Developing analytics platform on OpenShift and leveraging Apache Spark as the analytics engine.

To find out more about my teams project visit - http://radanalytics.io

You can follow me on twitter: @prospect1010
AGENDA

GETTING STARTED WITH DATA PREPARATION

- PROBLEMS
- SCHEMAS AND DATA FORMATS
- INTRODUCTION TO CAMEL
- INTRODUCTION TO KAFKA
- SPARK SQL
- DATA STORAGE

PROCESSING DATA STREAMS

- OVERVIEW OF ARCHITECTURE
- SPARK STRUCTURED STREAMING INTEGRATION WITH KAFKA
- SPARK SQL AD HOC QUERY ON LIVE DATA IN PARQUET ON HDFS
- RUNNING SPARK IN CONTAINERS USING RADANALYTICS.IO TOOLING
- DEMO
GETTING STARTED WITH DATA PREPARATION
There are a number of choices when it comes to data sources: MySQL/MariaDB, Cassandra, RDBMS, MongoDB, etc.

Data might be coming from sensors, medical devices, mobile phones, etc.

The format of the data may be structured, semi-structured or unstructured.

Mining data in text documents is important to developers.
ETL WITH CAMEL

- Apache Camel is great for getting data from a source to sink
- Allows for you to transform that data. During transformation you have the option to write custom java code and pass data into it and the results will come back and then move on to the sink.
- It comes out of the box with a lot of components more than 200+ camel components that exist.
```java
public class FileCopierWithCamel {
    public static void main(String args[]) throws Exception {
        CamelContext context = new DefaultCamelContext();
        context.addRoutes(new RouteBuilder()
            .public void configure() {
                from("file:data/inbox?noop=true")
                    .bean(new ExampleBean(), "someMethod")
                    .to("file:data/outbox")
            }
        );
        context.start();
        Thread.sleep(10000);
        context.stop();
    }
}
```
BASIC CONCEPT

MESSAGING CONCEPT

- Producer - Is the sender of the message
- Consumer - Is the receiver of the message
COMPLEX INTEGRATION
KAFKA
PUBLISH SUBSCRIBE SYSTEM

- DISTRIBUTED, REPLICALED COMMIT LOG
- MESSAGES ARRIVE IN A TOPIC
- RICH ECOSYSTEM OF LIBRARIES TO INTEGRATE WITH:
  - APACHE CAMEL KAFKA COMPONENT
  - SPARK CONNECTS WITH KAFKA
  - KAFKA CONNECT
  - ETC.
SCHEMAS & DATA FORMATS
SCHEMAS

- Schemas are important and can change.
- Users shouldn’t get confused and we need a way to have documentation for these schemas.
- If they change we should be able to version it.
Our criteria is to have a data format that is splittable, has good compression and has a schema.

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Splittable?</th>
<th>Compression?</th>
<th>Schema?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>JSON</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Avro</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Parquet</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Thrift</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Protobuf</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
AVRO

WHAT IS IT?

- DATA SERIALIZATION FORMAT
- HAS AN EXTERNAL SCHEMA FILE USED TO READ DATA CONTAINED IN AVRO.
- DATA IS STORED ROW BY ROW. GOOD FOR ROW ORIENTED DATA.
- https://github.com/FasterXML/jackson-dataformats-binary IS A GOOD LIBRARY FOR WORKING WITH AVRO
- AVRO-TOOLING IS A JAVA BINARY JAR THAT CAN BE USED TO WORK WITH AVRO FILES. I WILL DEMO THIS
SCHEMA OUTPUT

```json
{
    "type" : "record",
    "name" : "CustomerBean",
    "namespace" : "com.redhat.data.analytics.model",
    "fields" : [ {
        "name" : "id",
        "type" : "int"
    }, {
        "name" : "name",
        "type" : [ "null", "string" ]
    } ]
}
```
PARQUET

WHAT IS IT?

- DATA SERIALIZATION FORMAT.
- SCHEMA IS IN THE FILE.
- COLUMNR FORMAT - GREAT FOR COLUMN ORIENTED DATA
- INTEGRATES WITH SPARK
hive
hive> CREATE TABLE rocknroll ( productCategory int
productName string, productPrice double,
productQuantity int) stored as parquet;
load data inpath '/demo-mar-27/rocknroll.parquet'
into table rocknroll;
hive> select * from rocknroll;
SPARKSQL + PARQUET

```python
spark = SparkSession.builder.master("local[*]").getOrCreate()
spark.read.load("hdfs://<HDFS_IP_ADDRESS>/orders/warehouse")
orders.createOrReplaceTempView("orders")
values=spark.sql("select * from orders")
print(values.groupBy('word').count().toJSON().count())
```
STREAM PROCESSING
WHAT IS SPARK

Apache Spark is a framework for distributed computing based on a high-level, expressive abstraction.
SPARK FEATURES

- Can run standalone, with yarn or mesos
- Has language bindings for Java, Scala, Python, and R
- Access data from JDBC, HDFS, S3 or regular filesystem
- Can persist data in different data formats: parquet, avro, json, csv, etc.
SPARK ARCHITECTURE

- MASTER
  - Driver
- Worker
  - executor
- Worker
  - executor
ORDER PROCESSING APPLICATION DEMO

- CAMEL WEB SERVICE PROVIDES A REST INTERFACE FOR RECEIVING ORDERS.
- INSTEAD OF JUST PERSISTING TO CASSANDRA WE ARE PRODUCING A MESSAGE OF AN ORDER EVENT TO KAFKA.
- FOR THIS DEMONSTRATION PYTHON CLIENT WILL SIMULATE CUSTOMERS ORDERING PRODUCTS
- DATA IS FEED INTO KAFKA USING SPARK STRUCTURED STREAMING AND STORED IN HADOOP IN PARQUET FORMAT.
- WE USE CHECKPOINTING FOR FAULT TOLERANCE
ORDER DEMO: ARCHITECTURE

Order Processing Application

Order Event

Kafka

Send Order Events To Spark

Spark Streaming

Spark SQL

Ad hoc Query

Store Data In Parquet File

HDFS
DEMO TIME!
RECAP

You learned:

- ETL and using apache camel for processing data.
- Data formats like Avro, Parquet and compared with other formats.
- Integrating kafka with spark structured streaming
- Ad hoc queries using jupyter notebook on data that is updating as orders are processed.
- Running spark in a container using radanalytics.io tooling on OpenShift
- Deploying a spark application having OpenShift build and deploy your streaming app
LINK TO SLIDES
