René Jäkel
Center for Information Services and High Performance Computing (ZIH)

Introduction to data analytics with Apache Spark + Hands-On/Walkthrough

Science at Extreme Scales: Where Big Data Meets Large-Scale Computing Tutorials

Sept. 17 2018, IPAM

Course overview

Part 1 – Challenges

- Fundamentals and challenges in Big Data ((big) data analytics)
- Big Data methods useful for managing/transforming data and provide fast access to analytics functionality
- Complex analytics data-driven workflows vs. static parallel applications

Part 2 – Second generation (big) data processing

- Extending the Hadoop ecosystem
- Possible future directions

Part 3 – Introduction to Apache Spark and Hands-On

Course overview

What’s the purpose of the Hands-on?

- Approach – use Apache Spark as generic framework for data manipulation and analysis
- Not try to convince you to use Spark in general
- But: be aware of current trends and available methods
- Goal: you should be able to get an overview about functionalities, smaller investment into new/other tools (Flink, Mahout, Beam….)

Hands-On Preparation

Requirements for Hands-On part

If your want to try out things during the session (and later on) I highly recommend to install Apache Spark on your local system. The setting assumes that your system has a Java development kit installed (e.g. Open-JDK). Following, this is as small cookbook for Linux-based systems to get started (guidelines for Mac and Windows-Users are below):

- Create a new directory somewhere for this tutorial (e.g. ‘`tutorial`’)
- Go to spark website and download latest base release: http://spark.apache.org/downloads.html

  (the latest release from June 8 2018 should do – Apache Spark 2.3.1)
  - Unpack the preinstalled version into the tutorial directory
  - Add the path to the `bin` subdirectory into your `.bashrc` file
- Get the latest Anaconda release (or user your python environment, if present) from here:

  https://repo.continuum.io/archive/
  - Install the release on your system
  - Install the ‘jupyter’ and ‘findspark’ packages via anaconda:

```
> conda install jupyter
> conda install -c conda-forge findspark
```

Requirements for Hands-On part

Cont.…

- Set the `JAVA_HOME` environment variable to your JDK-location.
- Add the anaconda `bin`-Directory to your `PATH` variable in the `.bashrc` file. Finally it should look like this:

```
export JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"
export SPARK_HOME=~/tutorial/spark-2.3.1-bin-hadoop2.7
export PATH=~/anaconda3/bin:$SPARK_HOME/bin:$PATH
```

- Start the `jupyter` server:

```
> jupyter notebook
```

- A browser window should open showing the content in a new window

Requirements for Hands-On part (Mac and Windows users)

For Mac-Users the installation procedure is rather similar. A good installation guideline can be found here: http://jmedium.com/pyspark-in-mac/

Windows users might have to install SCALA and and HADOOP-binary file. A good installation guideline with screenshots can be found e.g. here; https://guendouz.wordpress.com/2017/07/18/how-to-install-apache-spark-on-windows-10/

The Anaconda binaries for Mac and Windows are also available via the official Anaconda Repository (https://repo.continuum.io/archive/)

Create a new notebook and parse the following lines

Create a new notebook and parse the following lines to test the Spark-Jupyter connection.

— First prepare the environment:

```
import findspark
findspark.init()
import pyspark
from pyspark import SparkContext
print('//// Start my local Spark session ////')
```

— Get the name SparkContext and give it a Name

```
sc = SparkContext("local", "Spark connection app")
sc.setLogLevel("WARN")
```

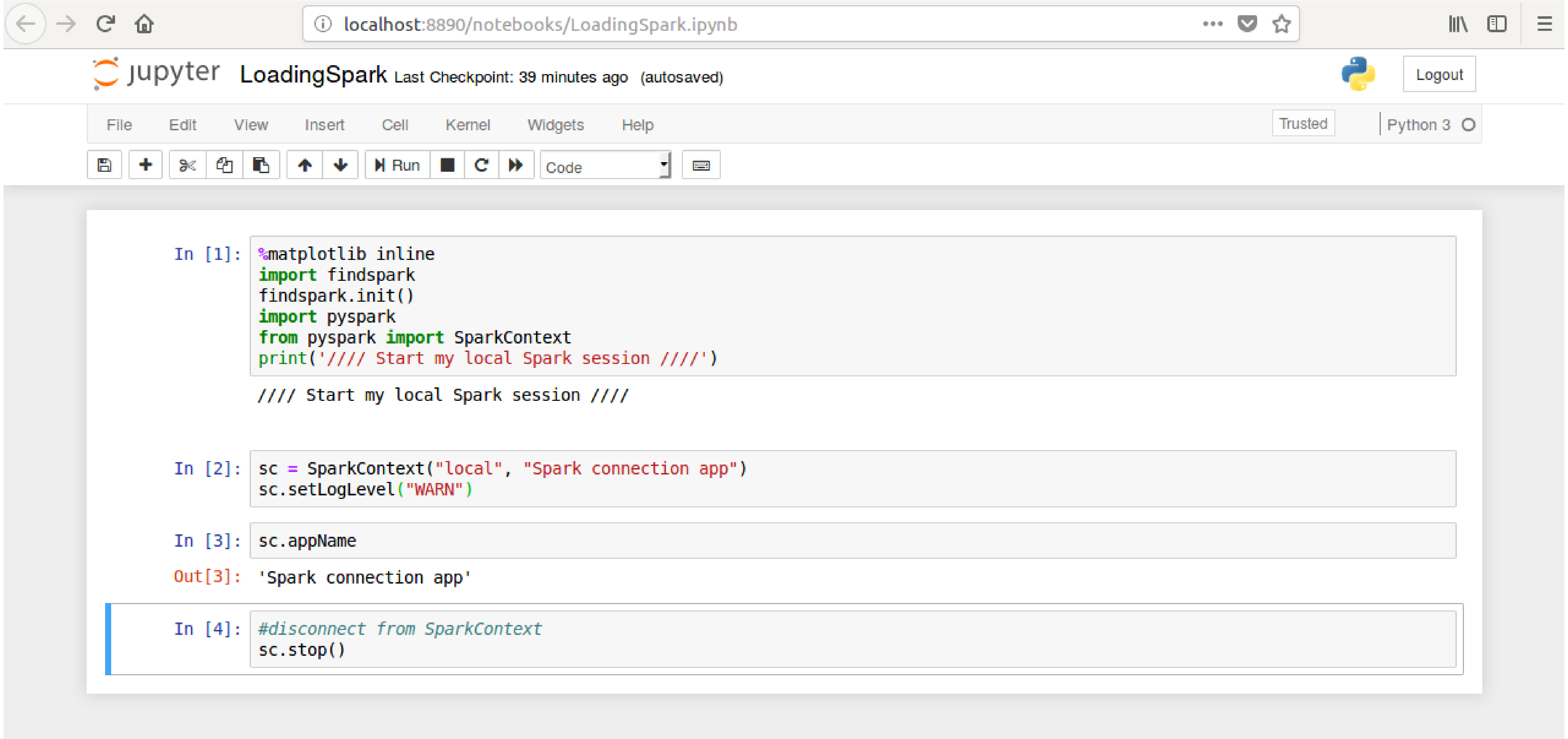
— Print the app-name again:

```
sc.appName
```

— Disconnect from SparkContext to clean up:

```
sc.stop()
```

Create a new notebook and parse the following lines


localhost:8890/notebooks/LoadingSpark.ipynb
jupyter LoadingSpark Last Checkpoint: 39 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

In [1]:

```
%matplotlib inline
import findspark
findspark.init()
import pyspark
from pyspark import SparkContext
print('//// Start my local Spark session ////')
```

```
//// Start my local Spark session ////
```

In [2]:

```
sc = SparkContext("local", "Spark connection app")
sc.setLogLevel("WARN")
```

In [3]:

```
sc.appName
```

Out[3]:

```
'Spark connection app'
```

In [4]:

```
#disconnect from SparkContext
sc.stop()
```

Content of example scripts

The aim of the Walkthrough-session is to make participants familiar with some of the basic concepts of Apache Spark and to illustrate the concept of the in-build data transformation and actions of Spark.

Under the following URL the sample data and Jupyter scripts are available as zip-file (tutorial.zip) for participants of the workshop:

https://cloudstore.zih.tu-dresden.de/index.php/s/uD8o2eIblL5adL2

Additional resources

Collection of Jupyter notebooks for data analytics

Some resources:

- Data Science and Big Data with Python, by Steve Phelps
  https://github.com/phelps-sg/python-bigdata
- A collection of IPython notebooks covering various topics
  https://github.com/jdwittenauer/ipython-notebooks
- A gallery of interesting Jupyter Notebooks
  https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks
- The big data micro benchmark suite HiBench (intel-hadoop/HiBench)
  https://github.com/intel-hadoop/HiBench