Transcript name: Working with Jaql

English
Jaql is a JSON-based query language that, like PigLatin and HiveQL, translates into Hadoop MapReduce jobs. JSON is the data interchange standard that is human-readable like XML but is designed to be lighter-weight. You run Jaql programs using the Jaql shell. You start the Jaql shell using the jaqlshell command. If you pass it no arguments, you start it in interactive mode. If you pass the -b argument and the path to a file, you will execute the contents of that file as a Jaql script. Finally, if you pass the -e argument, the Jaql shell will execute the Jaql statement that follows the -e. There are two modes that the Jaql shell can run in: The first is cluster mode, specified with a -c argument. It uses your Hadoop cluster if you have one configured. The other option is minicluster mode, which starts a minicluster that is useful for quick tests. The Jaql query language is a data-flow language. You describe flows of data by setting up sources and sinks and specifying how they connect to each other. For example, a read from, say, a local file is a source and a write to, say, HDFS is a sink. Along the way from the source to the sink, you can have the data pass through various filters, grouping functions, sorting functions, and transformation functions. These are called operators and there are nine different kinds. You can bind a source to a variable using the equals sign operator. You can then refer to that source by its variable name throughout your program. The way you connect the flows of data to consumers of that data is through the pipe operator, which is a minus sign followed by a greater than symbol. This operator expects an array as its input. For example, say we have a twitter feed stored in a file in HDFS and we turn it into a source and bind that source to the variable $tweets. We can supply $tweets as the array input to the pipe operator and use a filter operator to filter out tweets that did not come from tweetdeck. We can refer to each element in the array with the $ implicit variable and then access its individual members such as from_src using a dot after the $. The Jaql language has numerous built-in functions. There are 18 categories of such functions with many functions in each category. Jaql supports many different data stores including obvious ones like HDFS and the local filesystem but also DB2 and JDBC. The data formats are not limited to JSON. You can also interact with data in CSV (or comma separated values) or XML formats. Now let us take a look at Jaql in action. We start Jaql's shell by running the jaqlshell command from the jaql/bin directory under biginsights. We give it the -c option to tell it to run on the hadoop cluster.

Again, we will use the foreign aid data set.

We will start by reading from the text file using del because the file is in comma separated value format.

We assign this source to the variable $foreignaid.

We want to assign a schema so we can refer to the country and sum fields easily and ensure sum is treated as a long rather than an int.

Next, we pipe the data from $foreignaid source to a consumer called group by and it groups the data by country because we specify the country field of the $ implicit variable as the argument to "group by".

We use "into" to say we want a result with a country field and a field containing the sum of all the values in the sum field for that country.

This produces a result that is in JSON format.

This concludes this lesson on Pig, Hive, and Jaql. Thank you for watching.