

Jupyter and Spark on Mesos: Best Practices

June 21st, 2017

scrapinghub

Agenda

- About me
- What is Spark & Jupyter
- Demo
- How Spark+Mesos+Jupyter work together
- Experience
- Q & A

About me

- Graduated from EE @ Tsinghua Univ.
- Infrastructure Engineer @ Scrapinghub
- Contributor @ Apache Mesos & Apache Spark

Apache Spark

- Fast and general purpose cluster computing system
- Provides high level APIs in Java/Scala/Python/R
- Integration with Hadoop ecosystem



Why Spark

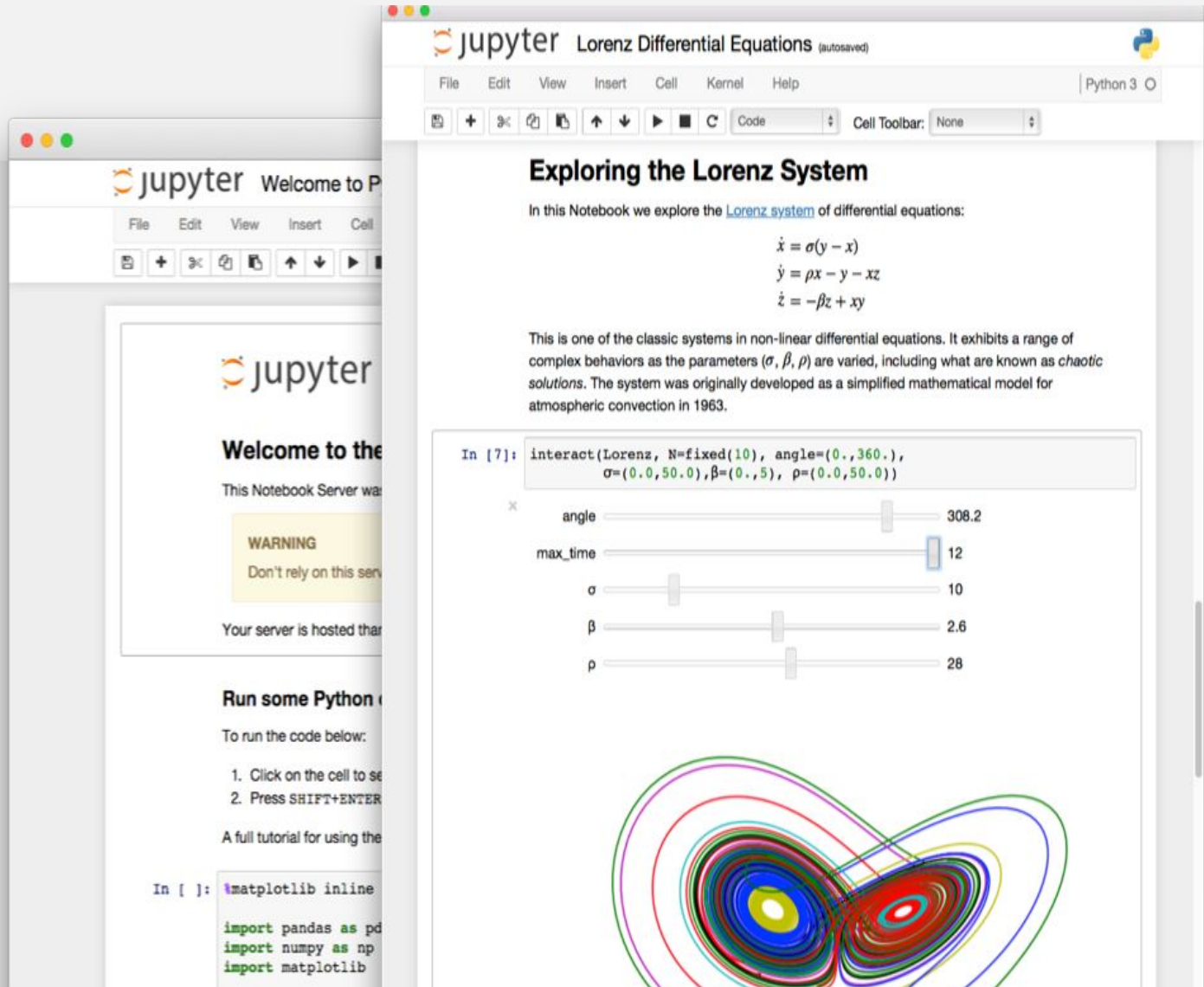
- Expressive API for distributed computing
- Support both streaming & batch
- Low level API (RDD) & High level DataFrame/SQL
- First-class Python/R/Java/Scala Support
- Rich integration with external data sources: JDBC, HBase, Cassandra, etc.

Jupyter Notebook Server

- IPython shell running in the web browser
- Not only code, also markdown & charts
- Interactive
- Ideal for demonstration & scratching

<http://jupyter-notebook.readthedocs.io/en/latest/notebook.html>

Jupyter Notebook Server



The image displays two overlapping windows of a Jupyter Notebook server. The background window shows a 'Welcome to the Jupyter Notebook Server' page with a warning message: 'WARNING: Don't rely on this server. Your server is hosted through a third party.' Below this, it instructs the user to 'Run some Python code' and provides instructions on how to execute code cells.

The foreground window shows a notebook titled 'Lorenz Differential Equations (autosaved)'. The notebook content includes:

Exploring the Lorenz System

In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

In [7]: `interact(Lorenz, N=fixed(10), angle=(0.,360.), sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))`

The interactive interface shows sliders for the following parameters:

- angle: 308.2
- max_time: 12
- σ : 10
- β : 2.6
- ρ : 28

Below the sliders is a plot of the Lorenz attractor, showing its characteristic butterfly shape with multiple overlapping trajectories in various colors.

Recap

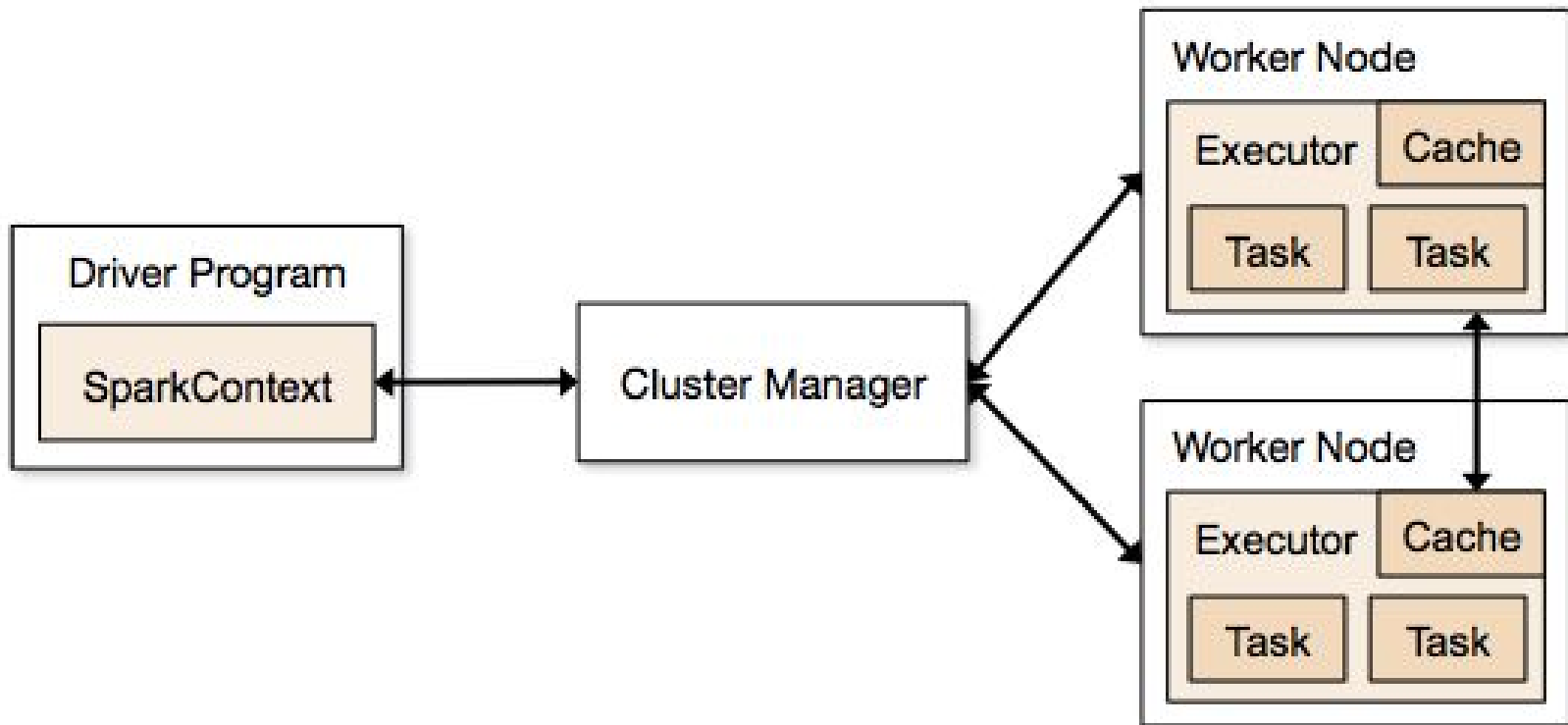
Prev:

- Introduction to Spark
- Introduction to Jupyter Notebook Server

Next:

- Why Spark on Mesos
- Why Spark+Mesos+Jupyter

Why Spark on Mesos



Why Spark on Mesos

- Run Spark drivers and executors in docker containers
(avoid python dependency hell)
- Run any version of spark!
- Making use of our existing mesos cluster
- Reuse the monitoring system built for mesos

Why Spark + Jupyter Notebook

- Run in Local computer
 - Not enough storage capacity for large datasets
 - Not enough compute power to process them
- Run in company cluster
 - takes too long to set up
 - Hard to debug (only through logs)

Why Spark + Jupyter Notebook

- Run in Notebook
 - No need to set up - just on click
 - Easy to debug
 - Full access to the cluster's compute power

Recap

Prev:

- Why Spark on Mesos
- Why Spark+Mesos+Jupyter

Next:

- Demo



DEMO



Recap

Prev:

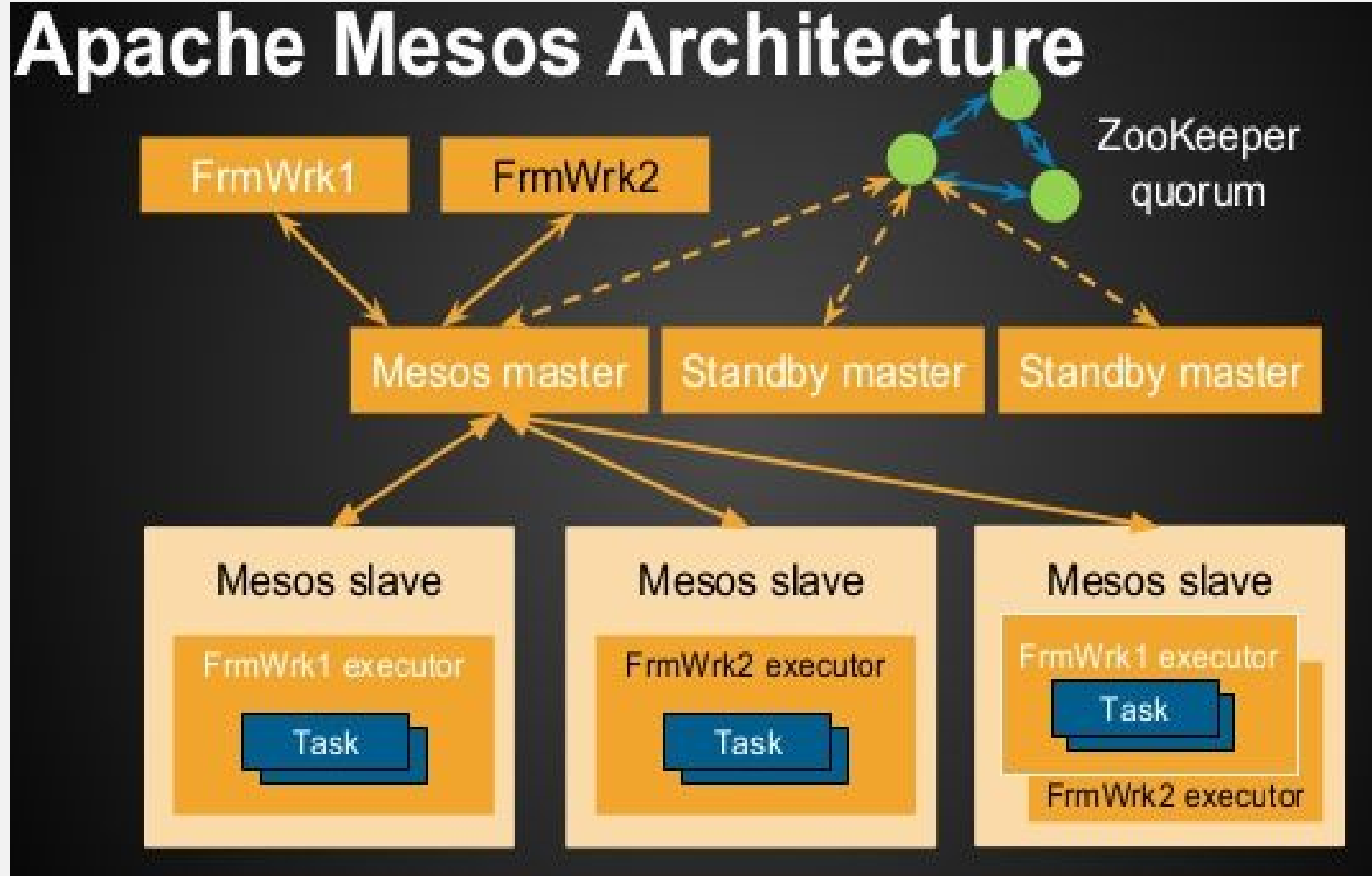
- Demo

Next:

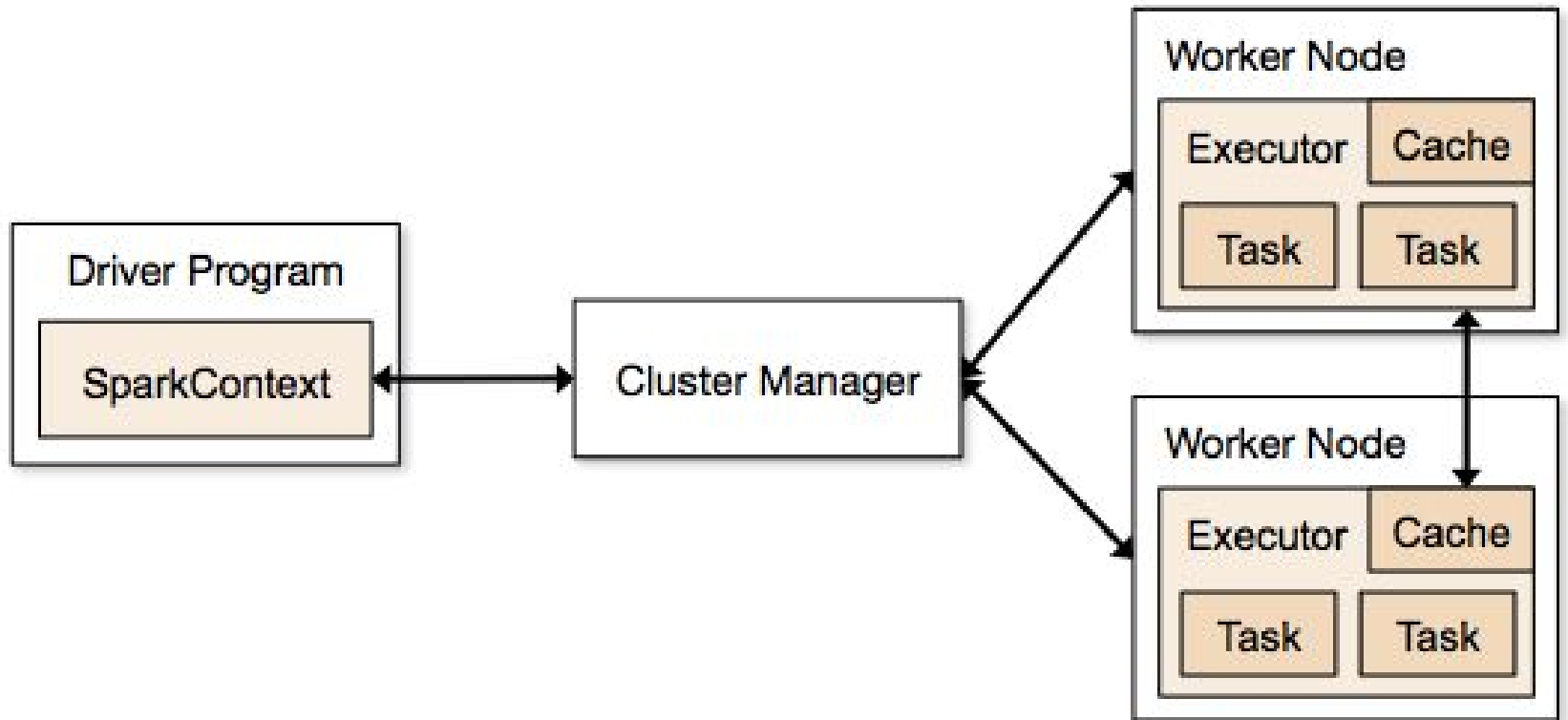
- How Spark and Mesos work together
- Experience & Caveats



Mesos & Spark: Mesos Architecture



Mesos & Spark: Spark Architecture



Mesos & Spark

- A Spark app/driver = a Mesos framework
- Spark executors = Mesos tasks

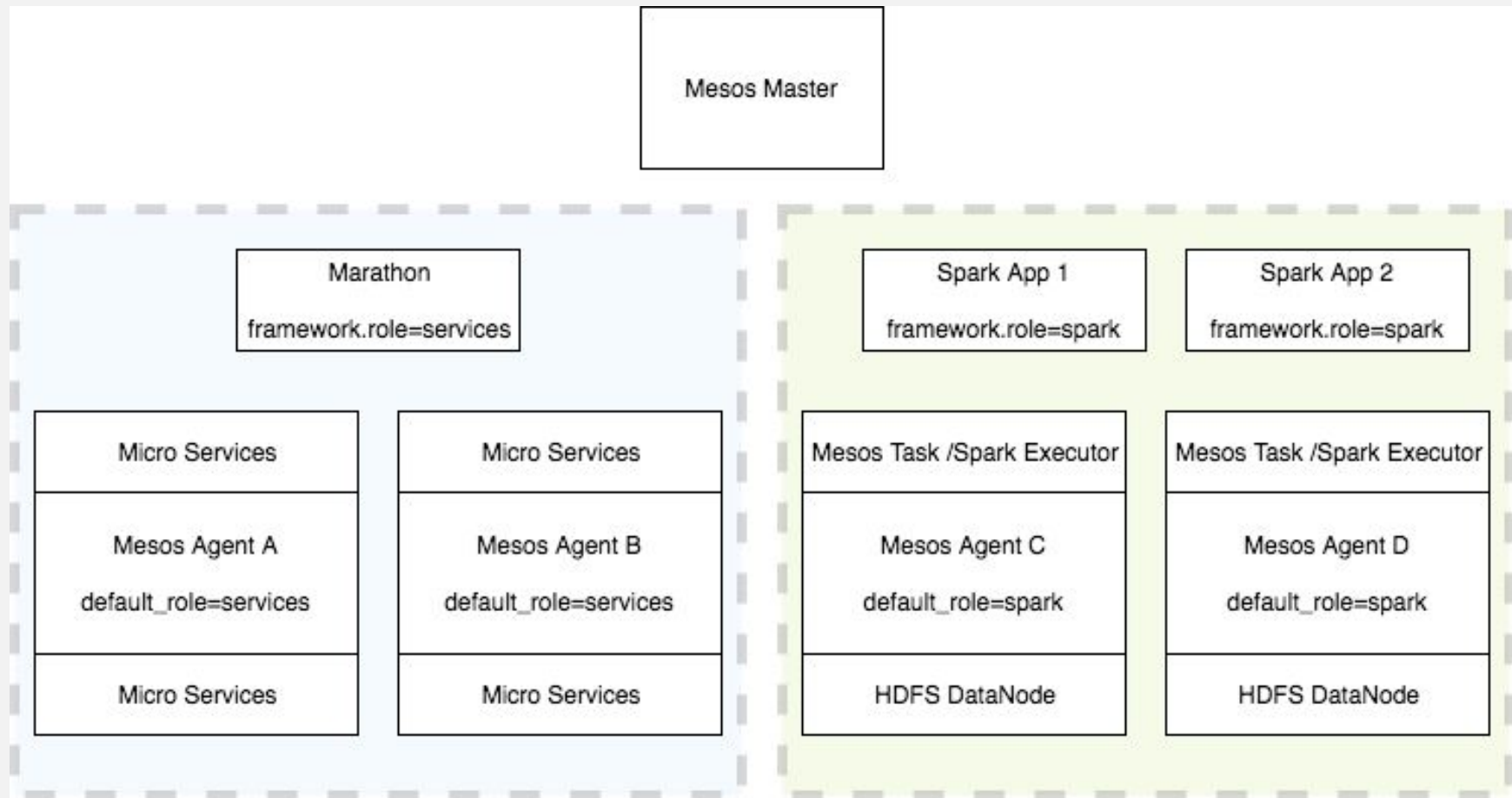


Mesos & Spark: Experience

- Single Cluster
- Marathon for long running services
- Constraints to pin spark tasks on certain nodes



Experience - Single Cluster



Experience - Single Cluster

- Pros & Cons

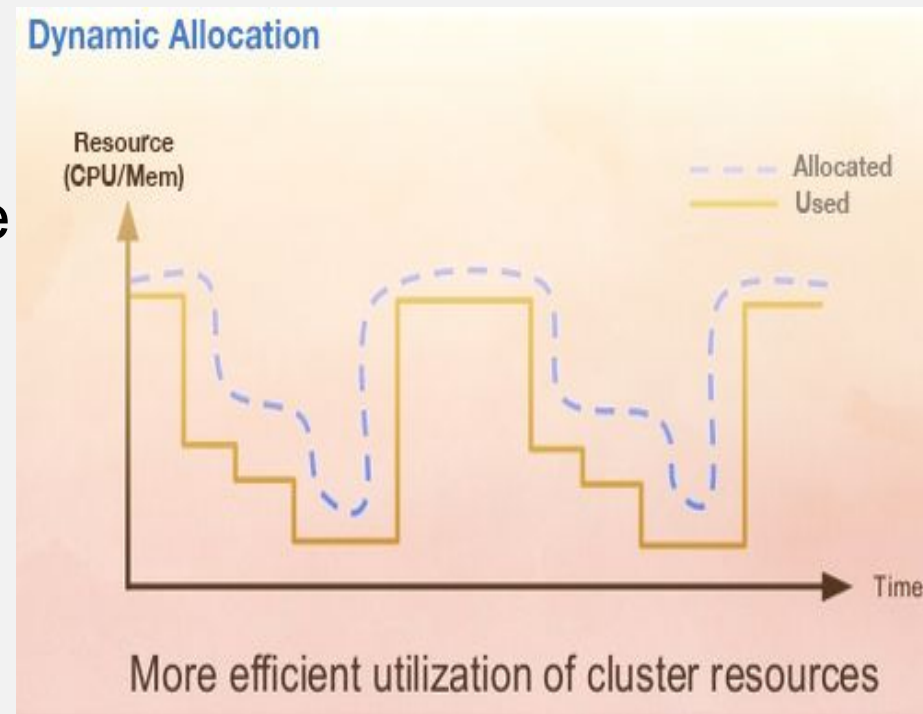


Experience: Dynamic Allocation is a must

- People tend to leave their spark executors running, even if they end their day of work
- No resource available for new launched spark apps, even if the cluster is doing no work
- Enable dynamic allocation: idle spark executors are terminated after a while

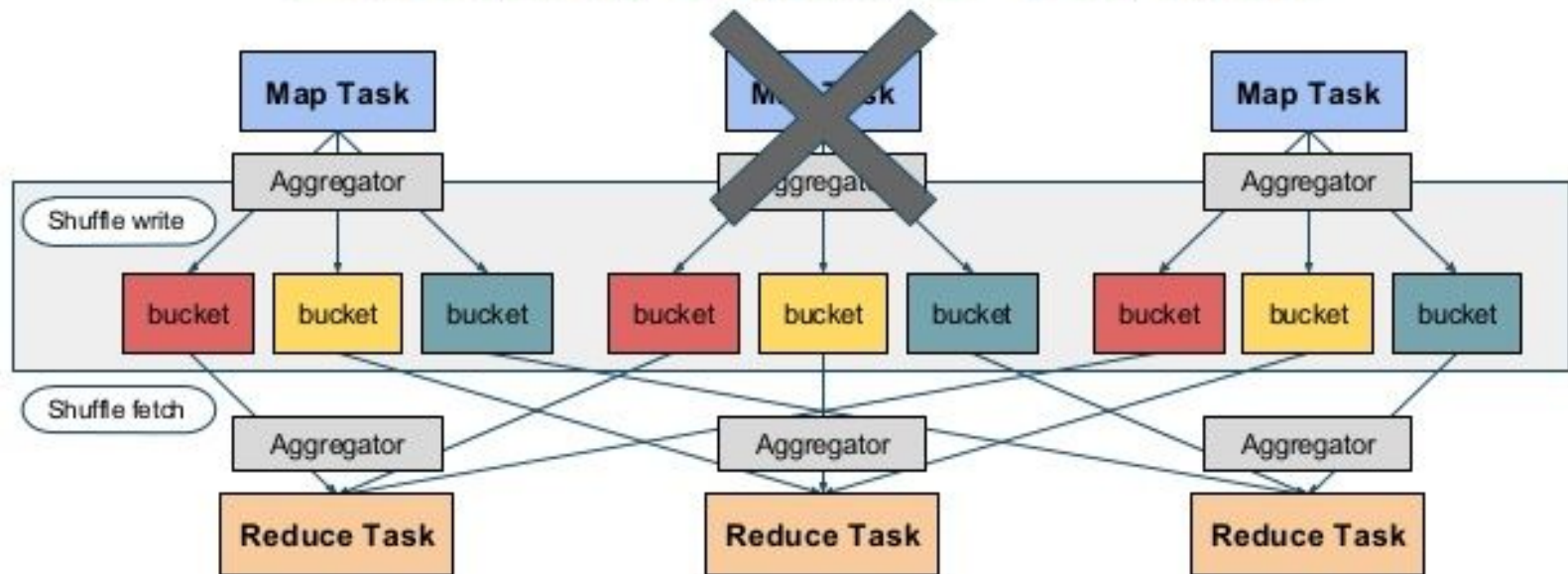
Spark Dynamic Allocation

- Spark executors are:
 - Killed after being idle for a while
 - Launched later when there are tasks waiting in the queue
- Requires long-running “spark external shuffle service” on each mesos node



Spark Dynamic Allocation - External Shuffle Service

External Shuffle Service



Experience: Battery-included docker base image

- Basics:
 - libmesos
 - java 8
- Libs:
 - Python 2 & Python 3 & libraries
 - Hadoop jars for AWS, kafka jars
- Configuration:
 - Resource spec (cpu/ram) for spark driver/executors
 - Dynamic allocation
 - Constraints: pin spark executors to colocate with HDFS DataNodes

Experience: Battery-included docker base image

```
1 # Default system properties included when running spark-submit.
2 # This is useful for setting default environmental settings.
3
4 spark.files                file:///etc/hadoop/conf/hdfs-site.xml,file:///etc/hadoop/conf/core-site.xml
5 spark.mesos.coarse         true
6 spark.cores.max            2
7 spark.executor.cores       1
8 spark.executor.memory      1g
9 spark.mesos.role            spark
10 spark.mesos.constraints    role:spark-node
11 spark.mesos.uris           file:///root/.dockercfg
12
13 #####
14 # Settings for dynamic resource allocation, DO NOT EDIT IT unless you know what you're doing.
15 #####
16 spark.dynamicAllocation.enabled      true
17 spark.shuffle.service.enabled        true
18 # default 0
19 spark.dynamicAllocation.initialExecutors 2
20 # default 1min
21 spark.dynamicAllocation.executorIdleTimeout 10m
22 # default infinity
23 spark.dynamicAllocation.cachedExecutorIdleTimeout 1h
24 # default infinity
25 spark.dynamicAllocation.maxExecutors 10
26 # The executor must share the same data dir with the external shuffle service
27 # See https://issues.apache.org/jira/browse/SPARK-17555
28 spark.mesos.executor.docker.volumes    /tmp/spark:/tmp/spark:rw
29 spark.local.dir                         /tmp/spark
```

Experience: Save Jupyter notebooks in database

- Jupyter does not support saving notebooks in databases
- but it provides a pluggable storage backend API
- pgcontents: Postgres backend, open sourced by Quantopian
- we ported it to support MySQL (straightforward thx to SQLAlchemy)

<https://github.com/quantopian/pgcontents>

<https://github.com/scrapinghub/pgcontents/tree/mysql>

Recap

Prev:

- How Spark and Mesos work together
- Experience & Caveats
 - Role & Constraints
 - Dynamic Allocation is a must

Next:

- Looking into the future
- Q & A

Looking into the Future

- Resource isolation between notebooks
- Python environment isolation between notebooks



Spark JobServer

- Learning spark+python is a bit too much for people like sales & QA
- But almost everyone knows about SQL
- So why not we just provide a web ui to execute spark sql?

Spark JobServer

- Much like AWS Athena, but tailored to our own use cases

The screenshot displays the Spark JobServer interface. On the left is a sidebar with navigation options: Query, Table Info, Schema, and Settings. The main area contains a query editor with the following SQL query:

```
select *  
from table178  
where url is null
```

Below the query editor, a table of results is displayed. The table has the following columns: `_key`, `_ts`, `_type`, `address`, `city`, `coming_soon`, `rent`, `sqft`, and `state`. The results are as follows:

<code>_key</code>	<code>_ts</code>	<code>_type</code>	<code>address</code>	<code>city</code>	<code>coming_soon</code>	<code>rent</code>	<code>sqft</code>	<code>state</code>
124600/6/30/126	1495802224813	RentItem	3856 Nw 107th Way...	Sunrise	1	2000	1656	FL
124600/6/30/131	1495802224813	RentItem	2050 Ne 60th Stre...	Fort Lauderdale	null	2650	1570	FL
124600/6/30/168	1495802224813	RentItem	6331 NW 72nd Plac...	Parkland	null	3240	2281	FL
124600/6/30/254	1495802224813	RentItem	128 Cortes Ave, R...	Royal Palm Beach	1	2050	1591	FL
124600/6/30/260	1495802224813	RentItem	122 Mayorca Court...	Royal Palm Beach	1	2050	1429	FL
124600/6/30/266	1495802224813	RentItem	10189 Flag Drive,...	Palm Beach Gardens	1	2250	1417	FL

At the top right of the query editor, there is a "More Actions" button. Below the query editor, there is a "Run & Download the result" button.

Spark JobServer

- Much like AWS Athena, but tailored to our own use cases

Q Query		
i Table Info		
Schema		
Settings		
	Column	Type
	_key	string
	_ts	long
	_type	string
	address	string
	city	string
	coming_soon	long
	rent	string
	sqft	string
	state	string
	street	string
	url	string
	zip_code	string

A faint, light-colored spider web graphic is centered on the left side of the slide, serving as a background element. The web consists of several concentric circles and radial lines, with some lines ending in small circular nodes.

Q & A