Reading and Writing Parquet files with Mortar

Using Mortar to read/write Parquet files
You can load Parquet formatted files into Mortar, or tell Mortar to output data in Parquet format using the Hadoop connectors that can be built from here or downloaded from here.

Last year, Cloudera, in collaboration with Twitter and others, released a new Apache Hadoop-friendly, binary, columnar file format called Parquet. Parquet is an open source serialization format that stores data in a binary column-oriented fashion. Instead of how row-oriented data is stored, where every column for a row is stored together and then followed by the next row (again with columns stored next to each other), Parquet turns things on its head. Instead, Parquet will take a group of records and store the values of the first column together for the entire row group, followed by the values of the second column, and so on. Parquet has optimizations for scanning individual columns, so it doesn't have to read the entire row group if you are only interested in a subset of columns.

Installation:

In order to use parquet pig hadoop, the jar needs to be in Pig's classpath. There are various ways of making that happen though typically the REGISTER command is used:

the command expects a proper URI that can be found either on the local file-system or remotely. Typically it's best to use a distributed file-system (like HDFS or Amazon S3) and use that since the script might be executed on various machines.

Loading Data from a Parquet file

As you would expect, loading the data is straight forward:

Writing data to a Parquet file

You can store the output of a Pigscript in Parquet format using the jar file as well.

Compression Options
You can select the compression to use when writing data with the parquet.compression property. For example

The valid options for compression are:

UNCOMPRESSED
GZIP
SNAPPY

The default is SNAPPY

Simple examples of Mortar to Parquet and back
One of the advantages is that Parquet can compress data files. In this example, I downloaded the CMS Medicare Procedure set. This file is a CSV file that is 1.79GB in size. It contains 9,153,274 rows. I wrote a simple Pig script that loads the data, orders the data by the provider's state and then write it back out as a pipe delimited text file and as a parquet file.

With the regular pipe delimited file, the file size is still 1.79GB in size. With parquet, the file is 838.6MB (a 53% reduction in size).

Another advantage is that you can take the resulting output of the file and copy it into an impala directory (/user/hive/warehouse/mortarP) for an existing Parquet formatted table. Once you refresh the table (refresh), the data is immediately available.

Finally, you can copy the parquet file from an Impala table to your local drive or to an S3 bucket, connect to it with Mortar and work directly with the data.

If we illustrate this job, we see that we see the resulting data sets.

Using Mortar and Parquet provides additional flexibility when dealing with large data sets.