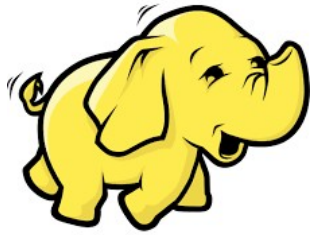
HDFS Operations

Tushar B. Kute,
http://tusharkute.com



Starting HDFS

- Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command.
  - `$ hadoop namenode -format`
- After formatting the HDFS, start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.
  - `$ start-dfs.sh`

Listing files in HDFS

- After loading the information in the server, we can find the list of files in a directory, status of a file, using ‘ls’.
- Given below is the syntax of ls that you can pass to a directory or a filename as an argument.
  - `$ hadoop fs -ls <args>`

Inserting Data into HDFS

- Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system.
- Follow the steps given below to insert the required file in the Hadoop file system.
  - Step 1

    You have to create an input directory.

    ```
    $ hadoop fs -mkdir /user/input
    ```

Inserting Data into HDFS

- Step 2
  - Transfer and store a data file from local systems to the Hadoop file system using the put command.
  - $ hadoop fs -put /home/file.txt /user/input
- Step 3
  - You can verify the file using ls command.
  - $ hadoop fs -ls /user/input

Shutting Down the HDFS

- You can shut down the HDFS by using the following command.
  - `$ stop-dfs.sh`

HDFS Operations

- `$ hadoop fs -help commandName`
- will display a short usage summary for the operation in question, if you are stuck.
- A table of all the operations is shown below. The following conventions are used for parameters:
  - "<path>" means any file or directory name.
  - "<path>..." means one or more file or directory names.
  - "<file>" means any filename.
  - "<src>" and "<dest>" are path names in a directed operation.
  - "<localSrc>" and "<localDest>" are paths as above, but on the local file system.

Commands

- `$ hadoop fs -ls <path>`
- Lists the contents of the directory specified by path, showing the names, permissions, owner, size and modification date for each entry.

Commands

- `$ hadoop fs -lsr <path>`
- Behaves like -ls, but recursively displays entries in all subdirectories of path.

Commands

- `$ hadoop fs -du <path>`
- Shows disk usage, in bytes, for all the files which match path; filenames are reported with the full HDFS protocol prefix.

Commands

- `$ hadoop fs -dus <path>`
- Like -du, but prints a summary of disk usage of all files/directories in the path.

Commands

- `$ hadoop fs -mv <src> <dest>`
- Moves the file or directory indicated by src to dest, within HDFS.

Commands

- `$ hadoop fs -cp <src> <dest>`
- Copies the file or directory indicated by src to dest, within HDFS.

Commands

- `$ hadoop fs -rm <path>`
- Removes the file or empty directory identified by <path>.

Commands

- `$ hadoop fs -rmr <path>`
- Removes the file or directory identified by path. Recursively deletes any child entries (i.e., files or subdirectories of path).

Commands

- `$ hadoop fs -put <localSrc> <dest>`
- Copies the file or directory from the local file system identified by localSrc to dest within the DFS.
- Following also similar command-
- `$ hadoop fs -copyFromLocal <localSrc> <dest>`

Commands

- `$ hadoop fs -moveFromLocal <localSrc> <dest>`
- Copies the file or directory from the local file system identified by localSrc to dest within HDFS, and then deletes the local copy on success.

Commands

- `$ hadoop fs -get <src> <localDest>`
- Copies the file or directory in HDFS identified by src to the local file system path identified by localDest.
- Following also similar command-
- `$ hadoop fs -copyToLocal <dest> <localSrc>`

Commands

- `$ hadoop fs -cat <filename>`
- Displays the contents of the file on STDOUT (screen).

Commands

- `$ hadoop fs -setrep [-R] [-w] rep <path>`

  Sets the target replication factor for files identified by path to rep. (The actual replication factor

Commands

- `$ hadoop fs -touchz <path>`

  Creates a file at path containing the current time as a timestamp. Fails if a file already exists at path, unless the file is already size 0.

Commands

- `$ hadoop fs -stat [format] <path>`

  Prints information about path. Format is a string which accepts file size in blocks

  (%b), filename (%n), block size (%o),

  replication (%r), and modification date (%y, %Y).

Commands

- `$ hadoop fs -tail [-f] <file2name>`

  Shows the last 1KB of file on stdout.

Commands

- `$ hadoop fs -chmod [-R] mode, mode, … <path>`

  Changes the file permissions associated with one or more objects identified by path.... Performs changes recursively with -R. mode is a 3-digit octal mode, or {augo}+/-{rwxX}. Assumes if no scope is specified and does not apply an umask.

Commands

- `$ hadoop fs -chown [-R][owner] [:[group]] <path>`

  Sets the owning user and/or group for files or directories identified by path.... Sets owner recursively if -R is specified.

Commands

- `$ hadoop fs -chgrp [-R] group <path>`

  Sets the owning group for files or directories identified by path.... Sets group recursively if -R is specified.

Commands

- `$ hadoop fs -help <cmd-name>`

  Returns usage information for one of the commands listed above. You must omit the leading '-' character in cmd.

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com