Import OSM Data

Download Data

https://download.geofabrik.de/index.html
该网站为各洲、各国家的openstreetmap开源数据库，例如，下载中国地区的数据：

```
wget https://download.geofabrik.de/north-america/china-latest.osm.pbf
```

以下过程中，我们使用的数据格式均为**osm.pbf。**

From osm.pbf To Dataframe

这里我们需要考虑两种情况：Spark3.x和Spark2.x.

Spark3.x

支持直接导入osm.pbf格式数据：

```
df = spark.read.format("osm.pbf").load("china-latest.osm.pbf")
```

参考：https://github.com/simplexspatial/osm4scala/blob/67864d8c9d69039e820c34cee2b0d6aede87fa8f/website/docs/spark-connector.mdx

Spark 2.x+（这是我现在在用的版本）

由于版本不支持osm.pbf格式的数据，所以需要用将osm.pbf数据转换为该版本可以支持的parquet格式，这里我们需要在终端运行以下代码安装osm-parquetizer工具，利用这个工具转换数据格式。
更加详细的说明请参考：https://github.com/adrianulbona/osm-parquetizer
以下为简版说明：

```
git clone https://github.com/adrianulbona/osm-parquetizer.git
cd osm-parquetizer
mvn clean package
java -jar target/osm-parquetizer-1.0.1-SNAPSHOT.jar china-latest.osm.pbf
```

上述代码运行时间较长，结果文件（三个）将储存在和china-latest.osm.pbf同一个文件下

```
-rw-r--r--  1 adrianbona  adrianbona   145M Apr  3 19:57 romania-latest.osm.pbf
-rw-r--r--  1 adrianbona  adrianbona   372M Apr  3 19:58 romania-latest.osm.pbf.node.parquet
-rw-r--r--  1 adrianbona  adrianbona   1.1M Apr  3 19:58 romania-latest.osm.pbf.relation.parquet
-rw-r--r--  1 adrianbona  adrianbona   123M Apr  3 19:58 romania-latest.osm.pbf.way.parquet
```

parquet的储存内容如下：

```
node
 |-- id: long
 |-- version: integer
 |-- timestamp: long
 |-- changeset: long
 |-- uid: integer
 |-- user_sid: string
 |-- tags: array
 |    |-- element: struct
 |    |    |-- key: string
 |    |    |-- value: string
 |-- latitude: double
 |-- longitude: double

way
 |-- id: long
 |-- version: integer
 |-- timestamp: long
 |-- changeset: long
 |-- uid: integer
 |-- user_sid: string
 |-- tags: array
 |    |-- element: struct
 |    |    |-- key: string
 |    |    |-- value: string
 |-- nodes: array
 |    |-- element: struct
 |    |    |-- index: integer
 |    |    |-- nodeId: long

relation
 |-- id: long
 |-- version: integer
 |-- timestamp: long
 |-- changeset: long
 |-- uid: integer
 |-- user_sid: string
 |-- tags: array
 |    |-- element: struct
 |    |    |-- key: string
 |    |    |-- value: string
 |-- members: array
 |    |-- element: struct
 |    |    |-- id: long
 |    |    |-- role: string
 |    |    |-- type: string
```

在python中即可利用pyspark导入以上osm数据为DataFrame格式（以way数据为例）：

```
sqlContext.setConf("spark.sql.parquet.binaryAsString","true")
```

```
df = sqlContext.read.parquet("./rawdata/china-latest.osm.pbf.way.parquet")
```

参考（osm数据分析，DataFrame，但是不用sedona已经完全可以处理了）：
https://github.com/bulutenesemre/OsmAnalysis
https://github.com/kkahloots/OpenStreetMapsHungary

From DataFrame to CSV

```
def array_to_string(my_list):
    return '[' + ','.join([str(elem) for elem in my_list]) + ']'

array_to_string_udf = udf(array_to_string, StringType())

osmWay = osmWay.withColumn('tags_array',
array_to_string_udf(osmWay["tags"]))
osmWay = osmWay.withColumn('nodes_array',
array_to_string_udf(osmWay["nodes"]))

osmWay.drop("tags").drop("nodes").write.csv('./csvData/osmWay.csv',header=
True)
```

这里的osmWay.csv是一个文件夹。用一下代码读入就是正常的一个dataframe，所以不用担心。但是特别提及的是有一些列是array形式，需要转换为string的格式才能储存，导入的时候也需要进行一定转换。

```
spark.read.csv("./csvData/osmWay.csv", header=True)
```

但上述保存csv到本地的方法花费时间较长，且容易报错内存溢出，下面采用直接将从Spark中导入数据到Hbase中去。

Spark-Hbase Connector

https://stackoverflow.com/questions/35228991/saving-pyspark-rdd-to-hbase-raises-attribute-error

```
host = 'localhost'
table = 'transaction_fee_table' #needs to be created before hand in hbase
shell
conf = {"hbase.zookeeper.quorum": host,
        "hbase.mapred.outputtable": table,
        "mapreduce.outputformat.class":
"org.apache.hadoop.hbase.mapreduce.TableOutputFormat",
        "mapreduce.job.output.key.class":
"org.apache.hadoop.hbase.io.ImmutableBytesWritable",
        "mapreduce.job.output.value.class":
"org.apache.hadoop.io.Writable"}
keyConv =
"org.apache.spark.examples.pythonconverters.StringToImmutableBytesWritable
Converter"
valueConv =
"org.apache.spark.examples.pythonconverters.StringListToPutConverter"

# data is a rdd
```

```
data.saveAsNewAPIHadoopDataset(conf=conf,
                               keyConverter=keyConv,
                               valueConverter=valueConv)
```

由于hbase环境还没配好，这段代码我还没有试验过