

Upgrade Guide for Hadoop-0.14

The procedure described here applies any Hadoop upgrade. In addition to this page, please read Hadoop-0.14 Upgrade when upgrading to Hadoop-0.14 (Check release notes). Hadoop-0.14 Upgrade describes how to follow the progress of upgrade that is specific to Hadoop-0.14.

Links

- Paul 06 Jul, 19:02
- Eric 06 Jul, 19:24
- Paul 07 Jul, 15:51