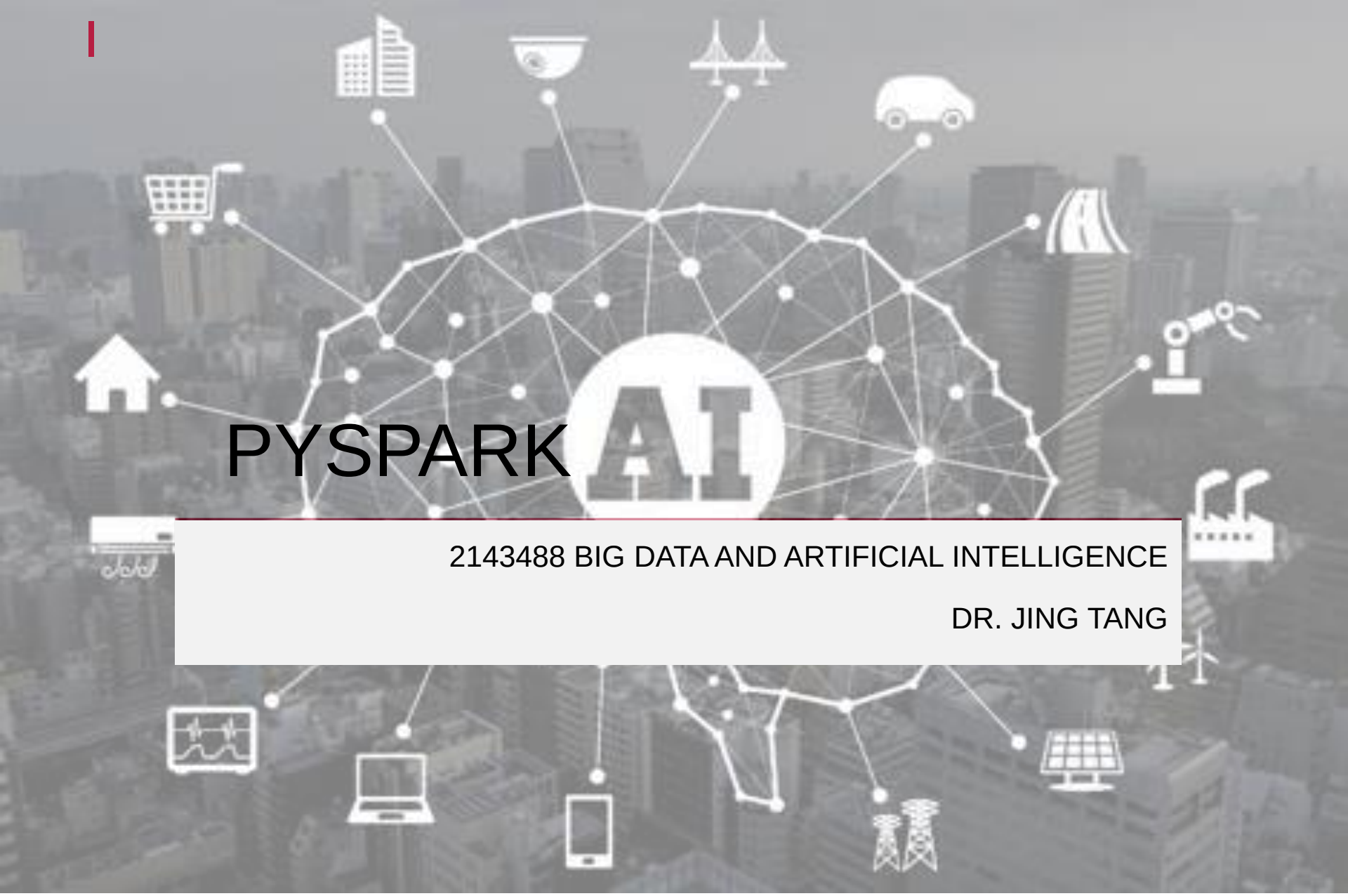


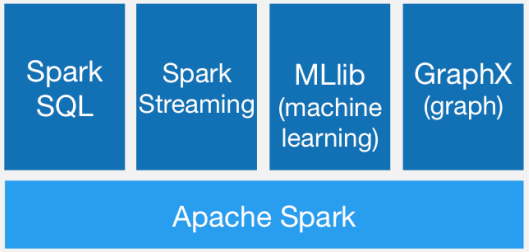
PYSPARK

2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

DR. JING TANG

APACHE SPARK

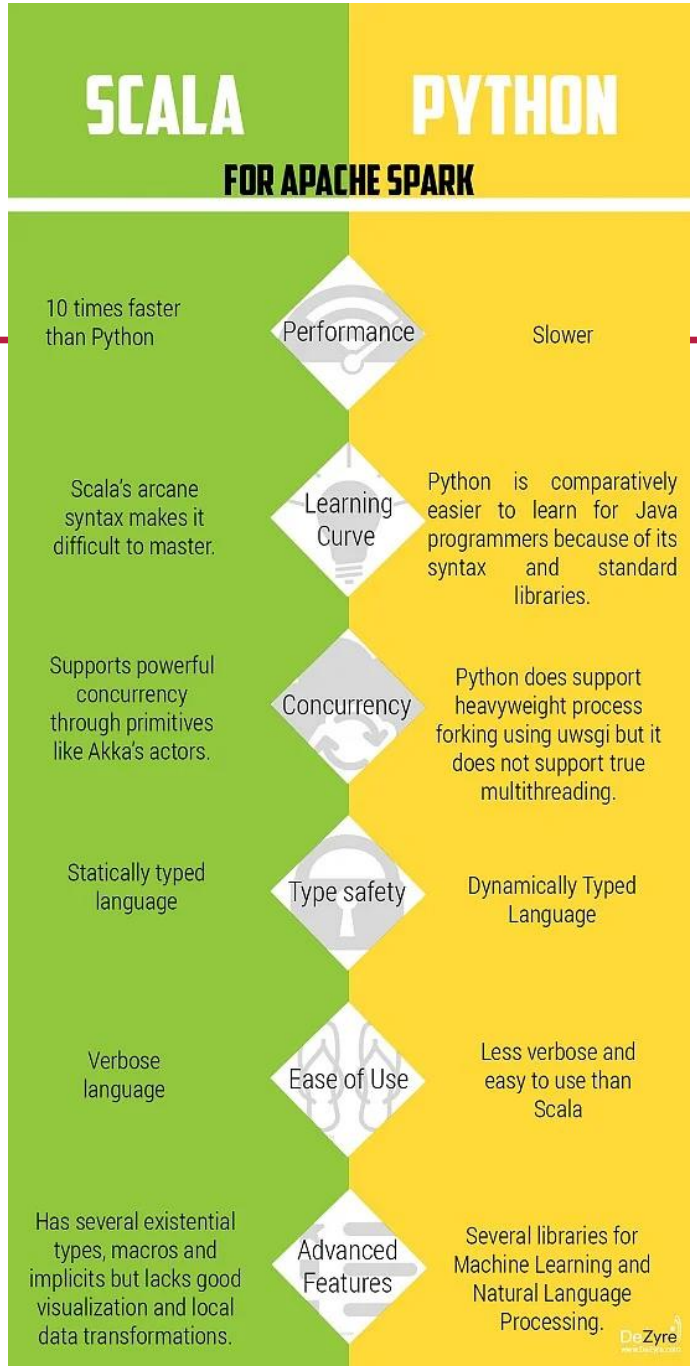
- Apache Spark™ is a unified analytics engine for large-scale data (**massive distributed processing**)
  - Speed: Run workloads 100x faster than Hadoop
  - Ease of Use: Write applications quickly in Java, Scala, Python, R, and SQL.
  - Generality: Combine SQL, streaming, and complex analytics.



  - Runs Everywhere: Run on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud. It can access diverse data sources.

PYSPARK

- While using Spark, most data engineers recommends to develop either in **Scala** (which is the “native” Spark language) or in Python through complete **PySpark API**.
  - “Scala is faster and moderately easy to use, while Python is slower but very easy to use.”

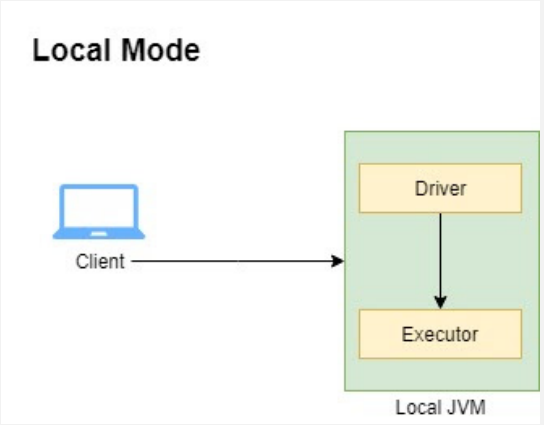


INSTALLATION (ANYONE OF FOLLOWING)

- https://www.knowledgehut.com/blog/big-data/how-to-install-apache-spark-on-windows
- https://phoenixnap.com/kb/install-spark-on-windows-10
- https://www.youtube.com/watch?v=WQErwxRTiW0

RUN APACHE SPARK LOCALLY

- Local mode： in which spark can be run locally without any cluster requirement.
  - This mode is suitable for scenarios when we do not have enough resources to create cluster.
  - But in this mode, you get only one executor and bot Driver, and Executer runs in the same JVM.




```
Spark context Web UI available at http://DESKTOP-41K2EH1:4043
Spark context available as 'sc' (master = local[*], app id = local-1619106270325).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.1
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_291)
Type in expressions to have them evaluated.
Type :help for more information.
```

FIRST PYSPARK

In [1]:
```
from pyspark.sql import Row
a=Row(name='Antony', age=27, height=168)
print("a:",a)
```

```
a: Row(name='Antony', age=27, height=168)
```

In [2]:
```
print(a.name)
```

```
Antony
```

In [5]:
```
Person=Row('Name','Age','Height')
```

In [6]:
```
a=Person('apple',21,168)
b=Person('pear',29,177)
print(a)
print(b)
```

```
Row(Name='apple', Age=21, Height=168)
Row(Name='pear', Age=29, Height=177)
```

FIRST PYSPARK (CONT.)

In [7]:
```
from pyspark import SparkContext, SparkConf
sc=SparkContext.getOrCreate()
sc
```

Out[7]: **SparkContext**

Spark UI

Version

`v3.1.1`

Master

`local[*]`

AppName

`PySparkShell`

In [8]:
```
df=sc.parallelize([a,b]).toDF()
df.show()
```

```
+-----+---+------+
| Name|Age|Height|
+-----+---+------+
|apple| 21|   168|
| pear| 29|   177|
+-----+---+------+
```

FIRST PYSPARK (CONT.)

In [8]:
```
df=sc.parallelize([a,b]).toDF()
df.show()
```

```
+-----+---+------+
| Name|Age|Height|
+-----+---+------+
|apple| 21|   168|
| pear| 29|   177|
+-----+---+------+
```

In [9]:
```
df.printSchema()
```

```
root
 |-- Name: string (nullable = true)
 |-- Age: long (nullable = true)
 |-- Height: long (nullable = true)
```

In [11]:
```
columns = ["name","age","tall"]
df=df.toDF(*columns)
df.show()
```

```
+-----+---+----+
| name|age|tall|
+-----+---+----+
|apple| 21| 168|
| pear| 29| 177|
+-----+---+----+
```

SPARK WEB UI MONITORING

Spark 3.1.1 | Jobs | Stages | Storage | Environment | Executors | SQL | PySparkShell application UI

Executors

Show Additional Metrics

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	0	0.0 B / 366.3 MiB	0.0 B	4	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	0.0 B / 366.3 MiB	0.0 B	4	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0

Executors

Show 20 entries

Search:

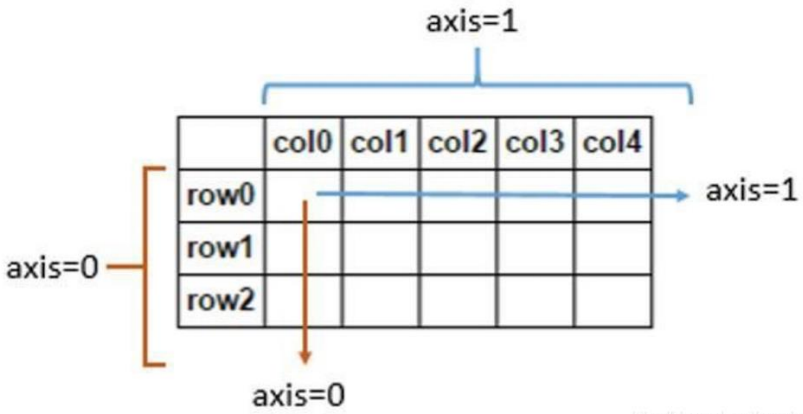
Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Thread Dump
driver	DESKTOP-41K2EH1:60868	Active	0	0.0 B / 366.3 MiB	0.0 B	4	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	Thread Dump

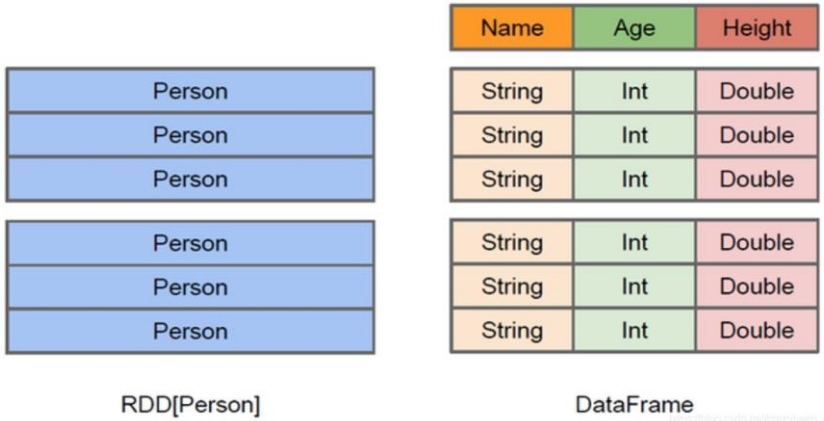
Showing 1 to 1 of 1 entries

Previous 1 Next

PYSPARK VS PANDAS DATAFRAME

PANDAS	PYSPARK



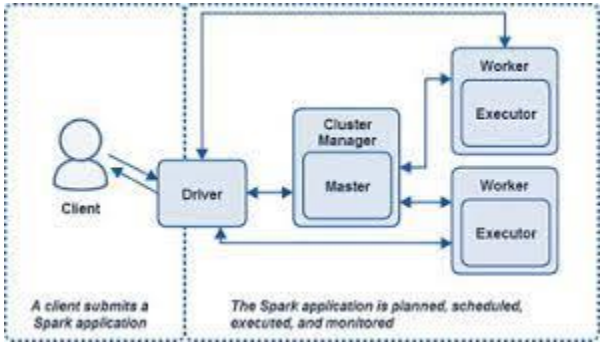


4 time faster!

OTHER EXAMPLES

- Pyspark_2: "Introduction to Spark DataFrame“
- Pyspark_3: "data preprocessing in Spark“
- Pyspark_4: "Regression in Spark"

MASTER AND WORKER



- The components of a Spark application:
  - Driver: plans and coordinates the set of tasks required to run a Spark application.
    - SparkContext, SparkSession (new entry point)
  - Master: the process that requests resources in the cluster and makes them available to the Spark Driver.
  - Cluster Manager: the process responsible for monitoring the Worker nodes and reserving resources on these nodes upon request by the Master.
  - The tasks themselves run in Executors, which are hosted on Worker nodes.

BUILD CLUSTER COMPUTER APACHE SPARK IN WINDOWS

- Run in your master machine :
  - ./bin/spark-class org.apache.spark.deploy.master.Master
  - It will give us IP:PORT
  - Web UI: http://localhost:8080/
- Run in your workers machine:
  - ./bin/spark-class org.apache.spark.deploy.worker.Worker spark://IP:PORT
- To create and deploy Spark code application, type : "pyspark --master spark://IP:PORT" in your master cluster.

MASTER

Anaconda Prompt (anaconda3) - spark-class org.apache.spark.deploy.master.Master

```
(base) C:\Users\Jing>cd %SPARK_HOME%\bin

(base) C:\Users\Jing\Spark\bin>spark-class org.apache.spark.deploy.master.Master
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
21/04/22 16:21:20 INFO Master: Started daemon with process name: 14604@DESKTOP-41K2EH1
21/04/22 16:21:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
21/04/22 16:21:21 INFO SecurityManager: Changing view acls to: Jing
21/04/22 16:21:21 INFO SecurityManager: Changing modify acls to: Jing
21/04/22 16:21:21 INFO SecurityManager: Changing view acls groups to:
21/04/22 16:21:21 INFO SecurityManager: Changing modify acls groups to:
21/04/22 16:21:21 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view per
missions: Set(Jing); groups with view permissions: Set(); users  with modify permissions: Set(Jing); groups with modify
permissions: Set()
21/04/22 16:21:23 INFO Utils: Successfully started service 'sparkMaster' on port 7077.
21/04/22 16:21:23 INFO Master: Starting Spark master at spark://192.168.0.29:7077
21/04/22 16:21:23 INFO Master: Running Spark version 3.1.1
21/04/22 16:21:24 INFO Utils: Successfully started service 'MasterUI' on port 8080.
21/04/22 16:21:24 INFO MasterWebUI: Bound MasterWebUI to 0.0.0.0, and started at http://DESKTOP-41K2EH1:8080
21/04/22 16:21:24 INFO Master: I have been elected leader! New state: ALIVE
```