Connecting Spark to MongoDB

Created by Mathieu Besson (@mbesson)

Overview

- Credit
- What is Spark?
- The Spark - Hadoop ecosystem
- Connecting Spark to MongoDB

Credit

These slides are my personal notes, taken after the course M233: Getting Started with Spark and MongoDB from the MongoDB university, by Bryan Reinero (great course by the way).

What is Spark?

Apache Spark is a framework for processing large amounts of data. It performs batch processing, that is, it applies the same operation to all elements of a data set.

The Spark - Hadoop ecosystem

Level 1: Distributed data

the first component of the Spark - Hadoop system is the Hadoop File System (HDFS). This is where the data lives. HDFS maintains different copies of the data across the different nodes of the system, which makes it fault tolerant.

The Spark - Hadoop ecosystem

Level 2: Distributed resources

Resources (CPU, memory and data locality) must be managed across the entire cluster. That is, we must perform job scheduling, based on the available resources each node has to offer.

To do so, three ways are available to us.

The Spark - Hadoop ecosystem

Level 2: Distributed resources

Spark Standalone

Spark can run in a standalone mode, managing himslef the resources. This configuration is easy to set up and is used for learning and prototyping purposes.

For production deployments we will prefer either YARN of Mesos.

The Spark - Hadoop ecosystem

Level 2: Distributed resources

YARN

YARN stands for Yet Another Resource Negociator. This scheduler has been designed especially for MapReduce jobs

The Spark - Hadoop ecosystem

Level 2: Distributed resources

Mesos

Mesos is a more general solution that enables the execution of different kinds of distributed tasks, not only MapReduce. In the language of Mesos, the MapReduce operation is just an other type of distributed task it calls a ***framework***.

Mesos is a platform for fine-grained resource management and has been tightly developed with Spark at the Berkeley University.

The Spark - Hadoop ecosystem

Level 3: Distributed processing

This is this layer that encures a processing job is completed. Hadoop and Spark are frameworks for managing parallel processing. However they absolutly not model these jobs in the same way.

The Spark - Hadoop ecosystem

Level 3: Distributed processing

Hadoop

Hadoop uses the framework of MapReduce. It is compases of two functions:

- The map function (ex: filtering, transformations and projections)
- The reducer (ex: sums, grouping and averaging).

The Spark - Hadoop ecosystem

Level 3: Distributed processing

Spark

Spark is more flexible. Its primary data abstraction is the Resilient Distributed Dataset (RDD). A RDD is an in-memory distributed data structure on which you can perform a set of operations: ***transformations*** and ***actions***.

The fault tolerance of Apache Spark is based on the logging of the operations applied to an RDD: the ***RDD's lineage***. If anything that causes a partition to be lost occurs, that partition can be regenerated and the lineage can be replayed.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

The processing layer, that is MapReduce and Spark, are written in Java and Scala respectively. This is not the easiest language for your data scientist collegue. That is why, on top of the processing layer, you can found many DSL.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

Pig

Apache Pig is a high level framework for writing MapReduce programs. It is based on the Pig Latin language. Pig was initialy developed at Yahoo in 2006 and then transfered to the Apache Software Foundation in 2007.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

Hive

Apache Hive is a data warehouse infrastructure that allows analysis and quering jobs bases on a language close to SQL. It was developed by Facebook.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

Spark SQL

Spark SQL allows the execution of SQL queries.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

Spark shell

The Spark shell is a command line interface for running jobs in an interactive fashion.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

Spark Streaming

Spark Streaming is a realtime API that allows the treatment of data as it flows into the system.

The Spark - Hadoop ecosystem

Level 4: Domain Specific Languages (DSL)

And many others...

Connecting Spark to MongoDB

Before connecting Spark to MongoDB, make sure the spark-shell is in your PATH.

In order to get connected with MongoDB, the Spark hell needs two things:

- Where to get the Spark connector?
- Where to connect to MongoDB?

Connecting Spark to MongoDB

Getting the Spark connector

To tell the Spark sell where to find the MongoDb connector, we pass the following option:

```
--packages org.mongodb.spark:mongo-spark-connector_2.10:0
```

Connecting Spark to MongoDB

Where the connect to MongoDB

The place to find MongoDB can be passed to the shell through the *--config* option:

```
--conf "spark.mongodb.input.uri=mongodb://127.0.0.1/nasa.
```

Here, the parameter *--conf* is used two times, first to specify the shell where to read, second to tell Spark where to write the results.

Connecting Spark to MongoDB

Global command

The entire command to run a Spark shell connected to a mongo instance is :

```
spark-shell --packages org.mongodb.spark:mongo-spark-conn
```

Now it's time to get your hands dirty...