Data Analytics with Spark on Great Lakes

Jonathan Potter
jonpot@umich.edu

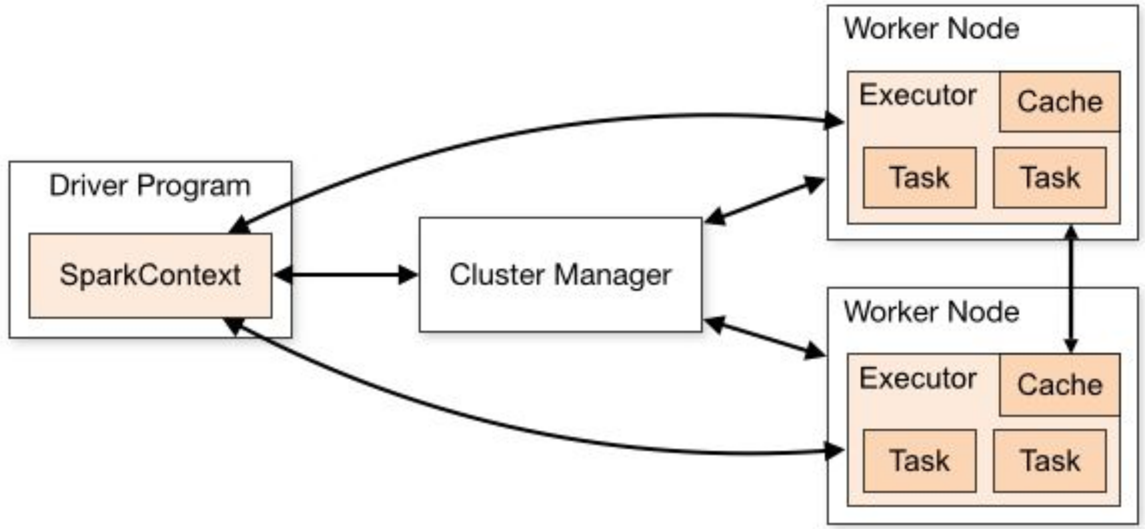
Agenda

- Introduction to Spark
- Getting Starting with Jupyter + Spark on Great Lakes
- Code Demo

Introduction to Spark

- Spark is "a unified analytics engine for large-scale data processing"
- Operates in-memory
- Interactive exploration of big datasets
- High-level APIs available in Java, Scala, Python and R

Spark Architecture



Spark Components

Driver

- Single process
- Runs main program of your application
- Distributes tasks to be run on executors

Executors

- Multiple processes
- Run tasks and deliver results back to driver

Spark DataFrame

Analogous to DataFrames in pandas

- Distributed collection of typed data in rows/columns
- Structured with schema, e.g. `df.printSchema()`
- Immutable

Lazy evaluation

- Transformations do not happen immediately
- Evaluate when result is needed

DataFrame Operations

Transformations

- Transform the data - lazy evaluated (not performed immediately)
- e.g. `select()`, `filter()`, `groupBy()`, `join()`, …

Actions

- Provides some output triggering evaluation
- e.g. `count()`, `show()`, `take()`, …

Spark Components

Class `SparkContext`

- Tells your application how/where to access the Spark cluster
- Reserves cores and memory on cluster
- A `sc` object, an instance of `SparkContext` class, is created automatically

Class `SparkSession`

- Entry point to work with RDD, DataFrame, and Dataset data structures
- A `spark` object, an instance of `SparkSession` class, is created automatically

Code Example

Create DataFrame with an explicit schema

```
df = spark.createDataFrame([
    (1, 2., 'string1', date(2000, 1, 1)),
    (2, 3., 'string2', date(2000, 2, 1)),
    (3, 4., 'string3', date(2000, 3, 1))
], schema='a long, b double, c string, d date')
```

Code Example

```
>>> df.show()
+---+---+-------+----------+
|  a|  b|      c|         d|
+---+---+-------+----------+
|  1|2.0|string1|2000-01-01|
|  2|3.0|string2|2000-02-01|
|  3|4.0|string3|2000-03-01|
+---+---+-------+----------+
```

Code Example

```
>>> df.printSchema()
root
 |-- a: long (nullable = true)
 |-- b: double (nullable = true)
 |-- c: string (nullable = true)
 |-- d: date (nullable = true)
```

Code Example

Create DataFrame from file

```
fruits = spark.read.csv('foo.csv', header=True,
         inferSchema=True)
```

Code Example

```
>>> fruits.show()
+-----+------+---+---+
|color| fruit| v1| v2|
+-----+------+---+---+
|  red|banana|  1| 10|
| blue|banana|  2| 20|
|  red|carrot|  3| 30|
| blue| grape|  4| 40|
+-----+------+---+---+
```

Code Example

```
>>> fruits.printSchema()
root
 |-- color: string (nullable = true)
 |-- fruit: string (nullable = true)
 |-- v1: integer (nullable = true)
 |-- v2: integer (nullable = true)
```

Getting Started with Spark

- Must be on-campus or connected through VPN
- Open in web browser https://greatlakes.arc-ts.umich.edu
- Launch the Jupyter + Spark app
  - Jupyter + Spark Basic
  - Jupyter + Spark Advanced

See Quickstart Examples

https://spark.apache.org/docs/3.1.2/api/python/getting_started/quickstart.html

Demo

Paycheck Protection Program Loan Dataset

- Loan data by state for loans of at least $150K
- For more background, see:

https://towardsdatascience.com/plotting-w-pandas-and-ppp-loan-data-2d8d1995a626

Get Data

Download data from terminal session

```
ssh jonpot@greatlakes.arc-ts.umich.edu
wget http://www-personal.umich.edu/
     ~arburks/workshops/pyspark/data/
     PPP_data_150k_plus.csv.gz
```

Load Data as DataFrame

- Launch Jupyter + Spark Basic
- Create a new Jupyter Notebook with a Python 3 kernel
- Load the compressed CSV file as below

```
df = spark.read.csv(
       'PPP_data_150k_plus.csv.gz',
       header=True, inferSchema=True)
```

Preview Data

```
# See the schema
df.printSchema()

# Display first 20 rows
df.show()

# Display first 3 rows without truncation
df.show(3, truncate=False)
```

Select and Drop Nulls

```
# Select columns of interest
subset = df.select(
           'City', 'State',
           'BusinessName', 'JobsRetained')

# Drop nulls
nonNull = subset.dropna()
nonNull.count()
```

Analysis

How many businesses in Ann Arbor, MI used PPP?

```
annArborPPP = nonNull.filter(
                (nonNull.City == 'ANN ARBOR') &
                (nonNull.State == 'MI'))

annArborPPP.count()
```

Analysis

Which businesses in Ann Arbor, MI used PPP?

```
# Display first 100 rows of data
annArborPPP.show(100, truncate=False)

# Show top 10 business by most jobs retained
annArborPPP.orderBy('JobsRetained', ascending=False)
  .show(10, truncate=False)
```

Analysis

Show top 10 states by most jobs retained

```
from pyspark.sql import functions as f

groupedByState = nonNull.groupBy('State')
jobsByState = groupedByState.agg(f.sum('JobsRetained')
                .alias('JobsRetained'))
jobsByState.orderBy('JobsRetained', ascending=0).show(10)
```

Stopping Notebook

Gracefully stop Jupyter Server to avoid excess charges.

- Close Jupyter Notebook
- On Jupyter Server tab, click `Quit` button

Gracefully stopping Jupyter Server this way also cleans up problematic temporary files.

Conclusion

Thank You

Jonathan Potter
jonpot@umich.edu