A quick introduction to Apache Spark

Joe Cauteruccio
www.linkedin.com/in/joecauterucciojr
joecjr.com

Manager, Research Group – Data Science Team
DigitasLBi

DigitasLBi

11.18.2014

What is Spark

Spark is an, accessible, flexible, and speed-optimized cluster computing framework.

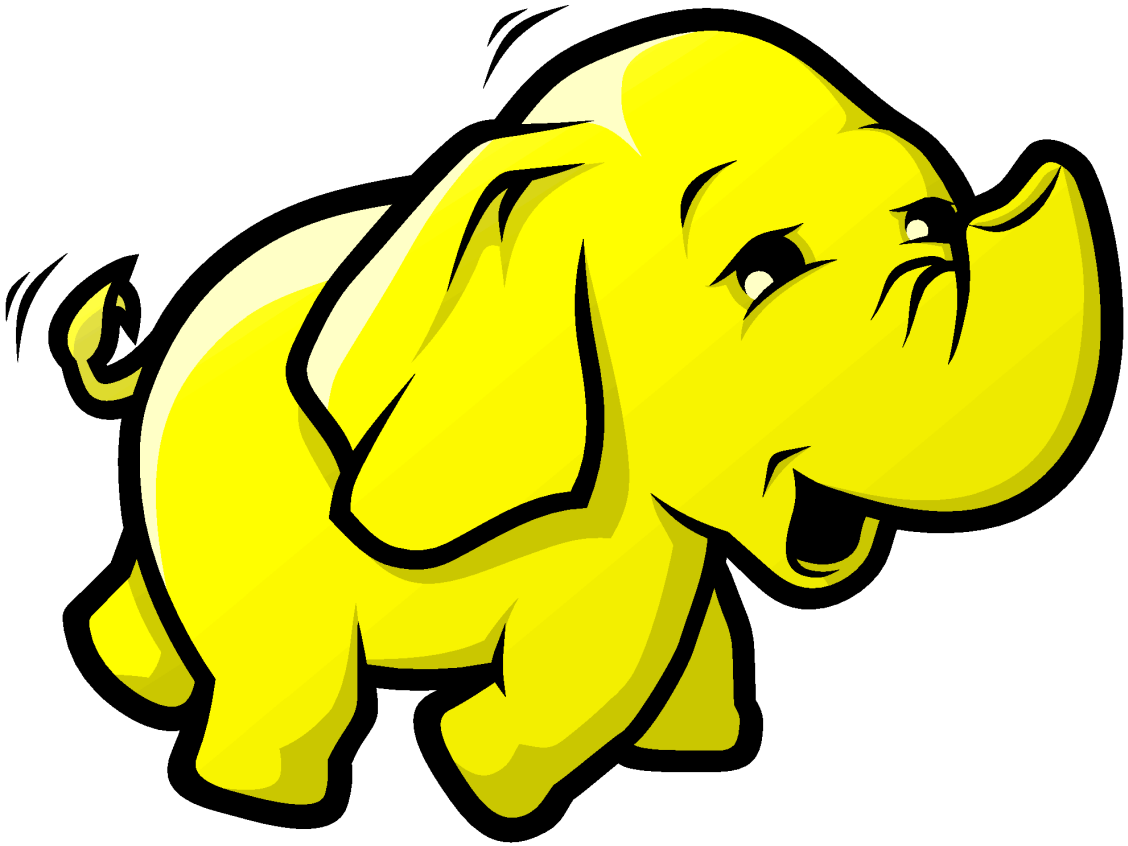
Accessible: utilize Spark from Python, Scala, Java

Flexible: Spark can play a role in production systems, **interactive** analysis and everything in between.

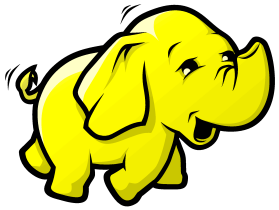
Speed: paradigm is designed to enable the fast iterative processing required for machine learning and analytics. Jobs can be run in Memory.

What is Spark

What about Hadoop?



What about Hadoop?



A platform for cluster computing:

- Provides a file-system HDFS
- Libraries needed to manage/support a cluster
- Manages computing resources

MapReduce

A programming paradigm for Data Processing:

- Map ➡ Shuffle ➡ Reduce

A alternative framework for Data Processing:

- Iterative, Interactive, Fault Tolerant
- Has its own ecosystem surrounding it

Terminology

Driver – The master node side process responsible for defining your analytic tasks

Executor – The slave node side process responsible for doing the heavy work

RDD – **R**esilient **D**istributed **D**ataset. This is the main data object in Spark.

Transformations – (roughly) operations that change our data

Actions – (roughly) operations that return output

A simple example: Conversion Count

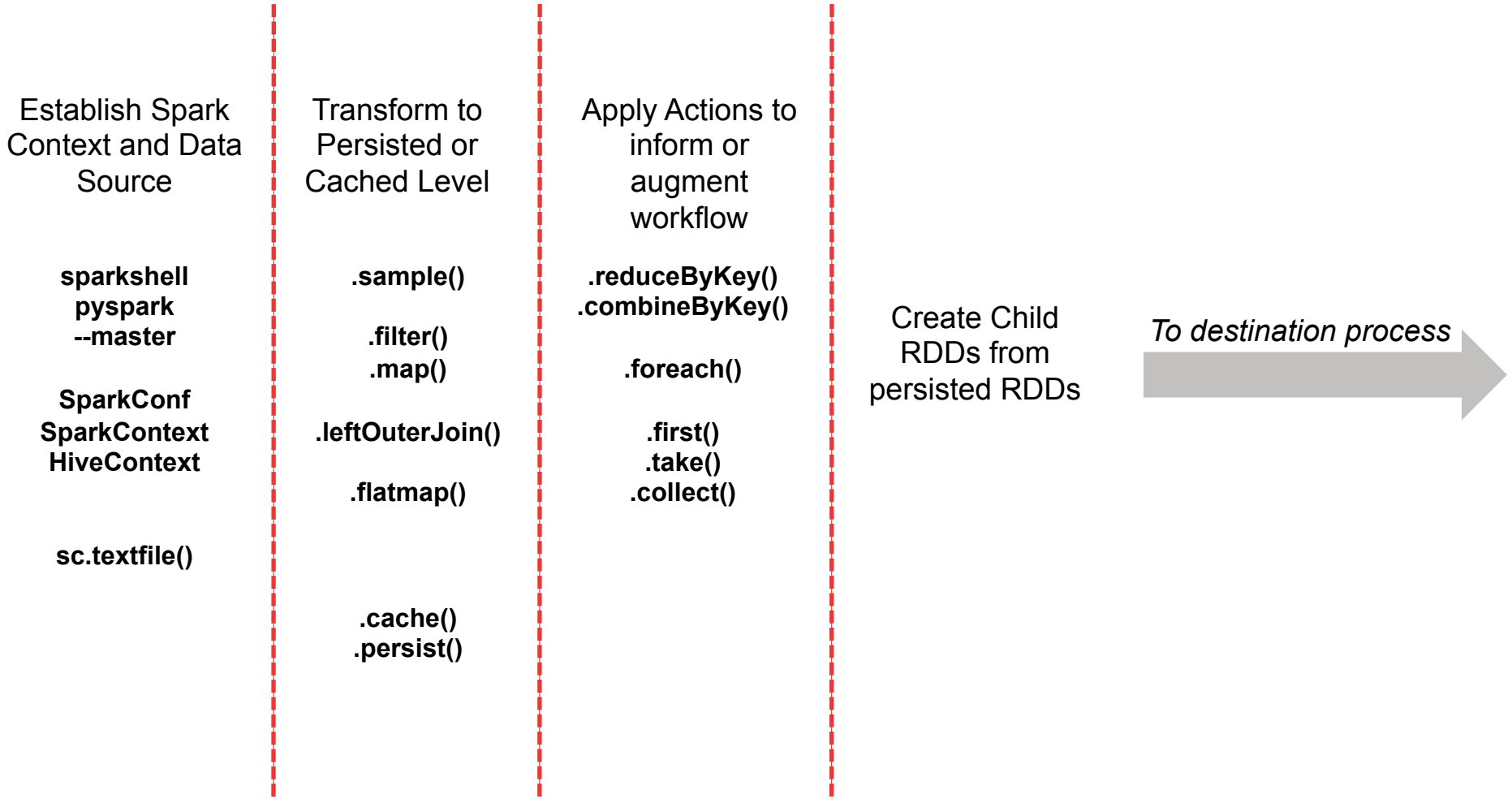
In Hive:

```
select site_id, count(*) from limited_activites where dt >= '2014-10-12' and dt <= '2014-10-18' and activity_sub_type = 'XXXXX'
```

In Spark:

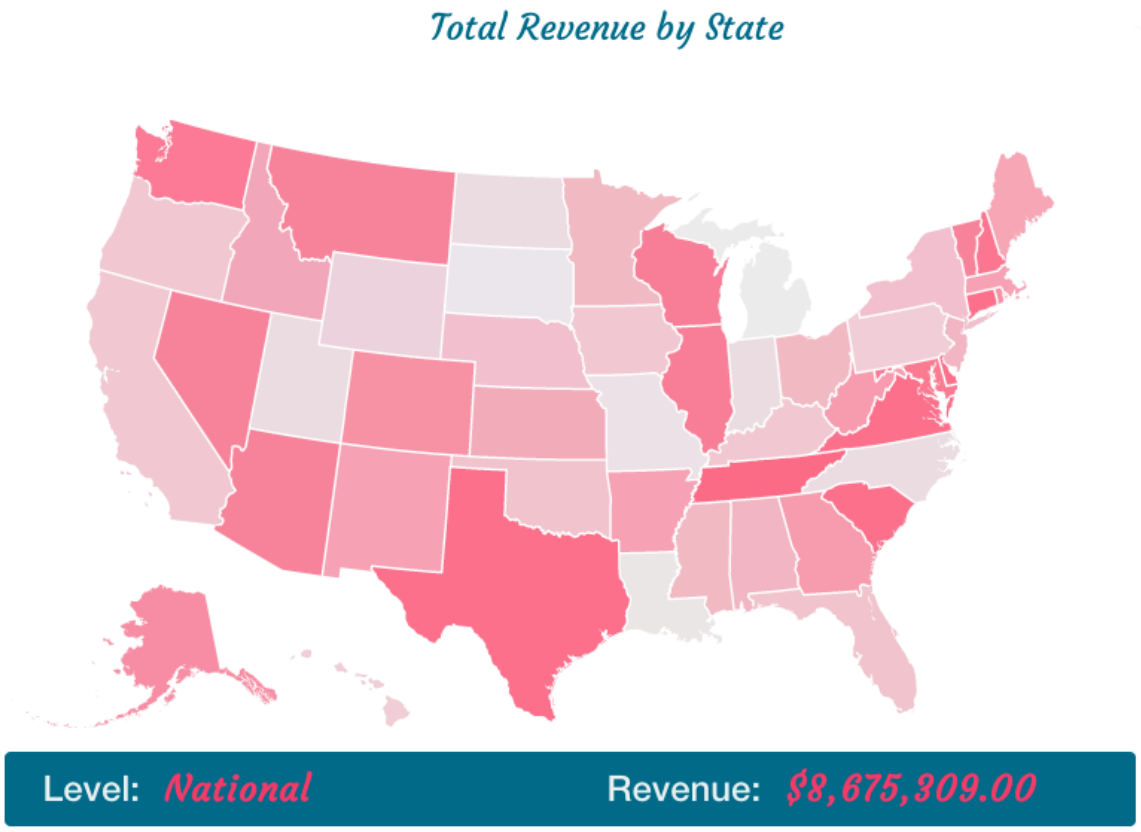
```
activty_info = sc.textFile('PATH/2014-10-[0-3][0-9]')
transaction_info = activity_info.filter(lambda s: 'XXXX' in s).map(lambda x: (x.split(u'\ufffd')[9], 1)).reduceByKey(lambda x, y: x + y)
transaction_info.cache()
```

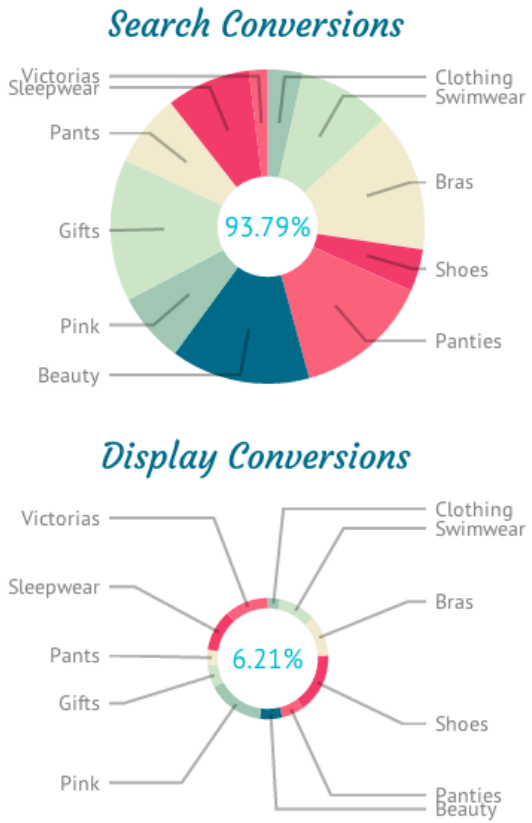
A Sample Spark Workflow



A destination process…

Example: Visualizing National Orders





*Sorry, the data displayed here is fake…

DigitasLBi

Passing Functions to Map

```
# function to pull order fields
def process_order(ord):
    # Get order skus
    me = ord.split('\x01')
    skus = re.search('u5=(.*);u3', me[0], re.IGNORECASE)
    if skus:
        skus = skus.group(1).split(',')
    qnty = re.search('u3=(.*);u1', me[0], re.IGNORECASE)
    if qnty:
        qnty = [int(q) for q in qnty.group(1).split(',')]
    info = (me[1], (me[len(me)-1], skus, qnty))
    return info

def process_trans(ord):
    me = ord.split(u'\ufffd')
    rec = (me[24], (me[13], me[21]))
    return rec

def process_skus(ord):
    me = ord.split('|')
    rec = [me[0], me[1], me[2]]
    return rec
```

Passing Functions to Map

```
# Process File 1
order_info = of_week_1.map(process_order)

# Process File 2
transaction_info = activity_info.filter(lambda s: 'XXXXX' in s).map(process_trans)

# Join
full_trans_info = order_info.leftOuterJoin(transaction_info)
```

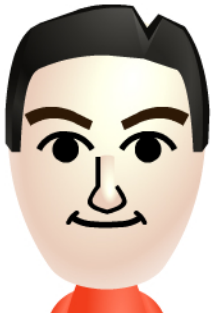
Mllib: Machine Learning Library

- Facilities for most common machine learning algorithms
- from pyspark.mllib.___ import ___
- RDDs + MLlib specific data types (Vectors, Arrays, Labeled)
- Definitely deserves its own talk!

https://spark.apache.org/docs/1.1.0/mllib-guide.html
http://stanford.edu/~rezab/sparkworkshop/slides/xiangrui.pdf
http://ampcamp.berkeley.edu/big-data-mini-course/movie-recommendation-with-mllib.html

Lastly a shameless plug…

Me



Joe Cauteruccio

I work for a magical Unicorn

DigitasLBi
Global Digital Marketing and Technology Company

www.digitaslbi.com

Resources and References

Zaharia, Et. al. Spark: Cluster Computing with Working Sets. *University of California, Berkley*. June 2010, http://www.cs.berkeley.edu/~matei/papers/2010/hotcloud_spark.pdf

Zaharia, Et. al. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. *NSDI 2012*. April 2012, http://www.cs.berkeley.edu/~matei/papers/2012/nsdi_spark.pdf

Karau, Et. Al. Learning Spark: Lightning-Fast Big Data Analytics. *O'Reilly Media.* June 2014, http://shop.oreilly.com/product/0636920028512.do

UC Berkley AMP Camp: http://ampcamp.berkeley.edu/