platform
By Per Scholas

Sqoop Fundamentals

Sqoop is a tool designed to transfer data between Hadoop and relational database servers.

Week 10 - Day 5

Overview

- What is Sqoop?
- How Sqoop works?
- Sqoop Import
- Sqoop Export
- Sqoop Incremental Import
- Sqoop Job

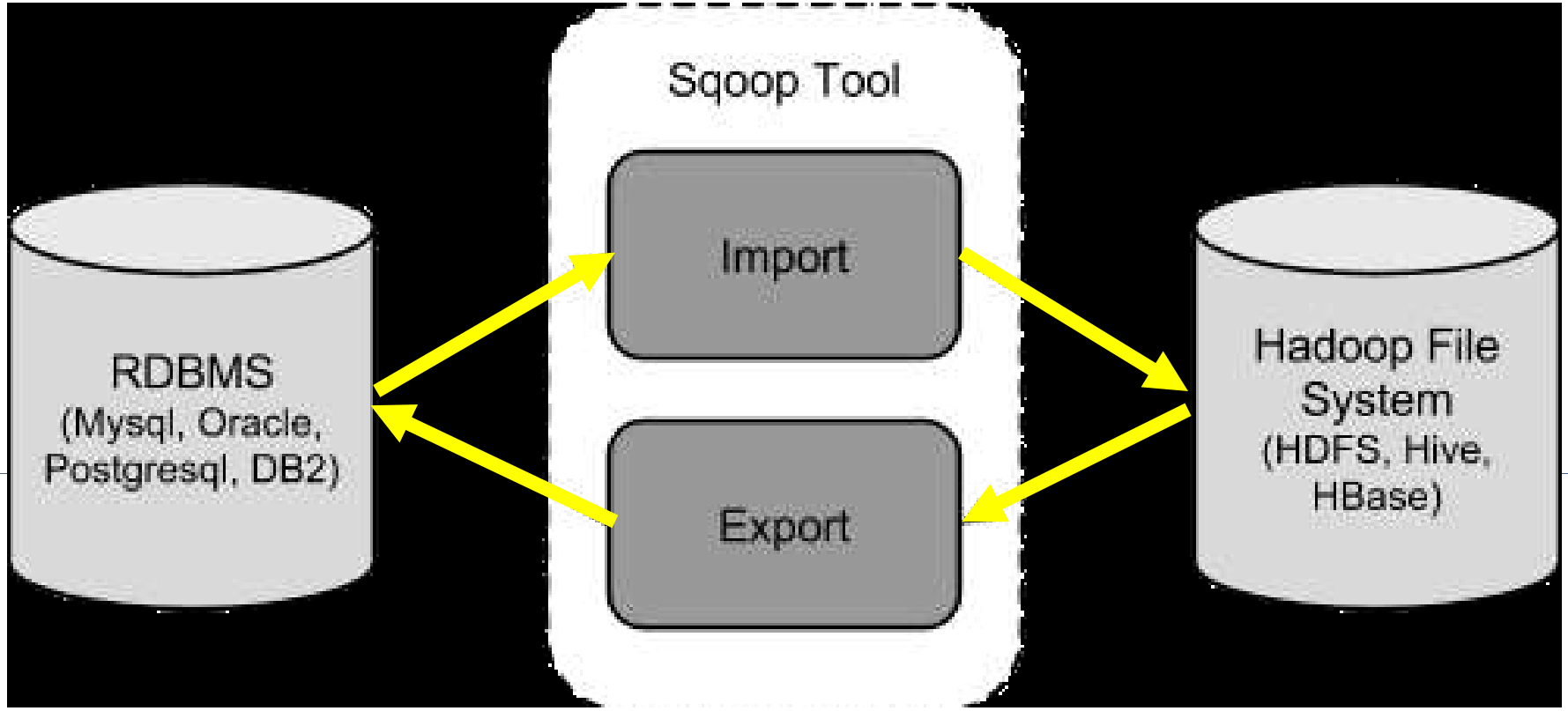
What is Sqoop?

- The traditional application management system, that is, the interaction of applications with relational database using RDBMS, is one of the sources that generate Big Data. Such Big Data, generated by RDBMS, is stored in **Relational Database Servers** in the relational database structure.
- When Big Data storages and analyzers such as MapReduce, Hive, HBase, Cassandra, Pig, etc. of the Hadoop ecosystem came into picture, they required a tool to interact with the relational database servers for importing and exporting the Big Data residing in them. Here, Sqoop occupies a place in the Hadoop ecosystem to provide feasible interaction between relational database server and Hadoop's HDFS.

- **Sqoop:** "SQL to Hadoop and Hadoop to SQL"
- Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases. It is provided by the Apache Software Foundation.

How Sqoop Works?

Sqoop Workflow



Sqoop Import

- The import tool imports individual tables from RDBMS to HDFS. Each row in a table is treated as a record in HDFS. All records are stored as text data in text files or as binary data in Avro and Sequence files.

- The following syntax is used to import data into HDFS:
  - $ sqoop import (generic-args) (import-args)
  - $ sqoop-import (generic-args) (import-args)
  - *While the Hadoop generic arguments must precede any import arguments, you can type the import arguments in any order with respect to one another.

Sqoop Generic Arguments

Argument	Description
--connect <jdbc-uri>	Specify JDBC connect string
--connection-manager <class-name>	Specify connection manager class to use
--driver <class-name>	Manually specify JDBC driver class to use
--hadoop-home <dir>	Override $HADOOP_HOME
--help	Print usage instructions
-P	Read password from console
--password <password>	Set authentication password
--username <username>	Set authentication username
--verbose	Print more information while working
--connection-param-file <filename>	Optional properties file that provides connection parameters

Sqoop Import Arguments

Argument	Description
--append	Append data to an existing dataset in HDFS
--as-avrodatafile	Imports data to Avro Data Files
--as-sequencefile	Imports data to SequenceFiles
--as-textfile	Imports data as plain text (default)
--boundary-query <statement>	Boundary query to use for creating splits
--columns <col,col,col…>	Columns to import from table
--direct	Use direct import fast path
--direct-split-size <n>	Split the input stream every *n* bytes when importing in direct mode
--inline-lob-limit <n>	Set the maximum size for an inline LOB
-m,--num-mappers <n>	Use *n* map tasks to import in parallel
-e,--query <statement>	Import the results of *statement*.
--split-by <column-name>	Column of the table used to split work units
--table <table-name>	Table to read
--target-dir <dir>	HDFS destination dir
--warehouse-dir <dir>	HDFS parent for table destination
--where <where clause>	WHERE clause to use during import
-z,--compress	Enable compression
--compression-codec <c>	Use Hadoop codec (default gzip)
--null-string <null-string>	The string to be written for a null value for string columns
--null-non-string <null-string>	The string to be written for a null value for non-string columns

Sqoop Import --- Continued

- **Importing a Table**
  - Sqoop tool 'import' is used to import table data from the table to the Hadoop file system as a text file or a binary file.
- **Importing into Target Directory**
  - We can specify the target directory while importing table data into HDFS using the Sqoop import tool.
- **Import Subset of Table Data**
  - We can import a subset of a table using the 'where' clause in Sqoop import tool. It executes the corresponding SQL query in the respective database server and stores the result in a target directory in HDFS.
  - The syntax for where clause is as follows:
    - --where <condition>

Sqoop Import --- Continued

- **Incremental Import**
  - Incremental import is a technique that imports only the newly added rows in a table. It is required to add 'incremental', 'check-column', and 'last-value' options to perform the incremental import.
  - The following syntax is used for the incremental option in Sqoop import command:
    - --incremental <mode>
    - --check-column <column name>
    - --last value <last check column value>
- **Import All Tables**
  - The following syntax is used to import all tables.
    - $ sqoop import-all-tables (generic-args) (import-args)
    - $ sqoop-import-all-tables (generic-args) (import-args)

Sqoop Export

- The export tool exports a set of files from HDFS back to an RDBMS. The target table must exist in the target database. The files given as input to Sqoop contain records, which are called as rows in table. Those are read and parsed into a set of records and delimited with user-specified delimiter.
- The default operation is to insert all the record from the input files to the database table using the INSERT statement. In update mode, Sqoop generates the UPDATE statement that replaces the existing record into the database.

- The following is the syntax for the export command:
  - $ sqoop export (generic-args) (export-args)
  - $ sqoop-export (generic-args) (export-args)

Sqoop Incremental Import

- Sqoop supports two types of incremental imports: append and lastmodified. You can use the --incremental argument to specify the type of incremental import to perform.
- You should specify append mode when importing a table where new rows are continually being added with increasing row id values. You specify the column containing the row's id with --check-column. Sqoop imports rows where the check column has a value greater than the one specified with --last-value.
- An alternate table update strategy supported by Sqoop is called lastmodified mode. You should use this when rows of the source table may be updated, and each such update will set the value of a last-modified column to the current timestamp. Rows where the check column holds a timestamp more recent than the timestamp specified with --last-value are imported.
- At the end of an incremental import, the value which should be specified as --last-value for a subsequent import is printed to the screen. When running a subsequent import, you should specify --last-value in this way to ensure you import only the new or updated data. This is handled automatically by creating an incremental import as a saved job, which is the preferred mechanism for performing a recurring incremental import. See the section on saved jobs later in this document for more information.

Sqoop Incremental Import --- Continued

Table. Incremental import arguments:

Argument	Description
--check-column (col)	Specifies the column to be examined when determining which rows to import.
--incremental (mode)	Specifies how Sqoop determines which rows are new. Legal values for mode include append and lastmodified.
--last-value (value)	Specifies the maximum value of the check column from the previous import.

Sqoop Job

- Sqoop job creates and saves the import and export commands. It specifies parameters to identify and recall the saved job. This re-calling or re-executing is used in the incremental import, which can import the updated rows from RDBMS table to HDFS.

- The following is the syntax for creating a Sqoop job:
  - $ sqoop job (generic-args) (job-args) [-- [subtool-name] (subtool-args)]
  - $ sqoop-job (generic-args) (job-args) [-- [subtool-name] (subtool-args)]
- Last Value from the previous import acts as the argument for --last-value
- Sqoop Job works in the following two ways:

1. Sqoop Job checks for any insert/update in data between the last value timestamp (Lower bound value) and Current timestamp (Upper bound value) and imports the modified or newly added rows.
2. Sqoop Job checks for any data that has the value in the column specified greater than the last value and imports the newly added rows.
   Values of the column should be **strictly monotonically increasing sequence of numbers.**

Sqoop Job -- Continued

- **Sqoop metastore**
- Sqoop metastore is used to store Sqoop job information in a central place. This helps collaboration between Sqoop users and developers for example User A can create a job to load some specific data, then any other user can access from any node in the cluster the same job and just run it again. This is very convenient when using Sqoop in Oozie workflows.
- Sqoop Metastore Syntax:
  - $ sqoop metastore (generic-args) (metastore-args)
  - $ sqoop-metastore (generic-args) (metastore-args)

Summary Review

- What is Sqoop?
  - Why use Sqoop?
- How Sqoop works?
- Sqoop Import
- Sqoop Export
- Sqoop Incremental Import
  - Why use incremental import?
- Sqoop Job
  - Meta Store
  - Last Value

Test Your Knowledge

Question document is located in LMS

- The document contains questions and answers for Pig
  - "Documents/Week11/Hands_on/Sqoop_Knowledge_Check.docx"

Questions?

End of Module