CouchSqoop: A Couchbase plugin for Sqoop

Documentation

Couchbase
Cloudera

Table of Contents

1 Introduction:

If you are reading this then you have just downloaded the Couchbase Sqoop plugin. This plugin allows you to connect to a Couchbase or Membase server and stream keys into HDFS or Hive for processing with Hadoop. Note that in this document we will refer to our database as Couchbase, but if you are using Membase everything will still. If you have used Sqoop before for doing imports and exports from other databases then using this plugin should be straightforward since it uses similar command line argument structure.

2 Installation:

The installation process for the Couchbase Sqoop plugin is simple. When you download the plugin from Cloudera you should find a set of files that need to be moved into you Sqoop installation. These files along with a short description of why they are needed are listed below.

- couchsqoop-plugin-1.0.jar – This is the jar file that contains all of the source code that makes Sqoop read data from Couchbase.
- couchsqoop-config.xml – This is a property file used to register a ManagerFactory for the Couchbase plugin with Sqoop.
- couchsqoop-manager.xml – This property file tells Sqoop what jar the ManagerFactory defined in couchsqoop-config.xml resides.
- memcached-2.7.jar – This is the client application used by our plugin to read and write data from Couchbase.
- commons-codec-1.4.jar – This is a dependency of memcached-2.7.jar.
- jettison-1.1.jar - This is a dependency of memcached-2.7.jar.
- netty-3.1.5GA.jar - This is a dependency of memcached-2.7.jar.
- install.sh – A script to automatically install the Couchbase plugin files to Sqoop.

2.1 Automatic Installation:

Automatic installation is done through the use of the install.sh script that comes with the plugin download. The script takes one argument, the path to your Sqoop installation. Below is an example of how to use the script.

- ./install.sh path_to_sqoop_home

2.2 Manual Installation:

Manual installation of the Couchbase plugin requires copying the files downloaded from Cloudera into your Sqoop installation. Below are a list of files that contained in the plugin and the name of the directory in your Sqoop installation to copy each file to.

- couchsqoop-plugin-1.0.jar – lib
- memcached-2.7.jar – lib
- commons-codec-1.4.jar – lib
- jettison-1.1.jar - lib
- netty-3.1.5GA.jar – lib
- couchsqoop-config.xml – conf
- couchsqoop-manager.xml – conf/managers.d

3 Using Sqoop:

Sqoop can be used with a variety of tools that are included with Sqoop. In this section we discuss usage of each tool.

3.1 Tables:

Since Sqoop is built for a relational model it requires that the user specifies a table to import and export into Couchbase. The Couchbase plugin uses the --table option to specify the type of tap stream for importing and exporting into Couchbase. For exports the user must enter a value for the --table option even though what is entered will not actually be used by the plugin. For imports the table command can take on only two values.

- DUMP - Causes all keys currently in Couchbase to be read into HDFS.
- BACKFILL_<time in minutes> - Streams all key mutations for a given amount of time (in minutes).

Note that for the --table value BACKFILL that a time should be put in place of the brackets. For example BACKFILL_5 means stream key mutations in the Couchbase server for 5 minutes and then stop the stream.

For exports a value for --table is required, but the value will not be used. Any value used for the --table option when doing export will be ignored by the Couchbase plugin.

3.2 Connect String:

A connect string option is required in order to connect to Couchbase. This can be specified with --connect on the command line. Below are two examples of connect strings.

- http://10.2.1.55:8091/pools
- http://10.2.1.55:8091/pools,http://10.2.1.56:8091/pools

When creating your connect strings simply replace the IP address above with the IP address of your Couchbase sever. If you have multiple servers you can list them in a comma-separated list.

Why list multiple servers? Let's say you create a backfill stream for 10,080 minutes or one week. In that time period you might have a server crash, have to add another server, or remove a server from your cluster. Providing an address to each server allows the import or export to proceed through topology changes to your cluster. In the first example above if you had a two-node cluster and 10.2.1.55 goes down then the import will fail even though the entire cluster didn't go down. If you list both machines then the import will continue unaffected by the downed server and your import will complete successfully.

3.3 Connecting to Different Buckets:

By default the Couchbase plugin connects to the default bucket. If you want to connect to a bucket other than the default bucket you can specify the bucket name with the --username option. If you have to connect to a sasl bucket use the --password option followed by the buckets password.

3.4 Importing:

Importing data to your cluster requires the use of the Sqoop import command followed by the parameters --connect and --table. Below are some example imports.

- bin/sqoop import --connect http://10.2.1.55:8091/pools --table DUMP
  This will dump all key-value pairs from Couchbase into HDFS.
- bin/sqoop import --connect http://10.2.1.55:8091/pools --table BACKFILL_10
  This will stream all key-value mutations from Couchbase into HDFS.

Note that Sqoop provides many more options to the import command than we will cover in this document. Run "bin/sqoop import help" for a list of all options and see the Sqoop documentation for more details about these options.

3.5 Exporting:

Exporting data to your cluster requires the use of the Sqoop import command followed by the parameters --connect, --export-dir, and --table. Below are some example imports.

- bin/sqoop export --connect http://10.2.1.55:8091/pools --table garbage_value --export-dir dump_4-12-11
  This will export all key-value pairs from the HDFS directory specified by export-dir into Couchbase.
- bin/sqoop export –connect http://10.2.1.55:8091/pools --table garbage_value --export-dir backfill_4-29-11
  This will export all key-value pairs from the HDFS directory specified by export-dir into Couchbase.

Note that Sqoop provides many more options to the export command than we will cover in this document. Run "bin/sqoop export help" for a list of all options and see the Sqoop documentation for more details about these options.

3.6 List table:

Sqoop has a tool called list tables that in a relational database has a lot of meaning since it shows us what kinds of things we can import. As noted in previous sections Couchbase doesn't have a notion of tables, but we use DUMP and BACKFILL_<time in minutes> as values to the --table option. As a result using the list-tables tool does the following.

- bin/sqoop list-tables --connect http://10.2.1.55:8091/pools
  ```
  DUMP
  BACKFILL_5
  ```

All this does in the case of the Couchbase plugin is remind us what we can use as an argument to the --table option. We give BACKFILL a time of 5 minutes so that the import-all-tables tool functions properly.

Note that Sqoop provides many more options to the list-tables command than we will cover in this document. Run "bin/sqoop list-tables help" for a list of all options and see the Sqoop documentation for more details about these options.

3.7 Import All Tables

In the Couchbase plugin the import-all-tables tool dumps all keys in Couchbase into HDFS and then streams all key-value mutations into HDFS for five minutes. This command is a direct result of running import on each database from the list-tables command. Below is an example of this command.

- bin/sqoop import-all-tables –connect http://10.2.1.55:8091/pools

Note that Sqoop provides many more options to the import-all-tables command than we will cover in this document. Run "bin/sqoop import-all-tables help" for a list of all options and see the Sqoop documentation for more details about these options.

3.8 Limitations:

While Couchbase provides many great features to import and export data from Couchbase to HDFS there is some functionality that the plugin doesn't implement in Sqoop. Here's a list of what isn't implemented.

- Querying: You cannot run queries on Couchbase. All tools that attempt to do this will fail with a NotSupportException.
- list-databases tool: Even though Couchbase is a multi-tenant system that allows for multiple databases. There is no way of listing these databases from Sqoop.
- eval-sql tool: Couchbase doesn't use SQL so this tool will definitely not work.

4 Internals:

The Couchbase plugin consists of two parts. The first part is the addition of code that allows the mappers in Hadoop to read the values sent to it from Couchbase. The second part is the use of the Spymemcached client to get data to and from Couchbase. For imports the plugin uses a new tap stream feature in Spymemcached. Tap streams allow users to stream large volumes of data from Couchbase into other applications and are also at the heart of replication in Couchbase. They enable a fast way to move data from Couchbase to HDFS for processing with Hadoop. Getting data back into Couchbase runs through the front end of Couchbase using the memcached protocol. This way of accessing Couchbase is probably more familiar to Couchbase users than the tap interface.

For more information about the internals of Sqoop see the Sqoop documentation.