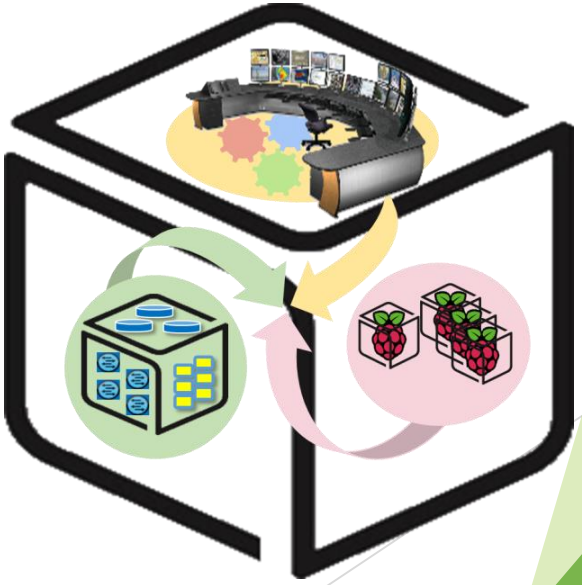
SmartX Labs
for Computer Systems

Analytics Lab

(2016, Spring)



NetCS Lab

History and Contributor of Cluster Lab (2016. 06. 29.)

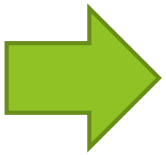
Version	Updated Date	Updated Contents	Contributor
v2r2	2015/10	(구) Analytics Lab 작성	송지원
v3	2016/05	Analytics Lab 수정	송지원
v4	2016/06/07	검수자 피드백 반영 및 내용 수정	송지원
v4r1	2016/06/08	Wordcount 예제 설명 추가	송지원
v4r2	2016/06/29	HDFS 관련 내용 수정, 예제 내용 보충	송지원

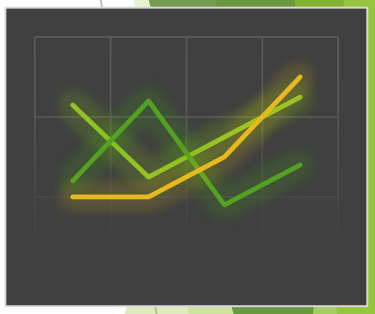
CSLab: Analytics LAB - Goal

- Data Processing with Spark & Zeppelin

Data Processing & Visualization

Data



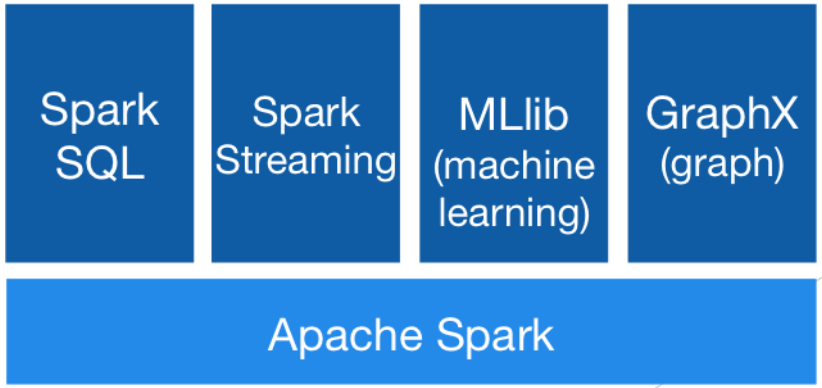


1. Background
- Apache Spark

Apache Spark™ is a fast and general engine for large-scale data processing.

- In-memory data processing framework: Fast!
- Easy to use, community fastly growing
- Libraries: SQL and DataFrame, Streaming, MLlib, GraphX
- Run on standalone or Mesos, Yarn, etc

- Scala, Java, Python



2. Background
- Spark RDD and APIs

RDD (Resilient Distributed Datasets): a distributed memory abstraction that allows programmers to perform in-memory computations on large clusters while retaining the fault tolerance of data flow models like MapReduce.

class pyspark.RDD

map()
groupBy(), groupByKey()
reduce(), reduceByKey()
join()
sort(), sortByKey()
union()
…

http://spark.apache.org/docs/latest/api.html

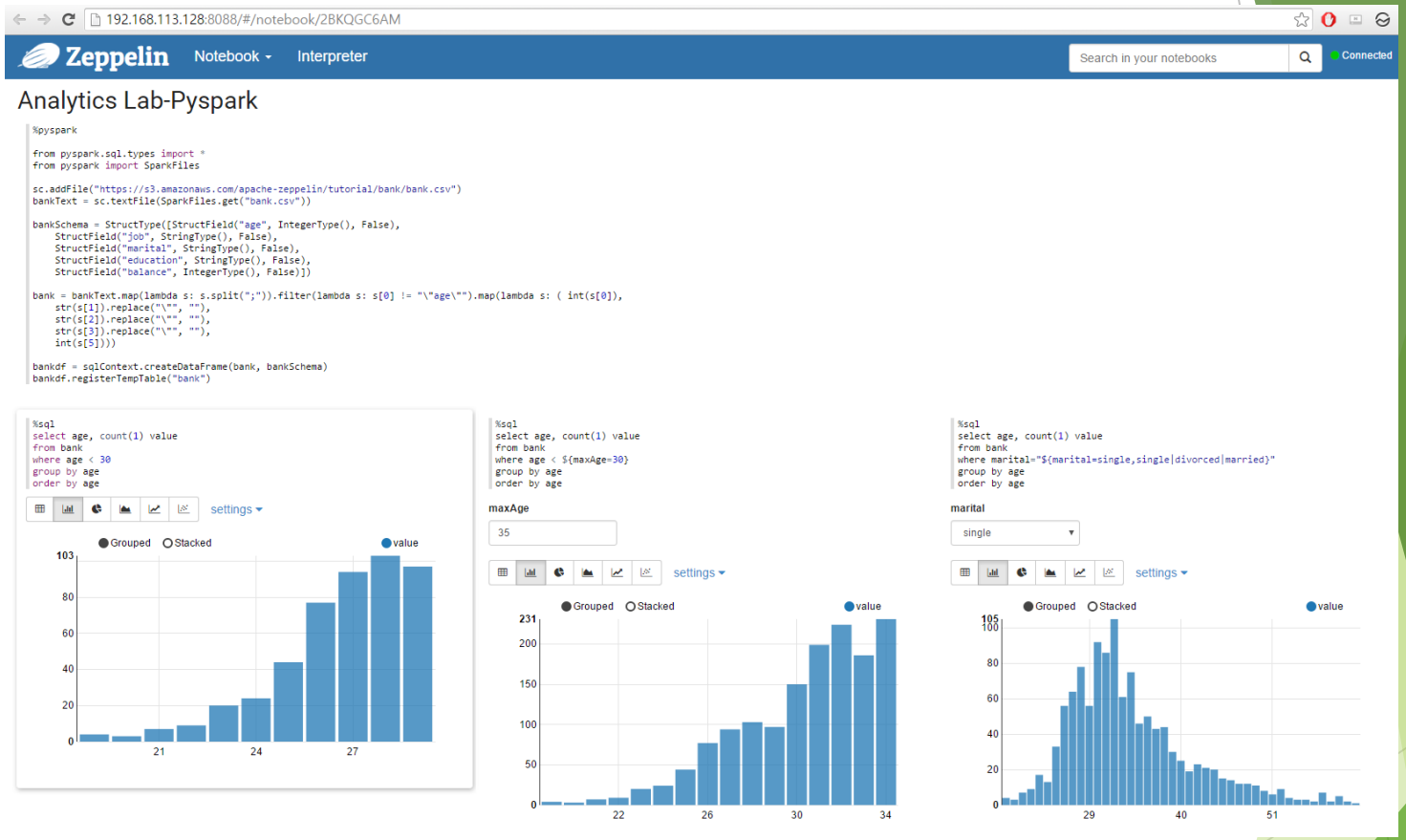
2. Background
- Preparation

- If you installed HDFS, log in to ‘hadoop.’
- Run Zeppelin daemon.

(See Cluster Lab)

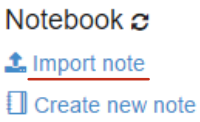
2. Background

- Pyspark: Spark in Python Language

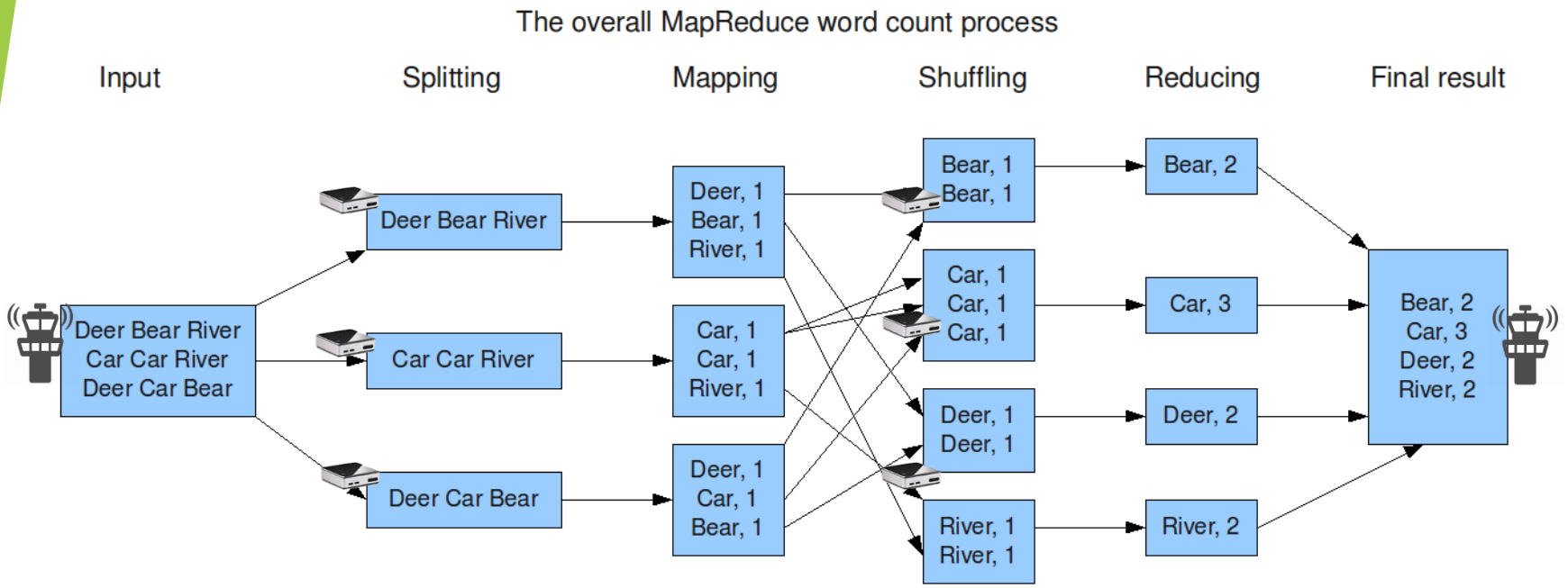


Zeppelin tutorial converted to pyspark (https://github.com/SmartX-Labs/Mini/blob/master/Lab-7.%20Analytics/Analytics_Lab-Pyspark.json)

- In Import note and run.



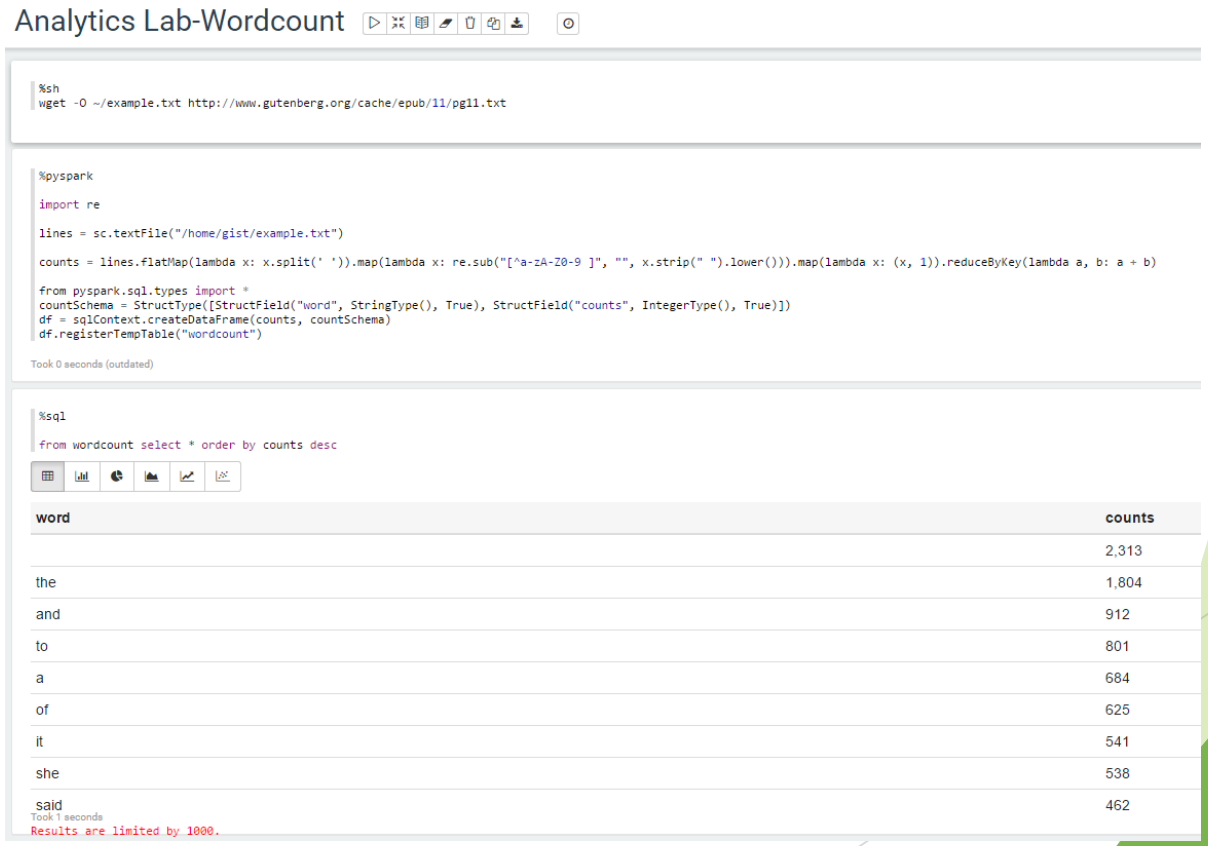
3. Apache Zeppelin
- Wordcount



3. Apache Zeppelin
- Wordcount – without HDFS

Download and import notebook:
https://github.com/SmartX-Labs/Mini/blob/master/Lab-7.%20Analytics/Analytics_Lab-Wordcount.json

Change path of example.txt



3. Apache Zeppelin
- Wordcount – with HDFS

Upload `example.txt` to HDFS.

- `hadoop fs –put example.txt /user/`

If you're not logged in as 'hadoop', instead try this.

- `sudo –u hadoop /usr/local/hadoop/bin/hadoop fs –put example.txt /user/`

Change %pyspark paragraph like this.

```
%pyspark

from pyspark import SparkFiles
import re

lines = sc.textFile("hdfs://nuc06:9000/user/example.txt")   # Change hostname or IP.

counts = lines.flatMap(lambda l: l.split(' ')) \
        .map(lambda x: re.sub("[^a-zA-Z0-9 ]", "", x.lower())).filter(lambda x: x != '') \
        .map(lambda x: (x, 1)) \
        .reduceByKey(lambda a, b: a + b)

#print(counts.collect())

from pyspark.sql.types import *
countSchema = StructType([StructField("word", StringType(), True), StructField("counts", IntegerType(), True)])
df = sqlContext.createDataFrame(counts, countSchema)
df.registerTempTable("wordcount")
```

`lines = sc.textFile("hdfs://<TOWER_IP>:9000/user/example.txt")`

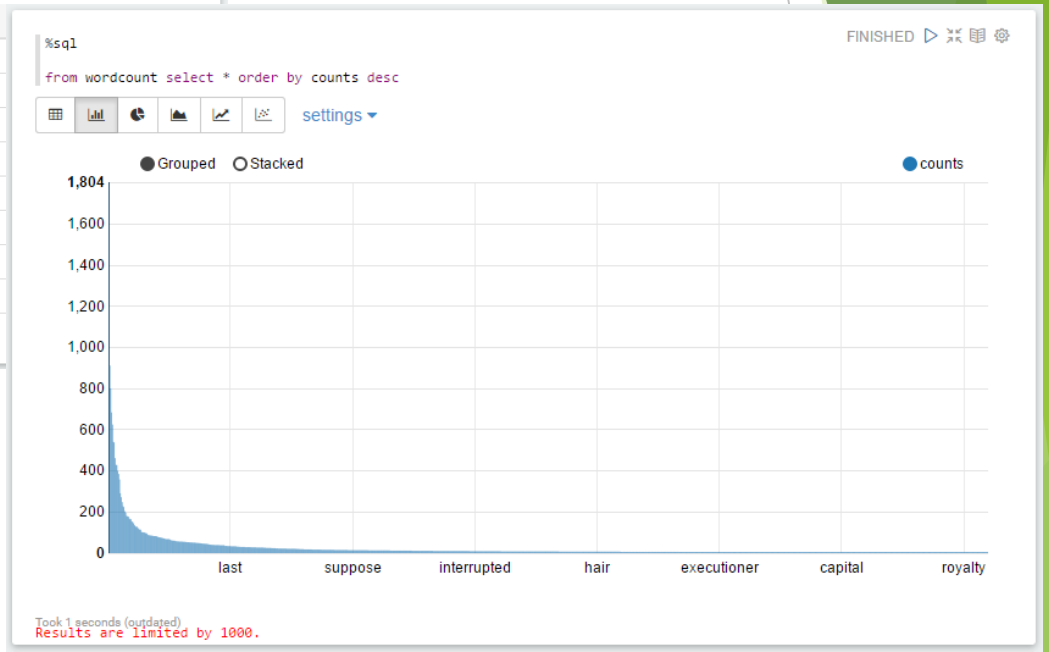
3. Apache Zeppelin
- Wordcount (Result-Visualization)

```
%sql
from wordcount select * order by counts desc
```

FINISHED

word	counts
the	1,804
and	912
to	801
a	684
of	625
it	541
she	538
said	462
you	429

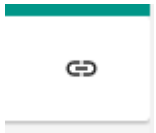
Took 1 seconds
Results are limited by 1000.



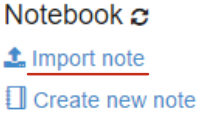
3. Apache Zeppelin
- Explore Zeppelin notebooks

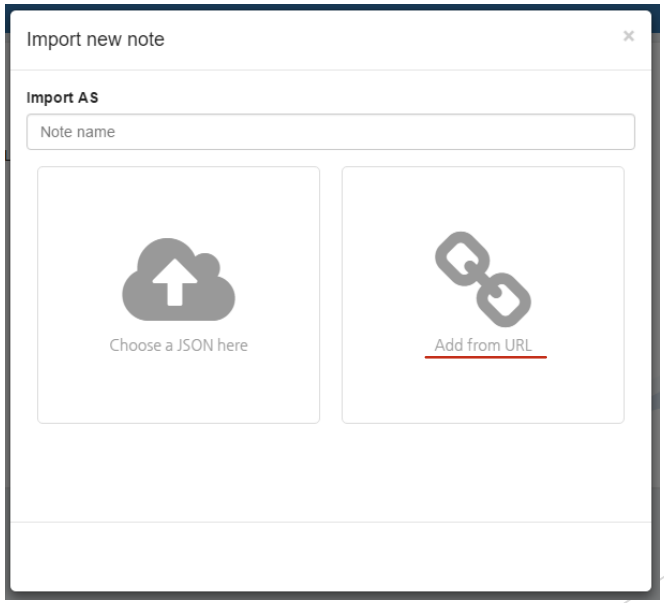
https://www.zeppelinhub.com/viewer

Look around the interesting notebooks.
And try to run on your machine by importing a notebook.



Copy url





Thank You for Your Attention Any Questions?