FROM NOTEBOOK TO CLOUD NATIVE
A MODERN PATH FOR DATA-DRIVEN APPLICATIONS

Michael McCune - msm@redhat.com
WHO AM I

- embedded to orchestration
- emerging technology @ Red Hat
- big data on OpenStack and OpenShift
WE'RE TALKING ABOUT A JOURNEY

#StrataData
LET'S TALK CONTAINERS

```
$SPARK_HOME/bin/spark-class \
org.apache.spark.deploy.worker.Worker \
master:7077
```

- **pid**
- **root**
- **net**

/
LET'S TALK CONTAINERS

```bash
$SPARK_HOME/bin/spark-class
  org.apache.spark.deploy.worker.Worker
  master:7077
```

container runtime

```
pid
root
net
```

```
/smp/foo
```

```
SPEED LIMIT
55
```

#StrataData
LET'S TALK CONTAINERS

```bash
$SPARK_HOME/bin/spark-class
 org.apache.spark.deploy.worker.Worker
 master:7077
```

- **pid**
- **root**
- **net**
WHAT ABOUT MICROSERVICES?
WHAT ABOUT MICROSERVICES?

- modular and flexible
- stateful vs stateless
- network resilient

#StrataData
WHAT ABOUT MICROSERVICES?
WHAT ABOUT MICROSERVICES?

#StrataData
WHAT ABOUT MICROSERVICES?
INTELLIGENCE ARCHITECTURE

- events
- databases
- cloud storage

transform

transform

transform

federate

train

models

publish

web

mobile

archives

#StrataData
INTELLIGENCE ARCHITECTURE

Resource manager

App 1

App 2

App 3

App 4

Cluster scheduler

Spark executor

Spark executor

Spark executor

Spark executor

Spark executor

Spark executor

Cloud storage

Databases

#StrataData
INTELLIGENCE ARCHITECTURE

Resource manager

App 1
App 2
App 3
App 4

Cluster scheduler

Spark executor
Spark executor
Spark executor
Spark executor
Spark executor
Spark executor

Cloud storage
Databases

#StrataData
INTELLIGENCE ARCHITECTURE

Resource manager

App 1

App 2

App 3

App 4

App 5

App 6

Cloud storage

Databases

#StrataData
INTELLIGENCE ARCHITECTURE

Resource manager

App 1

App 2

App 3

App 4

App 5

App 6

Cloud storage

Databases

#StrataData
INTELLIGENT APPLICATION LIFECYCLES

#StrataData
DEMO
GOING CLOUD NATIVE

#StrataData
WHAT IS CLOUD NATIVE?

- Containerized
- Dynamically orchestrated
- Microservice oriented
- cncf.io/about/faq
WHAT WILL YOUR APPLICATION DO?

Ingest  Process  Publish
STORYBOARD YOUR ARCHITECTURE

Browser → HTTP Server → Analytics

Data

#StrataData
CONSTRUCT YOUR APPLICATION

React <-> Flask -> Spark

Parquet
CASE STUDY: VAR-SANDBOX
GENERAL ARCHITECTURE

React → Flask → PySpark → Parquet → Spark → Spark → Spark → Spark
HOW WAS IT BUILT?

```python
def portfolio_value(pf):
    """Given a dictionary of stock values, return the total value."""
    return sum([v for v in pf.values()])

def seeds(count):
    """Return a list of random values of the specified length."""
    return [random.randint(0, 1 << 32 - 1) for i in range(count)]
```

from random import randint, seed

```python
def random_portfolio(symbols):
    result = {}
    for s in symbols:
        result[s] = prices[s] * (randint(1, 1000) * 11)
    return result
```
DEMO
MOVING OUT OF ALPHA

OpenShift

React → Flask → Flask → PySpark → Spark → Spark → Spark

Cloud storage

#StrataData
LESSONS LEARNED
Value-at-risk calculations

The basic idea behind the value-at-risk calculation is that we’re going to look at the historical returns of a portfolio of securities and run many simulations to determine the range of returns we can expect from these. We can then predict, over a given time horizon, what our expected loss is at a given probability, e.g., we might say that there is less than a 10% chance that the portfolio will lose more than $1,000,000.

Note that this is a didactic example and consequently makes some simplifying assumptions about the composition of the portfolio (i.e., only long positions in common stocks, so no options, dividends, or short selling) and the behavior of the market (i.e., day-to-day return percentages are normally-distributed and independent). Do not use this code to guide actual investment decisions!

Basic setup

Here we import the pyspark module and set up a SparkSession.

```python
In [1]:
import pyspark
from pyspark.context import SparkContext
from pyspark.sql import SparkSession, SQLContext

spark = SparkSession.builder.master("local[*]").getOrCreate()
```
IT'S NOT ALL ROSES

- tricky Spark configurations
- notebooks add infrastructure
- code can be difficult
YOUR GREATEST TOOL ON THE JOURNEY

#StrataData
TRY IT FOR YOURSELF!

https://radanalytics.io hosts tutorials, example projects, and more
Keep in touch: @FOSSjunkie

#StrataData