Hadoop Archives Guide

Table of contents

1. Overview

Hadoop archives are special format archives. A Hadoop archive maps to a file system directory. A Hadoop archive always has a *.har extension. A Hadoop archive directory contains metadata (in the form of _index and _masterindex) and data (part-*) files. The _index file contains the name of the files that are part of the archive and the location within the part files.

2. How to Create an Archive

```
Usage: hadoop archive -archiveName name -p <parent> [-r
<replication factor>] <src>* <dest>
```

-archiveName is the name of the archive you would like to create. An example would be foo.har. The name should have a *.har extension. The parent argument is to specify the relative path to which the files should be archived to. Example would be :

```
-p /foo/bar a/b/c e/f/g
```

Here /foo/bar is the parent path and a/b/c, e/f/g are relative paths to parent. Note that this is a Map/Reduce job that creates the archives. You would need a map reduce cluster to run this. For a detailed example the later sections.

-r indicates the desired replication factor; if this optional argument is not specified, a replication factor of 3 will be used.

If you just want to archive a single directory /foo/bar then you can just use

```
hadoop archive -archiveName zoo.har -p /foo/bar -r 3
/outputdir
```

3. How to Look Up Files in Archives

The archive exposes itself as a file system layer. So all the fs shell commands in the archives work but with a different URI. Also, note that archives are immutable. So, rename's, deletes and creates return an error. URI for Hadoop Archives is

```
har://scheme-hostname:port/archivepath/fileinarchive
```

If no scheme is provided it assumes the underlying filesystem. In that case the URI would look like

```
har:///archivepath/fileinarchive
```

4. Archives Examples

4.1. Creating an Archive

```
hadoop archive -archiveName foo.har -p /user/hadoop -r 3 dir1
dir2 /user/zoo
```

The above example is creating an archive using /user/hadoop as the relative archive directory. The directories /user/hadoop/dir1 and /user/hadoop/dir2 will be archived in the following file system directory -- /user/zoo/foo.har. Archiving does not delete the input files. If you want to delete the input files after creating the archives (to reduce namespace), you will have to do it on your own. In this example, because `-r 3` is specified, a replication factor of 3 will be used.

4.2. Looking Up Files

Looking up files in hadoop archives is as easy as doing an ls on the filesystem. After you have archived the directories /user/hadoop/dir1 and /user/hadoop/dir2 as in the example above, to see all the files in the archives you can just run:

```
hadoop dfs -lsr har:///user/zoo/foo.har/
```

To understand the significance of the -p argument, lets go through the above example again. If you just do an ls (not lsr) on the hadoop archive using

```
hadoop dfs -ls har:///user/zoo/foo.har
```

The output should be:

```
har:///user/zoo/foo.har/dir1
har:///user/zoo/foo.har/dir2
```

As you can recall the archives were created with the following command

```
hadoop archive -archiveName foo.har -p /user/hadoop dir1 dir2
/user/zoo
```

If we were to change the command to:

```
hadoop archive -archiveName foo.har -p /user/ hadoop/dir1
hadoop/dir2 /user/zoo
```

then a ls on the hadoop archive using

```
hadoop dfs -ls har:///user/zoo/foo.har
```

would give you

```
har:///user/zoo/foo.har/hadoop/dir1
har:///user/zoo/foo.har/hadoop/dir2
```

Notice that the archived files have been archived relative to /user/ rather than /user/hadoop.

5. Hadoop Archives and MapReduce

Using Hadoop Archives in MapReduce is as easy as specifying a different input filesystem than the default file system. If you have a hadoop archive stored in HDFS in /user/zoo/foo.har then for using this archive for MapReduce input, all you need to specify the input directory as har:///user/zoo/foo.har. Since Hadoop Archives is exposed as a file system MapReduce will be able to use all the logical input files in Hadoop Archives as input.