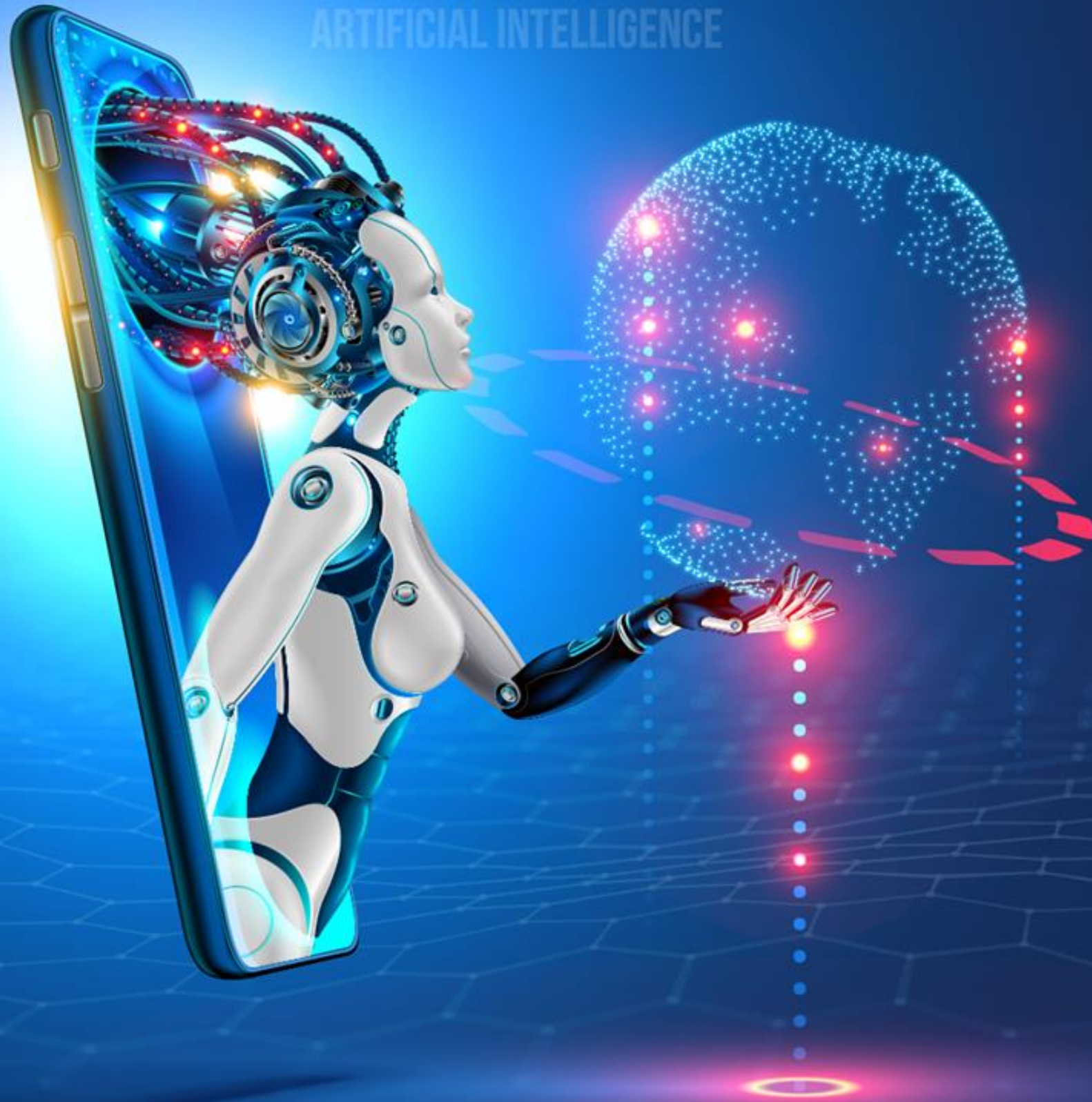


# DATA AND ARTIFICIAL INTELLIGENCE



## Data Science with Python

# DATA AND ARTIFICIAL INTELLIGENCE



## Python Integration with Hadoop MapReduce and Spark



## Learning Objectives

By the end of this lesson, you will be able to:

- 🕒 Explain why Python should be integrated with Hadoop
- 🕒 Outline the ecosystem and architecture of Hadoop
- 🕒 Explain the functioning of MapReduce
- 🕒 Discuss Apache Spark functions and their benefits
- 🕒 Write Python programs for Hadoop operations



# Quick Recap: Need for Real-Time Analytics

We have seen how big data is generated and understood that to extract insights, proper analysis of data is more important than its size.



# Quick Recap: Need for Real-Time Analytics

Real-time analytics is the rage right now because it helps extract information from different data sources almost instantly.

| Date  | Description                 | Deposit | Withdrawal | Balance   |
|-------|-----------------------------|---------|------------|-----------|
| Apr 1 | ATM Post Debit              |         | 100        | \$200,000 |
| Apr 2 | Paypal Transfer 231054      | 200     |            | \$202,000 |
| Apr 3 | Simplilearn course fee      |         | 150        | \$200,500 |
| Apr 4 | Starluck Café               |         | 210        | \$198,400 |
| Apr 5 | Walcart TX                  |         | 230        | \$196,100 |
| Apr 6 | ebuy swiss watch 239        |         | 250        | \$193,600 |
| Apr 7 | Caterpallor black boots men |         | 270        | \$190,900 |
| Apr 8 | Halo blue shirt 831         |         | 160        | \$189,300 |

Information source;  
overall patterns not  
clearly visible

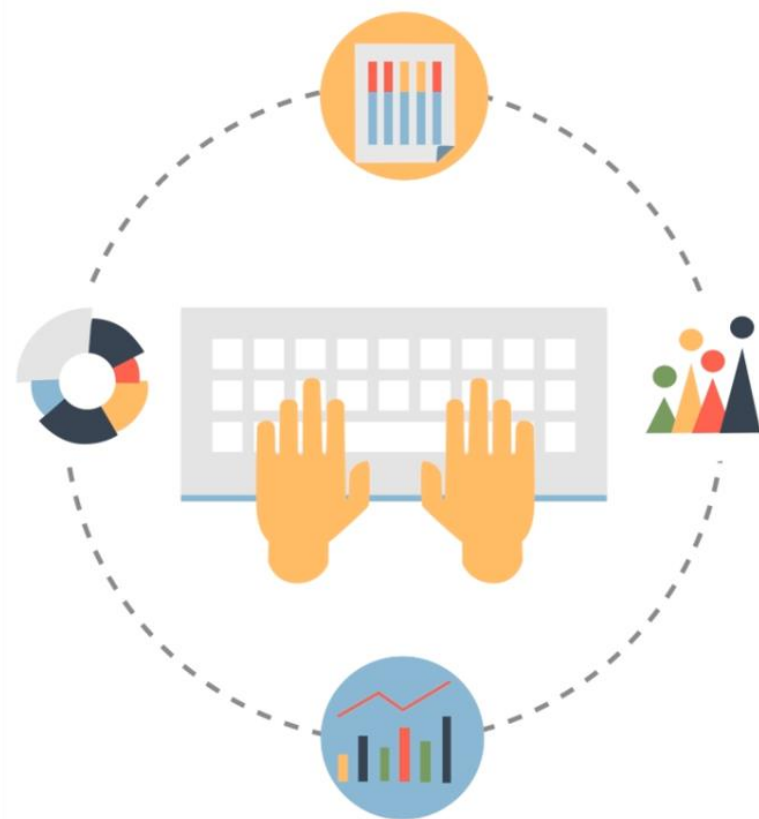


# Quick Recap: Need for Real-Time Analytics



# Quick Recap: Why Python

Data Scientists all over the world prefer Python because it is an easy-to-use language that has diverse libraries required for analytics.



Data Science



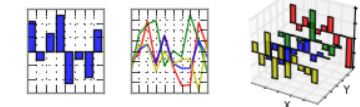
Acquire



Wrangle

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Explore



Model



Visualize



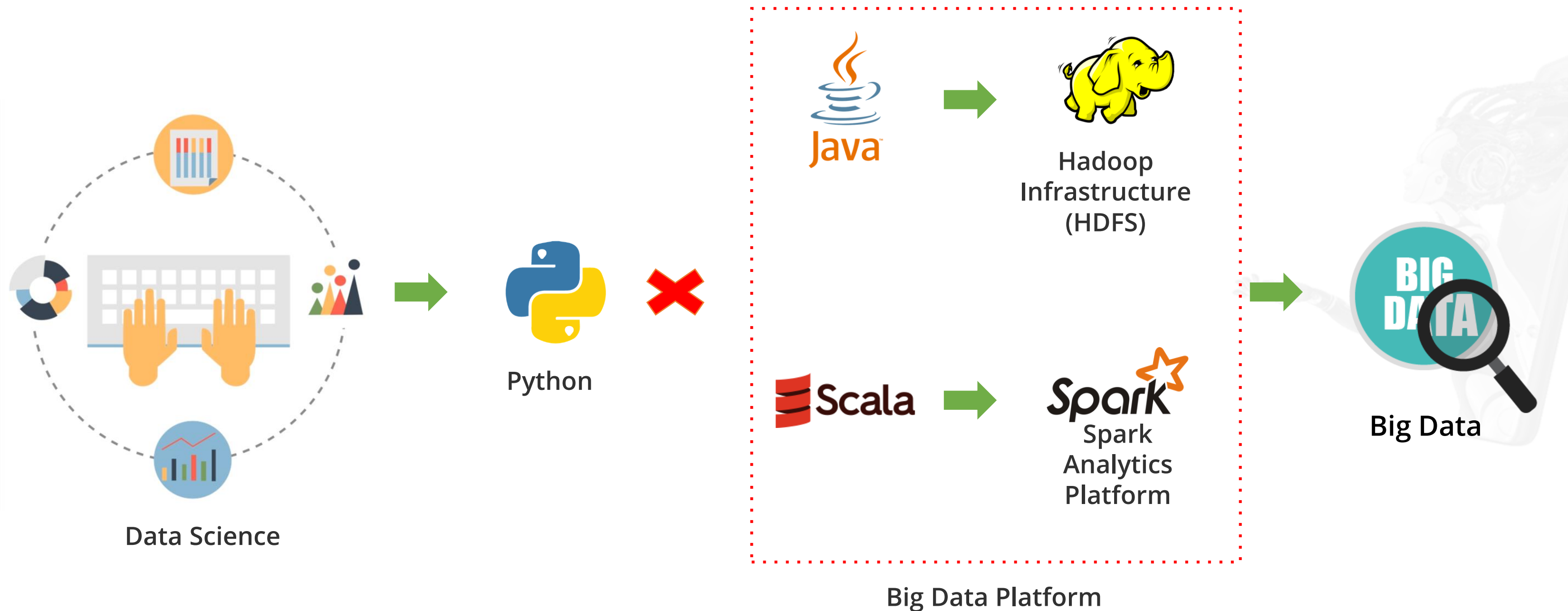
Bokeh





# Disparity in Programming Languages

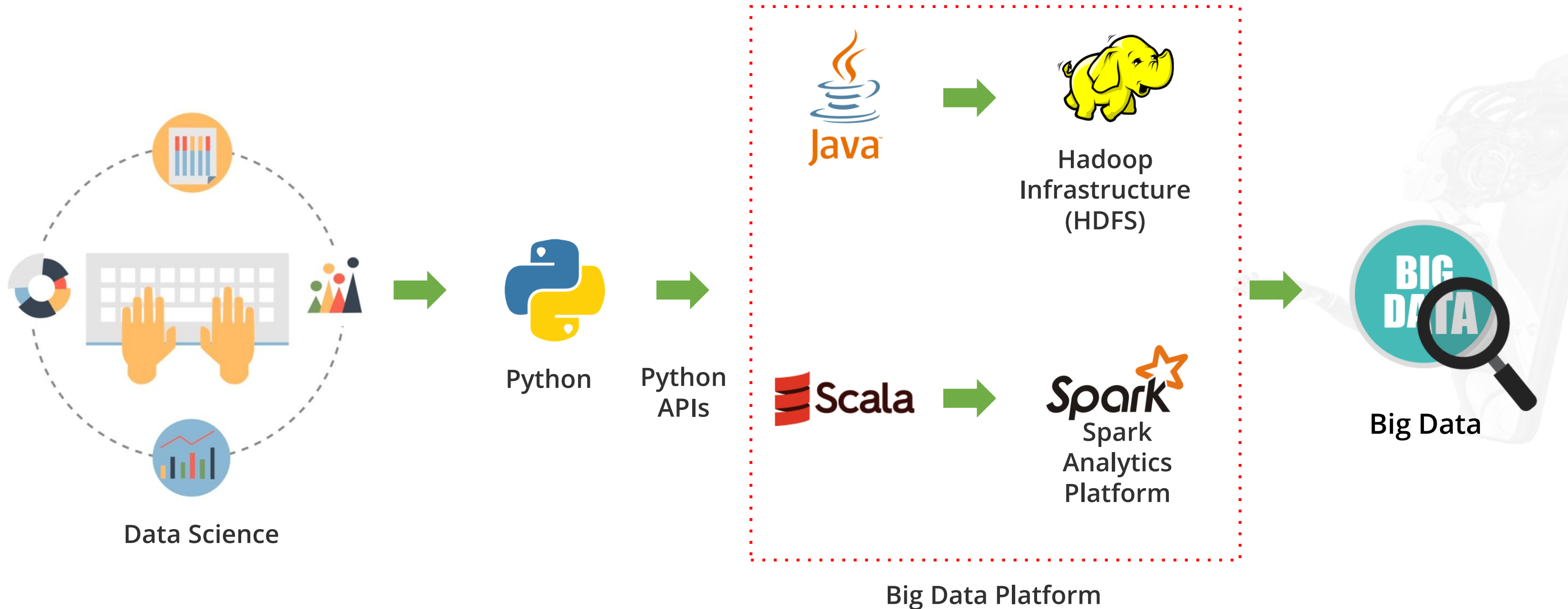
However, Big Data can only be accessed through Hadoop which is completely developed and implemented in Java. Also, analytics platforms are coded in different programming languages.





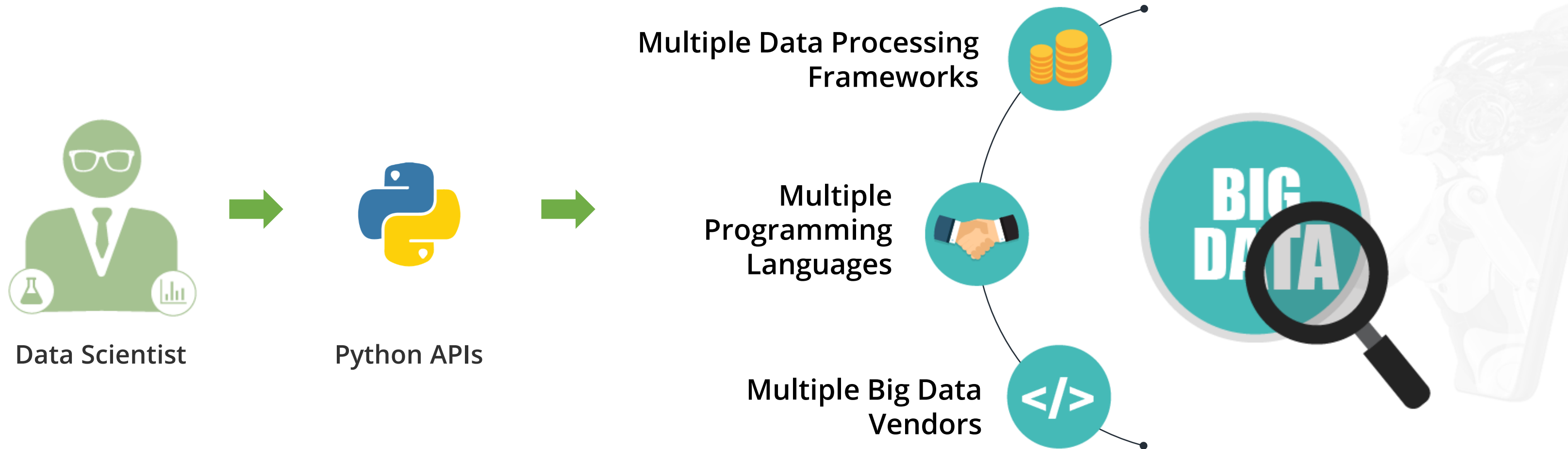
# Integrating Python with Hadoop

As Python is a Data Scientist's first language of choice, both Hadoop and Spark provide Python APIs that allow easy access to the Big Data platform.

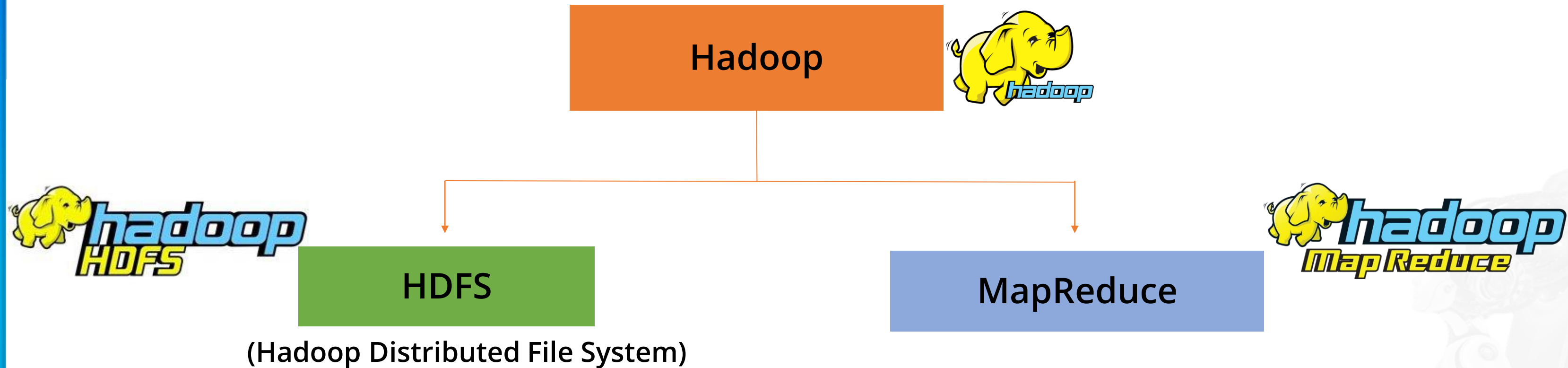


# Need for Big Data Solutions in Python

There are several reasons for creating Big Data solutions in Python.



# Hadoop: Core Components

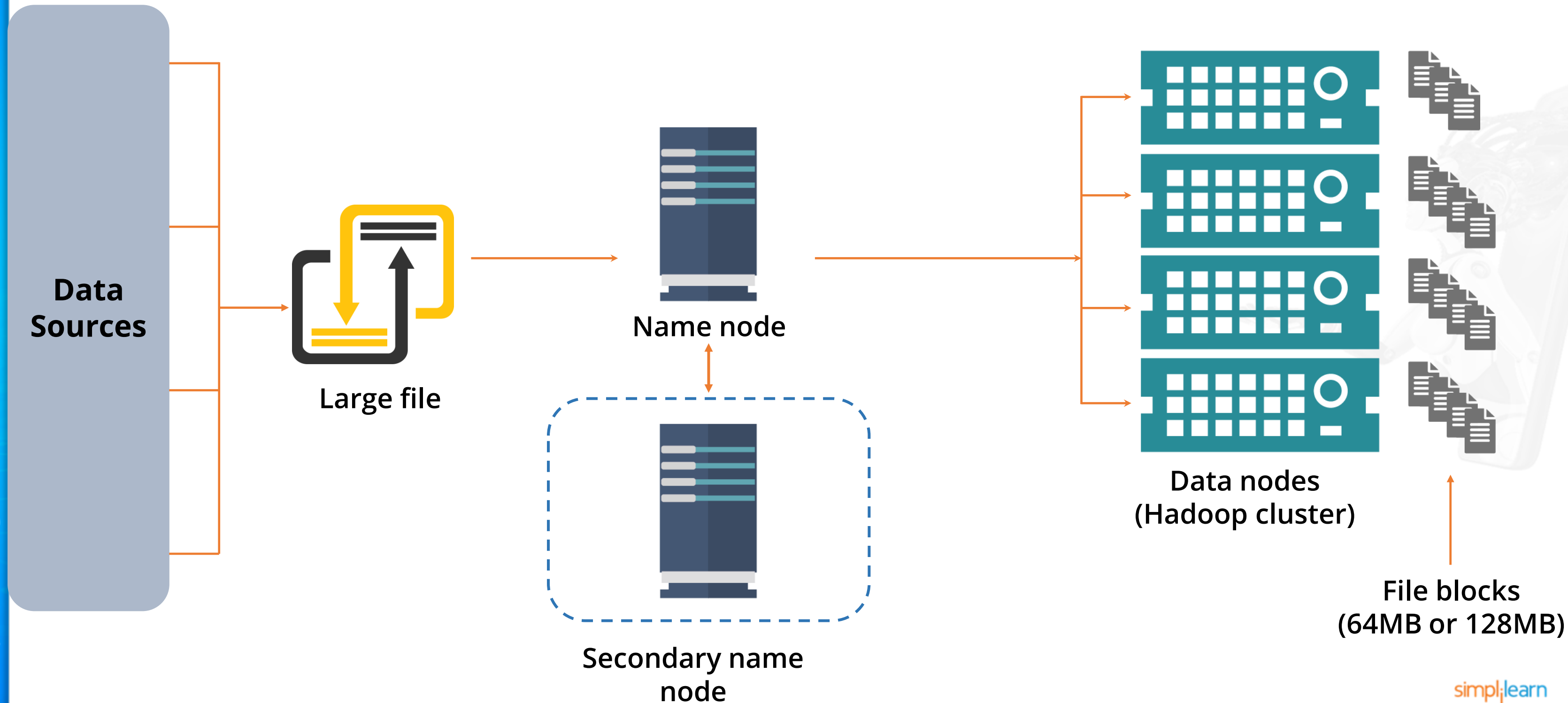


- It is responsible for storing data on a cluster
- Data is split into blocks and distributed across multiple nodes in a cluster
- Each block is replicated multiple times
  - Default is 3 times
  - Replicas are stored on different nodes

- MapReduce is a data processing framework to process data on the cluster
- Two consecutive phases: Map and Reduce
- Each map task operates on discrete portions of data
- After map, reduce works on the intermediate data distributed on nodes

# Hadoop: System Architecture

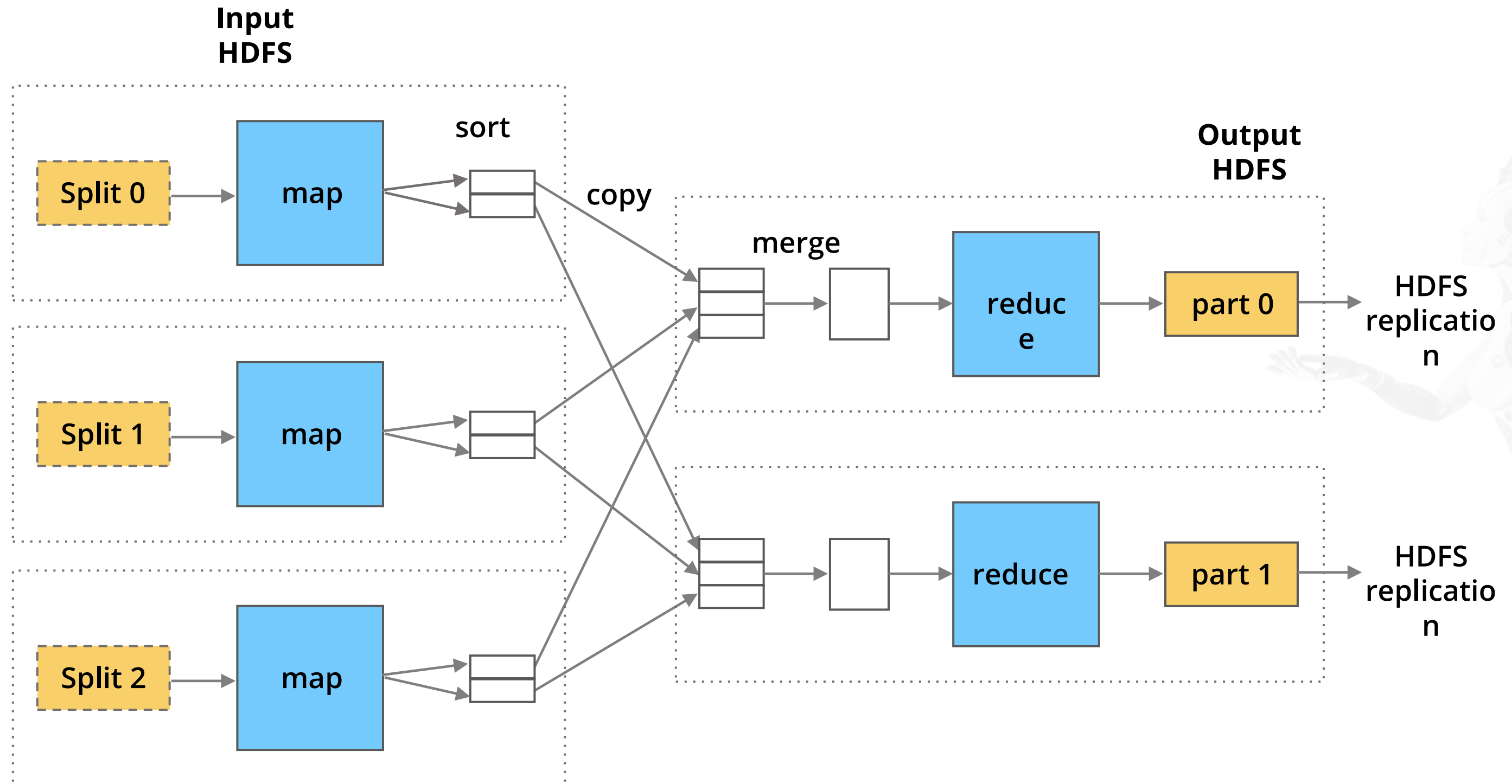
This example illustrates the Hadoop system architecture and the ways to store data in a cluster.





# MapReduce

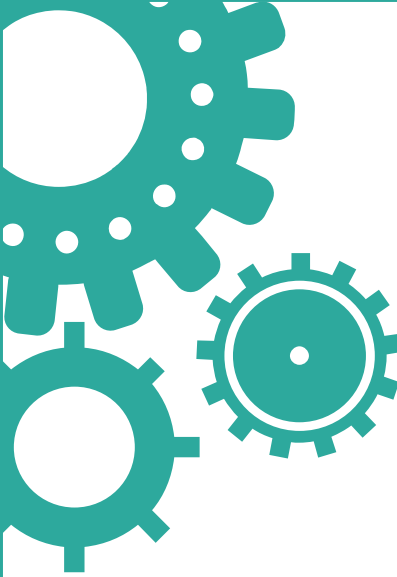
The second core component of Hadoop is MapReduce, the primary framework of the HDFS architecture.




# MapReduce: Mapper and Reducer

Let us discuss the MapReduce functions, mapper and reducer, in detail.

## Mapper

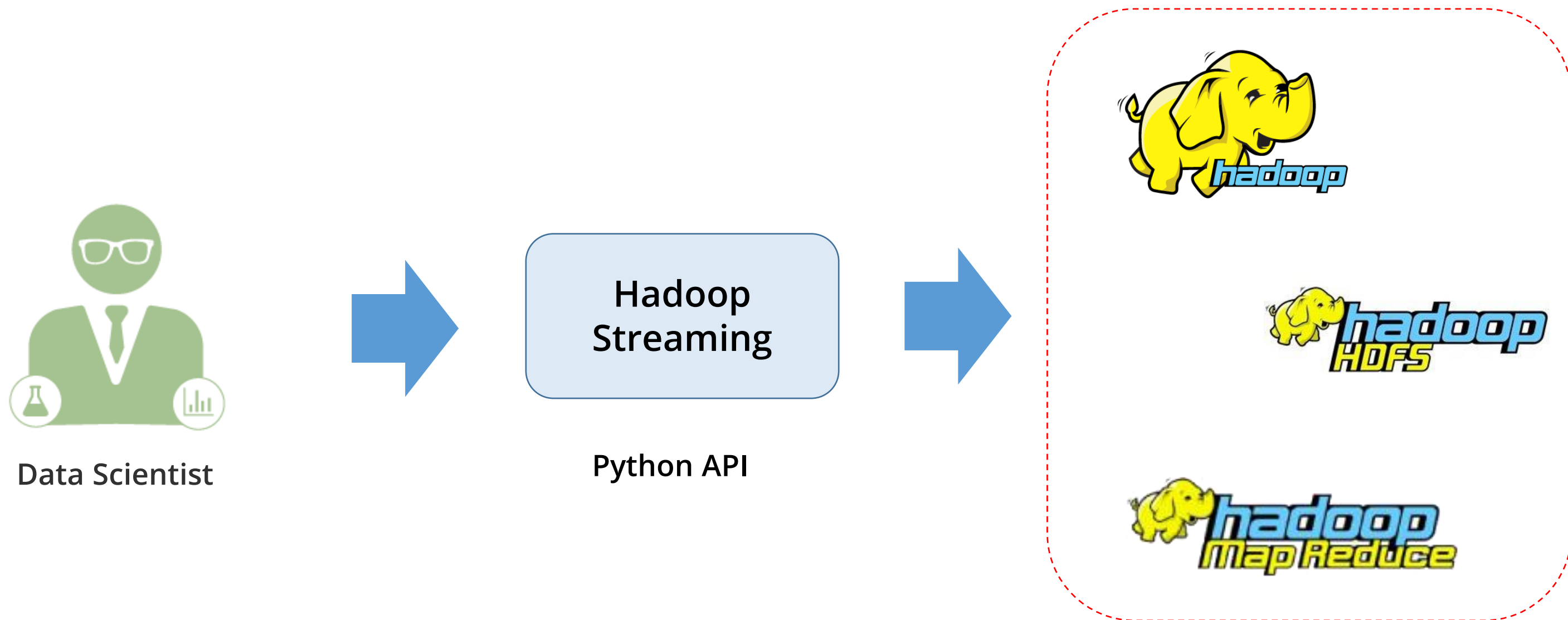
- 
- Mappers run locally on the data nodes to avoid the network traffic.
  - Multiple mappers run in parallel processing a portion of the input data.
  - The mapper reads data in the form of key-value pairs.
  - If the mapper write generates an output, it is written in the form of key-value pairs.

## Reducer

- 
- All intermediate values for a given intermediate key are combined together into a list and given to a reducer.
  - This step is known as **shuffle** and **sort**.
  - The reducer outputs either zero or more final key-value pairs. These are written to HDFS.

# Hadoop Streaming: Python API for Hadoop

Hadoop Streaming acts like a bridge between your Python code and the Java-based HDFS, and lets you seamlessly access Hadoop clusters and execute MapReduce tasks.



# Mapper in Python

Python supports map and reduce operations:

Suppose you have list of numbers you want to square = [1, 2, 3, 4, 5, 6]

Square function is written as follows:

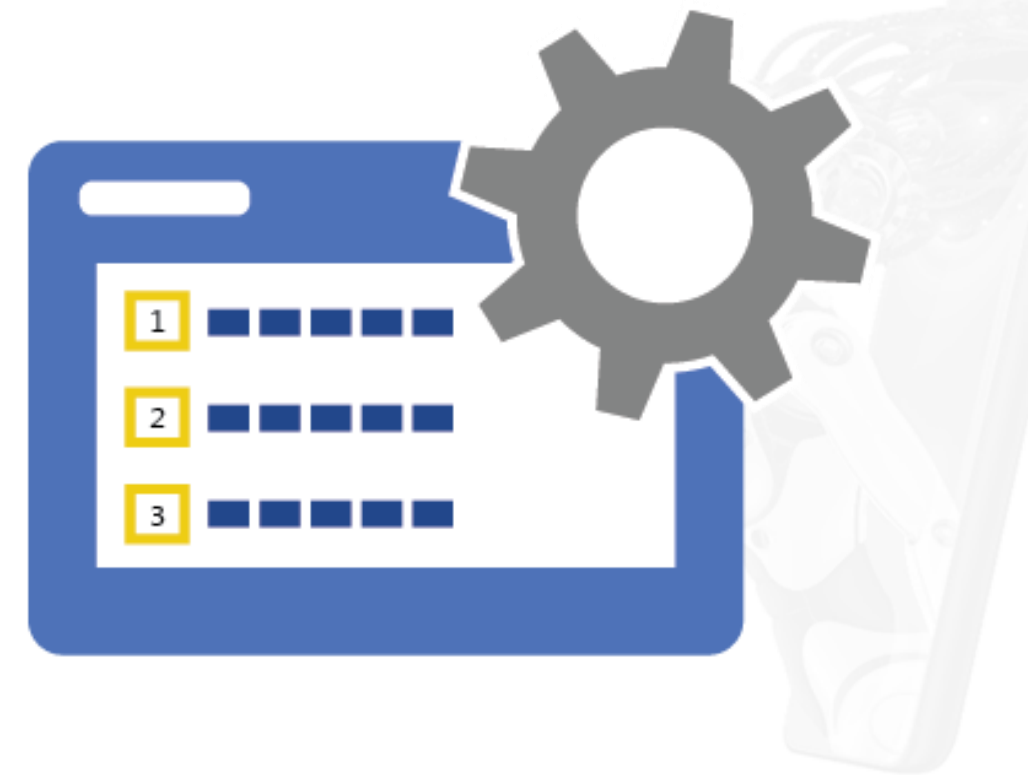
```
def square(num):  
    return num * num
```

You can square this list using the following code:

```
squared_nums = map(square, numbers)
```

Output would be:

```
[1, 4, 9, 16, 25, 36]
```





# Reducer in Python

Suppose you want to sum the squared numbers:  
[1, 4, 9, 16, 25, 36]

Use the **sum** function to add two numbers  

```
def sum(a, b):  
    return a + b
```

You can now sum the numbers using the **reduce** function  
Import functools as f  

```
sum_squared = f.reduce(sum, a)
```

Output would be:  
[91]



# Setting Up Cloudera QuickStart Virtual Machine

Cloudera provides enterprise-ready Hadoop Big Data platform which supports Python as well.  
To set up the Cloudera Hadoop environment, visit the Cloudera link:

[http://www.cloudera.com/downloads/quickstart\\_vms/5-7.html](http://www.cloudera.com/downloads/quickstart_vms/5-7.html)

A banner for Cloudera QuickStart Downloads for CDH 5.5. The left side features a background image of a person wearing glasses and a dark sweater, with text overlay. The right side is a dark sidebar with white and blue buttons and dropdown menus.

**QuickStart Downloads for CDH 5.5**  
Easy-to-deploy Apache Hadoop clusters for easy learning!

Cloudera QuickStart downloads contain complete Apache Hadoop clusters in the form of VMs or Docker images, including Cloudera Manager to manage them.

Cloudera QuickStart downloads are for personal and demo purposes only, and are not to be used as a starting point for production clusters.

Get Started

QUICKSTART DOWNLOADS FOR CDH 5.5 ▾

VMWARE ▾

DOWNLOAD NOW 

Cloudera recommends that you use 7-Zip to extract these files. To download and install it, visit the link:  
<http://www.7-zip.org/>

# Cloudera QuickStart VM: Prerequisites

- **These 64-bit VMs** require a 64-bit host OS and a virtualization product that can support a 64-bit guest OS.
- To use a VMware VM, you must use a player compatible with WorkStation 8.x or higher:
  - Player 4.x or higher
  - Fusion 4.x or higher
- Older versions of WorkStation can be used to create a new VM using the same virtual disk (VMDK file), but some features in VMware tools are not available.
- The amount of RAM required varies by the run-time option you choose

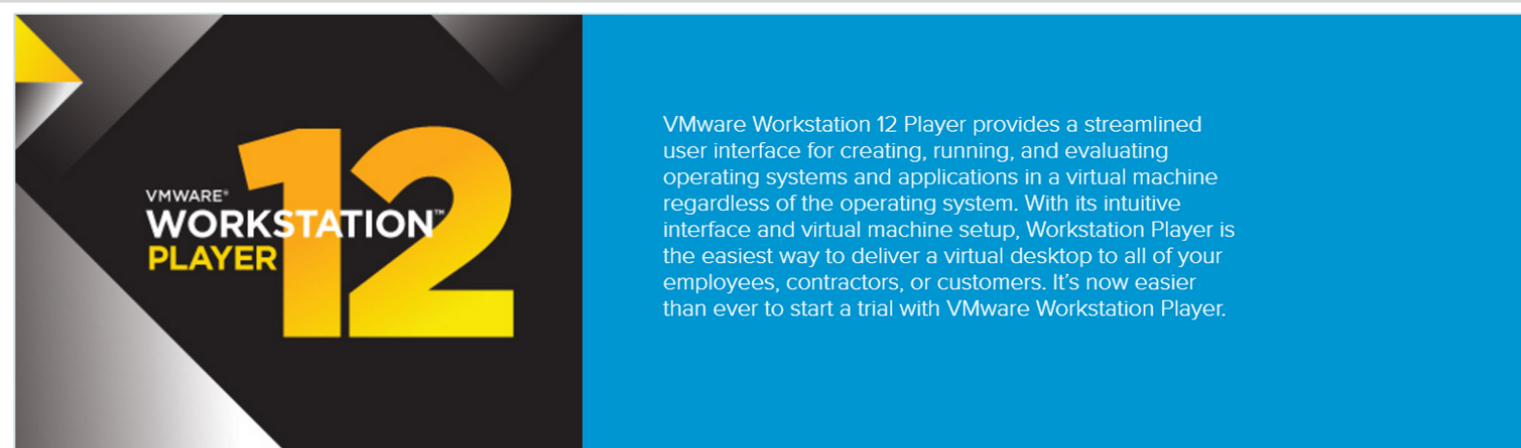
| CDH and Cloudera Manager Version | RAM Required by VM |
|----------------------------------|--------------------|
| CDH 5 (default)                  | 4+ GiB*            |
| Cloudera Express                 | 8+ GiB*            |
| Cloudera Enterprise (trial)      | 10+ GiB*           |

# Launching VMware Image

To launch the VMware, visit the VMware link:

<https://www.vmware.com/products/player/playerpro-evaluation.html>

<https://www.vmware.com/products/fusion/fusion-evaluation.html>



VMware Workstation 12 Player provides a streamlined user interface for creating, running, and evaluating operating systems and applications in a virtual machine regardless of the operating system. With its intuitive interface and virtual machine setup, Workstation Player is the easiest way to deliver a virtual desktop to all of your employees, contractors, or customers. It's now easier than ever to start a trial with VMware Workstation Player.

VMware Workstation 12 Player for Windows 64-bit

Download Now

VMware Workstation 12 Player for Linux 64-bit

Download Now



**VMware Fusion 8** is the easiest, fastest and most reliable way to run Windows applications on a Mac without rebooting.

**VMware Fusion 8 Pro** takes virtualization on the Mac to the next level with powerful features designed for advanced users and technical professionals.

VMware Fusion 8

Download Now

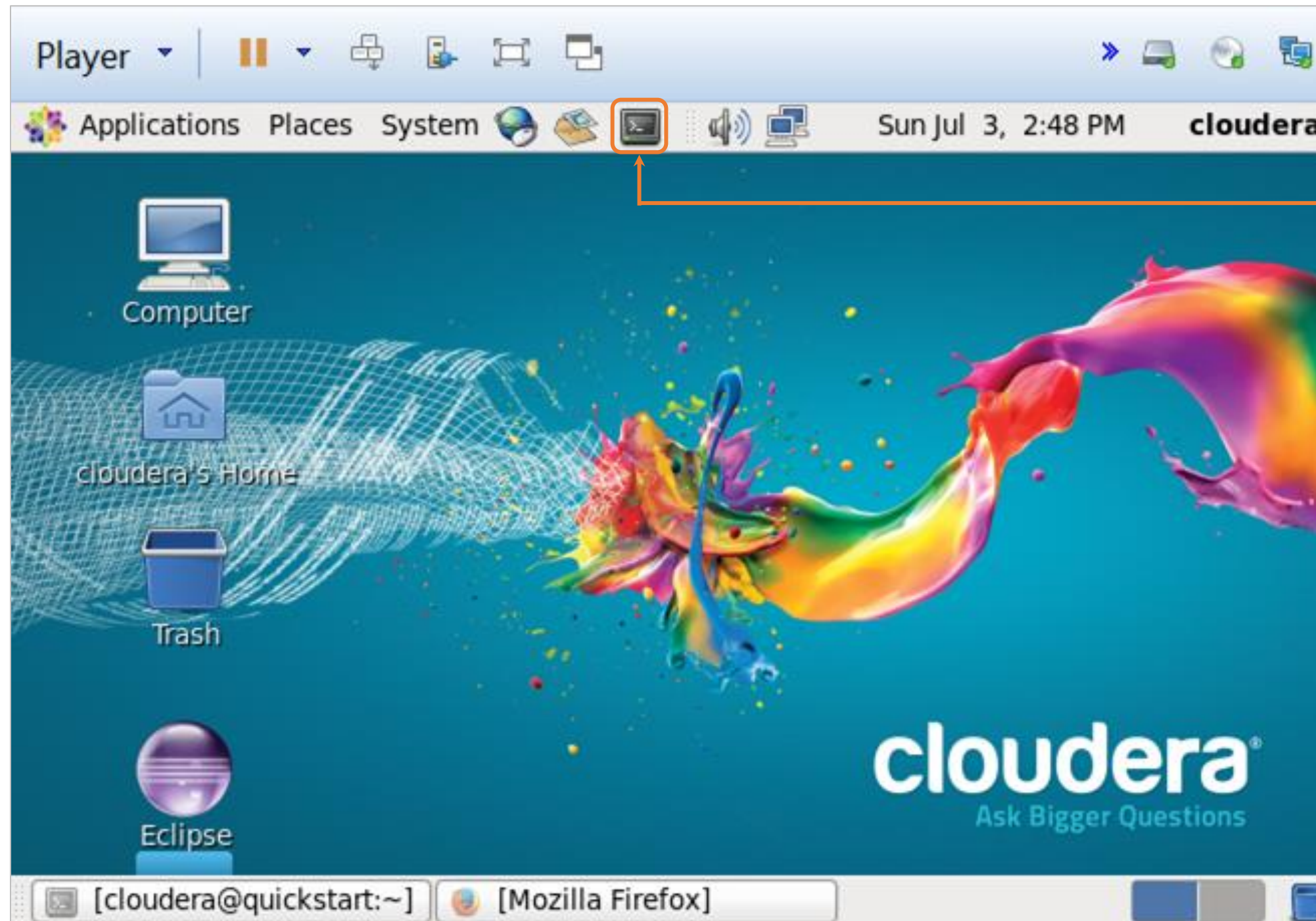
VMware Fusion 8 Pro

Download Now



# QuickStart VMware Image

Launch VMware player with Cloudera VM



Launch Terminal

**Account:**  
username: cloudera  
password: cloudera

# QuickStart VM Terminal

## Step 1

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$
```

## Step 2

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ pwd  
/home/cloudera  
[cloudera@quickstart ~]$ ls -lrt  
total 1036  
drwxrwsr-x 9 cloudera cloudera 4096 Feb 24 2013 eclipse  
drwxrwxr-x 4 cloudera cloudera 4096 Apr 23 2015 workspace  
drwxrwxr-x 2 cloudera cloudera 4096 Apr 23 2015 lib  
drwxrwxr-x 4 cloudera cloudera 4096 Apr 23 2015 Documents  
drwxrwxr-x 2 cloudera cloudera 4096 Apr 23 2015 Desktop  
drwxrwxr-x 2 cloudera cloudera 4096 Apr 23 2015 datasets  
-rw-rw-r-- 1 cloudera cloudera 1092 Apr 23 2015 cm_api.sh  
-rwxrwxr-x 1 cloudera cloudera 3978 Apr 23 2015 cloudera-manager  
drwxr-xr-x 2 cloudera cloudera 4096 May 14 2015 Videos  
drwxr-xr-x 2 cloudera cloudera 4096 May 14 2015 Templates  
drwxr-xr-x 2 cloudera cloudera 4096 May 14 2015 Public  
drwxr-xr-x 2 cloudera cloudera 4096 May 14 2015 Pictures  
drwxr-xr-x 2 cloudera cloudera 4096 May 14 2015 Music  
drwxr-xr-x 2 cloudera cloudera 4096 May 14 2015 Downloads  
-rw-rw-r-- 1 cloudera cloudera 984565 Jun 30 12:00 test_file  
-rw-rw-r-- 1 cloudera cloudera 187 Jun 30 12:04 mapper.py  
-rw-rw-r-- 1 cloudera cloudera 51 Jun 30 12:07 example_test_file  
-rw-rw-r-- 1 cloudera cloudera 868 Jun 30 12:16 reducer.py  
-rw-rw-r-- 1 cloudera cloudera 21 Jul 3 15:04 test_01  
[cloudera@quickstart ~]$
```

Unix command :

- pwd to verify present working directory
- ls -lrt to list files and directories

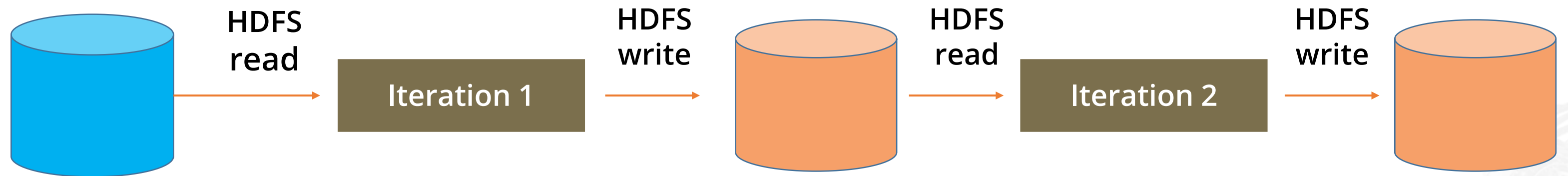
# Using Hadoop Streaming for Calculating Word Count



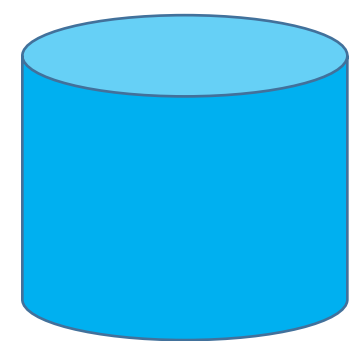
Demonstrate how to create a MapReduce program and use Hadoop Streaming to determine the word count of a document

ASSISTED PRACTICE

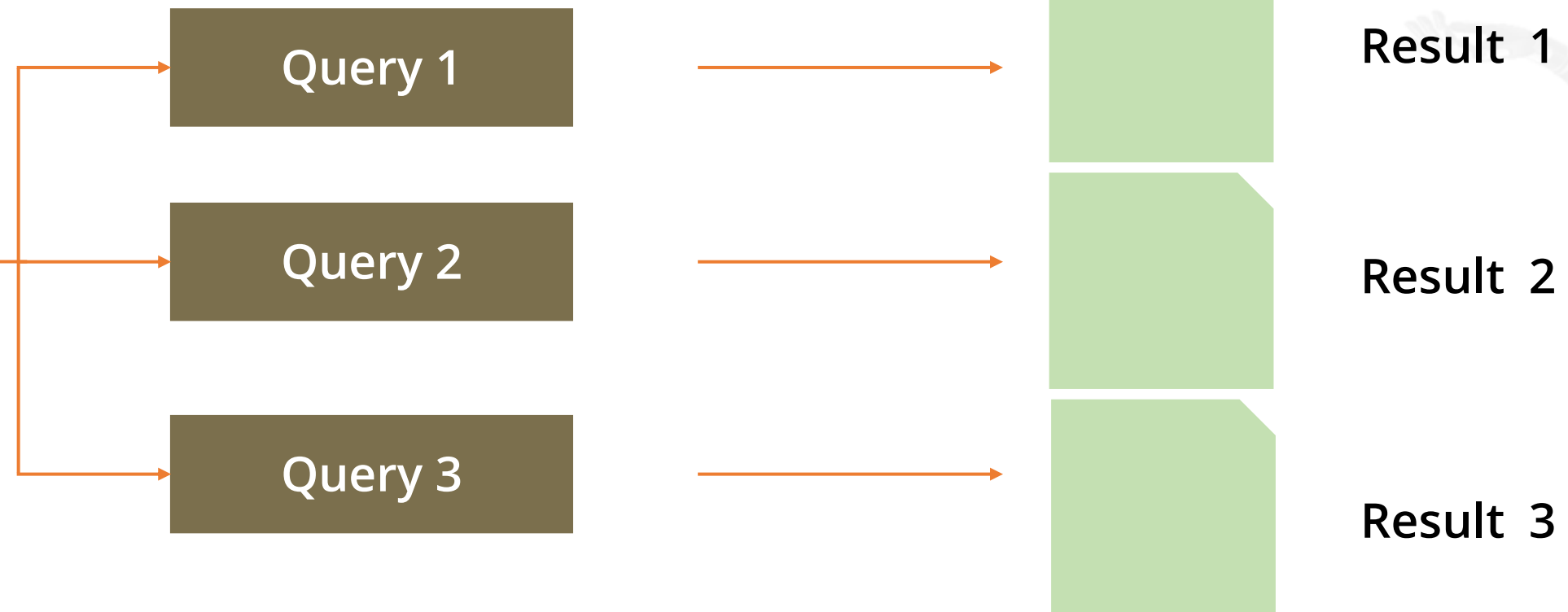
# MapReduce Uses Disk I/O Operations



Input



Input



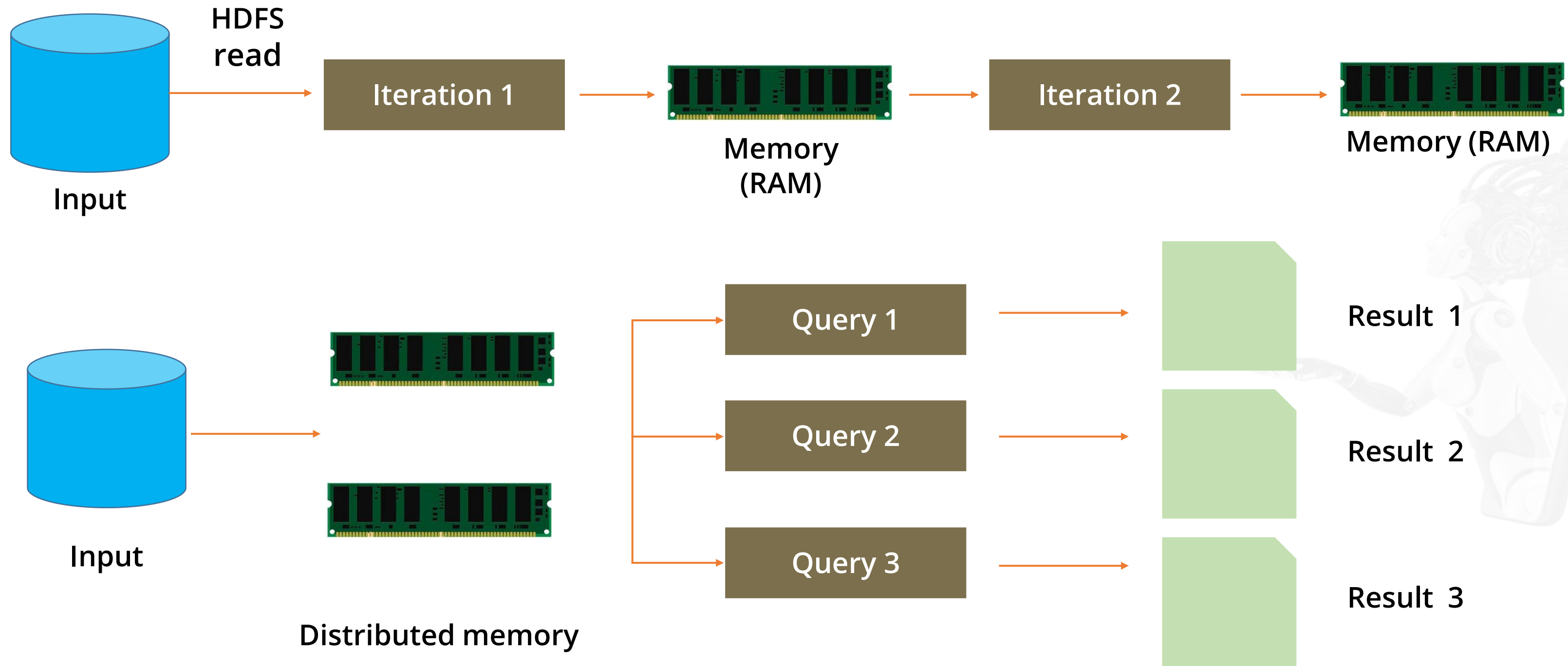
Result 1

Result 2

Result 3

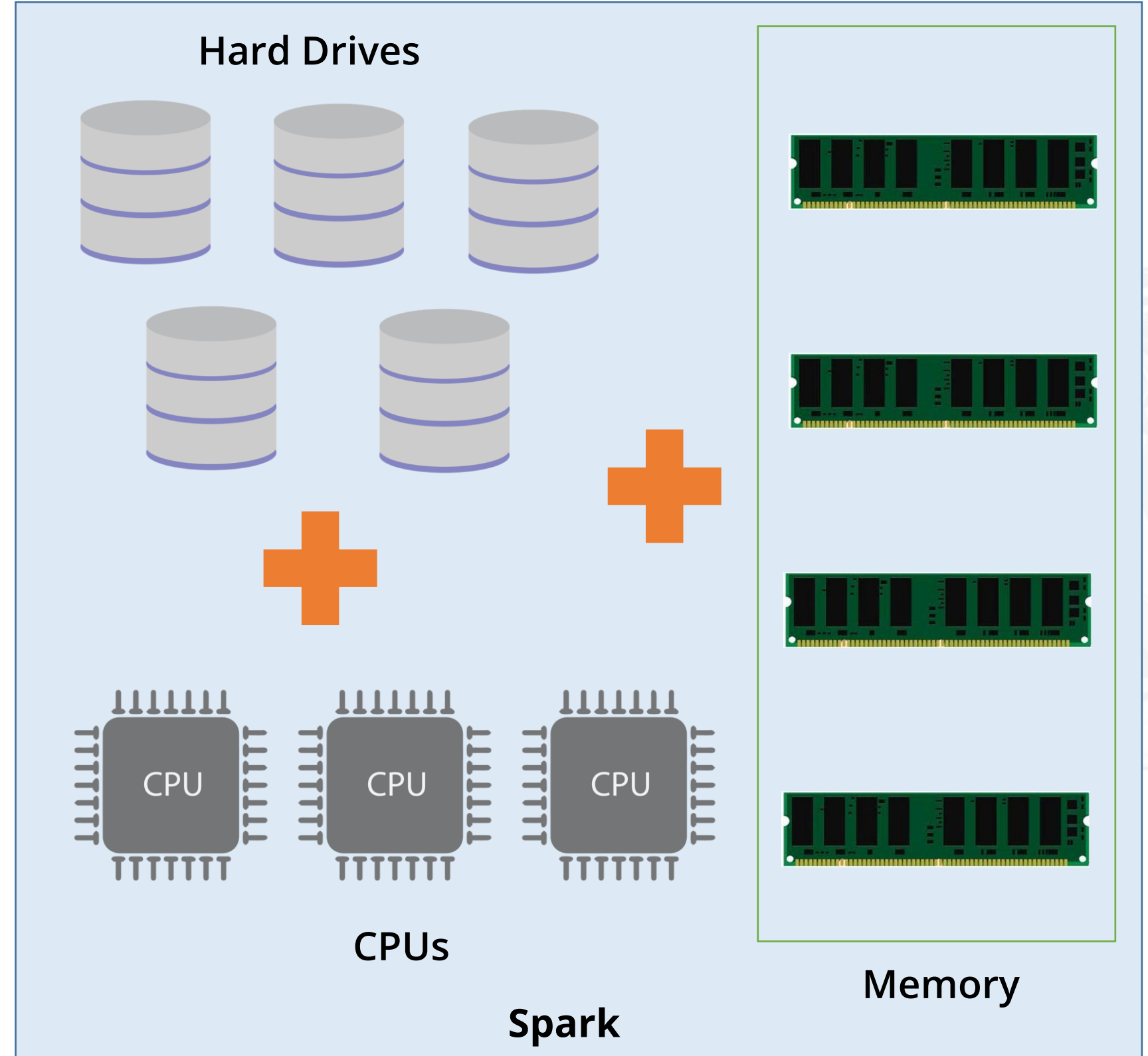
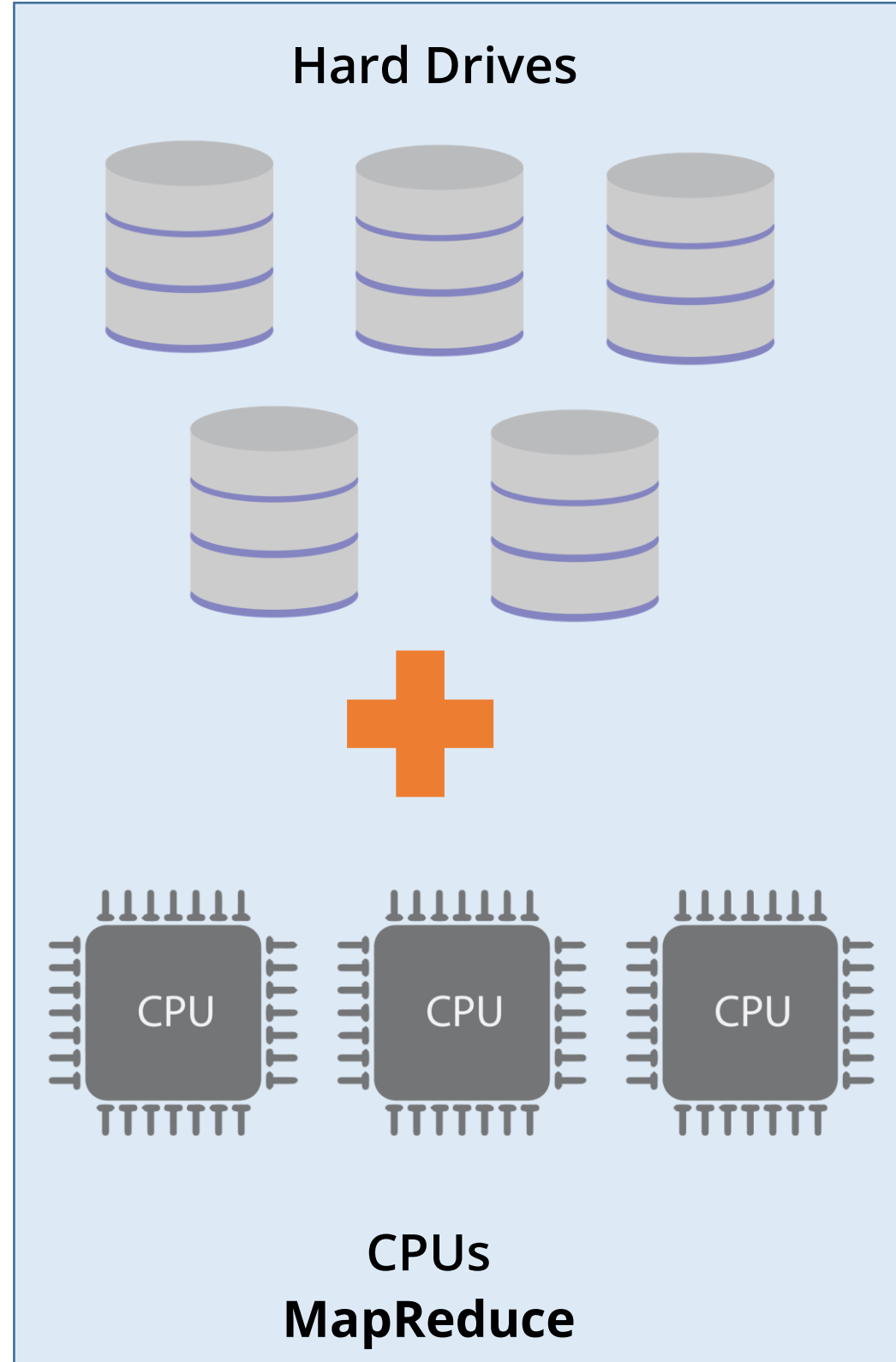


# Apache Spark Uses In-Memory Instead of Disk I/O



10-100 X faster than network and disk

# Hardware Requirements for MapReduce and Spark



# Apache Spark Resilient Distributed Systems (RDD)

Some basic concepts about Resilient Distributed Datasets (RDD) are:



- The main programming approach of Spark is RDD.
- They are fault-tolerant collections of objects spread across a cluster that you can operate on in parallel. They can automatically recover from machine failure.
- You can create an RDD either by copying the elements from an existing collection or by referencing a dataset stored externally.
- RDDs support two types of operations: transformations and actions.
  - Transformations use an existing dataset to create a new one.
    - Example: Map, filter, join
  - Actions compute on the dataset and return the value to the driver program.
    - Example: Reduce, count, collect, save



If the available memory is insufficient, then the data is written to disk.

# Advantages of Spark



|                    |   |
|--------------------|---|
| <b>Faster:</b>     | 10 to 100 times faster than Hadoop MapReduce  |
| <b>Simplified:</b> | <ul style="list-style-type: none"><li>• Simple data processing framework</li><li>• Interactive APIs for Python for faster application development</li></ul> |
| <b>Efficient:</b>  | Has multiple tools for complex analytics operations   |
| <b>Integrated:</b> | Can be easily integrated with existing Hadoop infrastructure  |

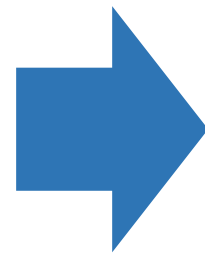


# PySpark: Python API for Spark

PySpark is the Spark Python API which enables data scientists to access Spark programming model.

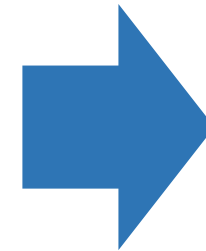


Data  
Scientist



PySpark

Python API



Spark

# PySpark: RDD Transformations and Actions

## Transformation

| Transformation | Description  |
|----------------|--|
| map()          | Returns RDD, formed by passing data element of the source                      |
| filter()       | Returns RDD based on selection   |
| flatMap()      | Maps items present in the dataset and returns sequence                         |
| reduceByKey()  | Returns key value pairs where values for which each key is aggregated by value |

## Action

| Action    | Description  |
|-----------|--|
| collect() | Returns all elements of the dataset as an array                              |
| count()   | Returns the number of elements present in the dataset                        |
| first()   | Returns the first element in the dataset                                     |
| take(n)   | Returns number of elements (n) as specified by the number in the parenthesis |

**SparkContext or SC** is the entry point to spark for the spark application

# Spark Tools

Spark  
SQL

Spark  
Streaming

MLlib  
(machine  
learning)

GraphX  
(graph)

Spark



Interactive Python APIs

# Setting Up Apache Spark

To set up the Apache Spark environment, access the link:

<http://spark.apache.org/downloads.html>

Please use [7-Zip](#) to extract these files.



Download Libraries ▾ Documentation ▾ Examples Community ▾ FAQ

## Download Apache Spark™

Our latest stable version is Apache Spark 1.6.2, released on June 25, 2016 ([release notes](#)) ([git tag](#))

1. Choose a Spark release:
2. Choose a package type:
3. Choose a download type:
4. Download Spark: [spark-1.6.2-bin-hadoop2.4.tgz](#)
5. Verify this release using the [1.6.2 signatures and checksums](#).

*Note: Scala 2.11 users should download the Spark source package and build [with Scala 2.11 support](#).*

# Setting Up Environmental Variable of Apache Spark

Environment Variables



User variables for niteen

| Variable   | Value  |
|------------|--|
| PATH       | C:\Niteen\Anaconda2;C:\Niteen\Anaconda2\Scripts;C:\Niteen\An...  |
| SPARK_HOME | C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-... |
| TEMP       | %USERPROFILE%\AppData\Local\Temp                                 |
| TMP        | %USERPROFILE%\AppData\Local\Temp                                 |

New...

Edit...

Delete

System variables

| Variable                     | Value  |
|------------------------------|--|
| MONETDB_INSTALL_DIR          | C:\Pentaho\monetdb   |
| NUMBER_OF_PROCESSORS         | 4  |
| OnlineServices               | Online Services  |
| OS                           | Windows_NT   |
| Path                         | C:\ProgramData\Oracle\Java\javapath;C:\Program Files (x86)\Inte... |
| PATHEXT                      | .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC              |
| PENTAHO_HOME                 | C:\Pentaho   |
| PENTAHO_INSTALLED_LICENSE... | C:\Pentaho\installedlicenses.xml                                   |

New...

Edit...

Delete

OK

Cancel

[installed directory]\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4

[installed directory] \spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin



# Integrating Jupyter Notebook with Apache Spark

Command Prompt - pyspark

```
C:\Users\niteen>cd C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin
C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin>set PYSARK_DRIVER_PYTHON=ipython
C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin>set PYSARK_DRIVER_PYTHON_OPTS=notebook
C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin>pyspark
[W 19:33:44.451 NotebookApp] Permission to listen on port 8888 denied
[I 19:33:44.612 NotebookApp] Serving notebooks from local directory: C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin
[I 19:33:44.618 NotebookApp] 0 active kernels
[I 19:33:44.618 NotebookApp] The Jupyter Notebook is running at: http://localhost:8889/
[I 19:33:44.716 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Setup the  
pyspark  
notebook  
specific  
variables

jupyter PySpark - Env Set Up Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Help

Code CellToolbar

```
In [1]: sc
Out[1]: <pyspark.context.SparkContext at 0x48b9a20>
```

```
In [ ]:
```

Run the pyspark  
command

Check SparkContext

# Using PySpark to Determine Word Count



Demonstrate how to use the Jupyter integrated PySpark API to determine the word count of a given dataset

ASSISTED PRACTICE

# Word Count



Determine the word count of the given Amazon dataset:

- Create a MapReduce program to determine the word count of the Amazon dataset
- Submit the MapReduce task to HDFS and run it
- Verify the output

UNASSISTED PRACTICE

# Word Count



Use the given dataset to count and display all the airports based in New York using PySpark. Perform the following steps:

- View all the airports listed in the dataset
- View only the first 10 records
- Filter the data for all airports located in New York
- Clean up the dataset if required

UNASSISTED PRACTICE



## Knowledge Check



## Knowledge Check

1

What are the core components of Hadoop? *Select all that apply.*

- a. MapReduce
- b. HDFS
- c. Spark
- d. RDD



## Knowledge Check

1

What are the core components of Hadoop? *Select all that apply.*

- a. MapReduce
- b. HDFS
- c. Spark
- d. RDD



The correct answer **a and b** is

MapReduce and HDFS are the core components of Hadoop.

## Knowledge Check

2

MapReduce is a data processing framework which gets executed \_\_\_\_.

- a. at DataNode
- b. at NameNode
- c. on client side
- d. in memory



## Knowledge Check

2

MapReduce is a data processing framework which gets executed \_\_\_\_.

- a. at DataNode
- b. at NameNode
- c. on client side
- d. in memory



The correct answer **a** is

The MapReduce program is executed at the data node and the output is written to the disk.

**Knowledge  
Check**  
**3**

Which of the following functions is responsible for consolidating the results produced by each of the Map() functions/tasks?

- a. Reducer
- b. Mapper
- c. Partitioner
- d. All of the above





## Knowledge Check

3

Which of the following functions is responsible for consolidating the results produced by each of the Map() functions/tasks?

- a. Reducer
- b. Mapper
- c. Partitioner
- d. All of the above



The correct answer **a** is

Reducer combines or aggregates results produced by mappers.

## Knowledge Check

4

What transforms input key-value pairs to a set of intermediate key-value pairs?

- a. Mapper
- b. Reducer
- c. Combiner
- d. Partitioner



## Knowledge Check

4

What transforms input key-value pairs to a set of intermediate key-value pairs?

- a. Mapper
- b. Reducer
- c. Combiner
- d. Partitioner



The correct answer **a** is

Mapper processes input data to intermediate key-value pairs which are in turn processed by reducers.

## Key Takeaways

You are now able to:

- 🕒 Explain why Python should be integrated with Hadoop
- 🕒 Outline the ecosystem and architecture of Hadoop
- 🕒 Explain the functioning of MapReduce
- 🕒 Discuss Apache Spark functions and their benefits
- 🕒 Write Python programs for Hadoop operations



# Stock Market Data Analysis



Import the financial data using Yahoo data reader for the following companies:

- Yahoo
- Apple
- Amazon
- Microsoft
- Google

Perform fundamental data analysis

- Fetch the previous year's data
- View the values of Apple's stock
- Display the plot of closing price
- Display the stock trade by volume
- Plot all companies' data together for closing prices



# Stock Market Data Analysis



Perform Daily Return Analysis and show the relationship between different stocks

- Plot the percentage change plot for Apple's stock
- Show a joint plot for Apple and Google
- Use PairPlot to show the correlation between all the stocks

Perform risk analysis

# Titanic Data Set Analysis



On April 15, 1912, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This tragedy shocked the world and led to better safety regulations for ships. Here, we ask you to perform the analysis through the exploratory data analysis technique. In particular, we want you to apply the tools of machine learning to predict the survived passengers.

# Titanic Data Set Analysis



The details of these projects and their scope are below:

Data acquisition of the Titanic dataset

- Train dataset
- Test dataset

Perform the Exploratory Data Analysis (EDA) for train dataset

- Passengers age distribution
- Passengers survival by age
- Passengers survival breakdown
- Passengers class distribution
- Passengers embarkation by locations



# Titanic Data Set Analysis



Perform machine learning to train the model and

- Create user defined function to load train data set
- Create user defined function to load test data set
- Create machine model
- Train the machine
- Predict whether a passenger survived the tragedy or not
- Persist the mode for future re-use

**Thank You**