Intro to Spark

Week 12 - Day 02

How do we run our models on big data?

Summary from yesterday

- Operational DB vs. DWH
- Supercomputers vs. Parallelism
- Big data
- Map Reduce

A models takes 10 hours to train

(cross validation)

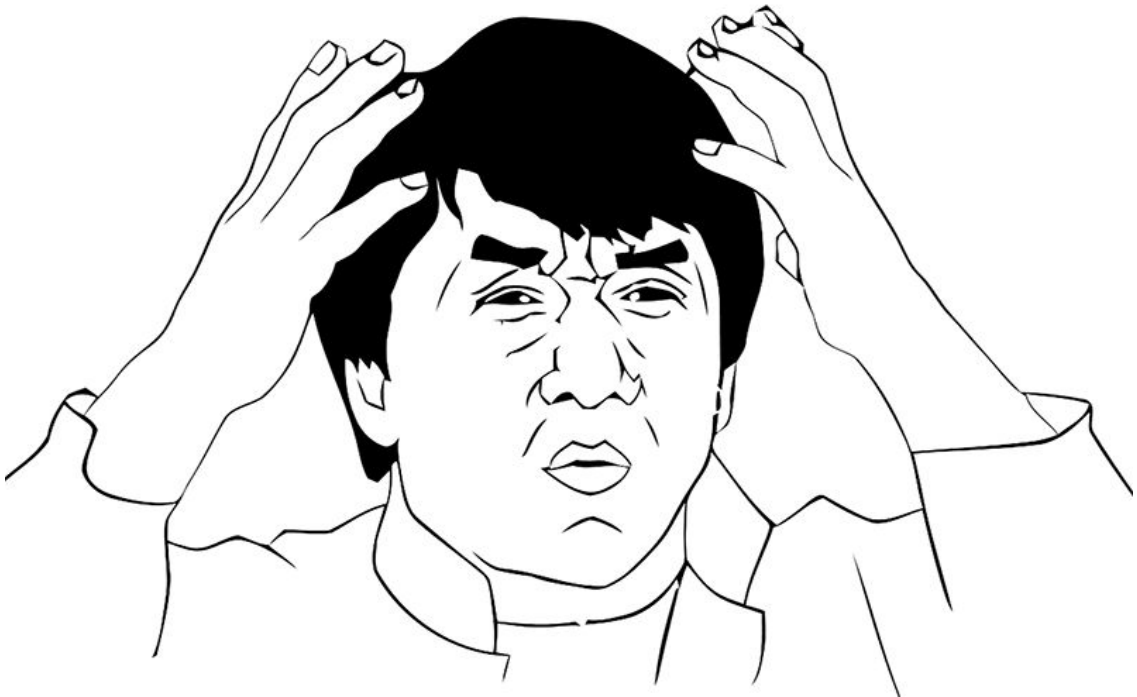
What do you do?

Map-Reduce is just a generic approach.

What is the implementation?

What is Hadoop?

It’s a “framework for big data”



Framework

A software **development** tool where all the **hard work** has been **done for** you

*I'm so glad I can use this **framework** instead of recreating all **the basic** stuff every time I start **a project**.*

Hadoop is a framework that allows you to use a cluster of serves as it’s one single server

Distributed file-system (HDFS)

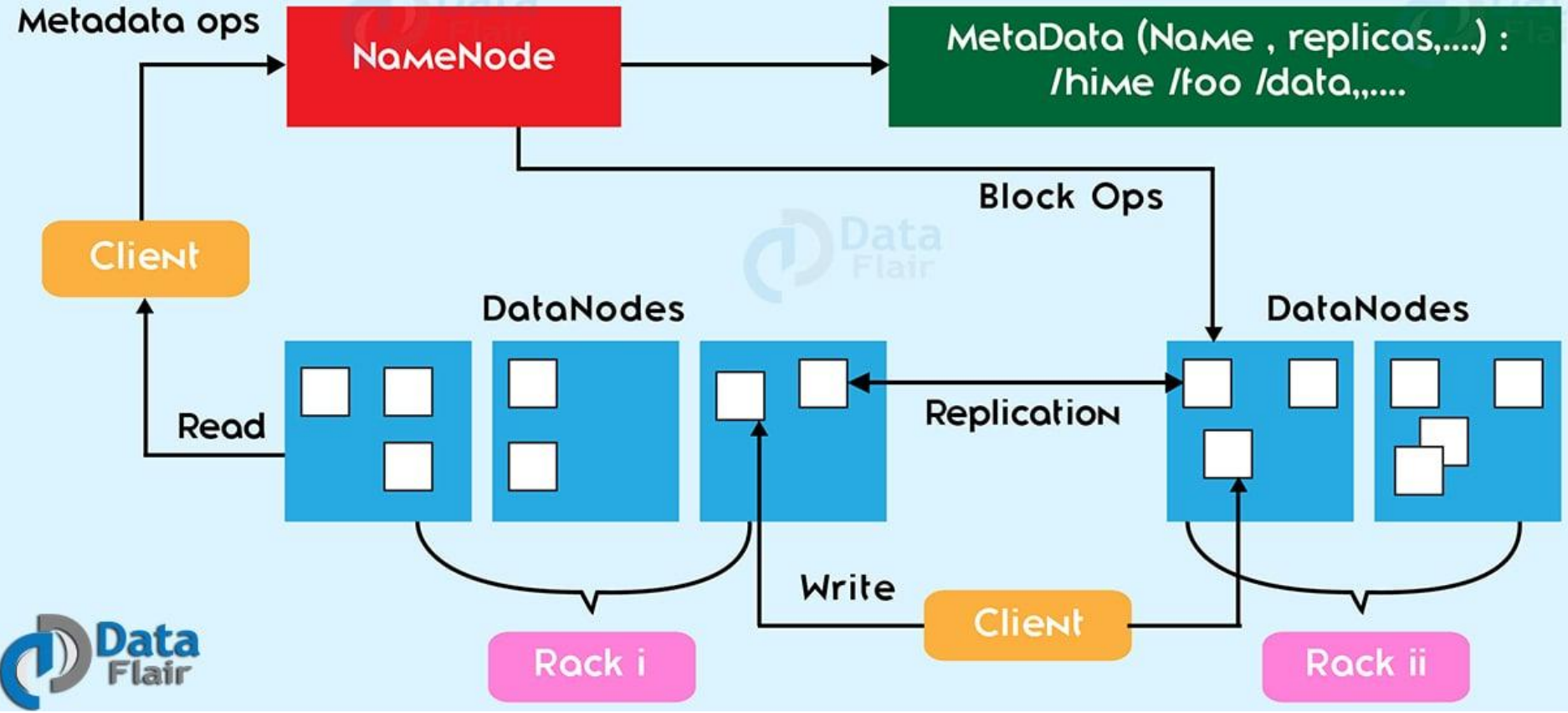
+

Map-Reduce

Distributed file-system

You see one file, in reality it’s split and replicated

HDFS Architecture



Map-Reduce

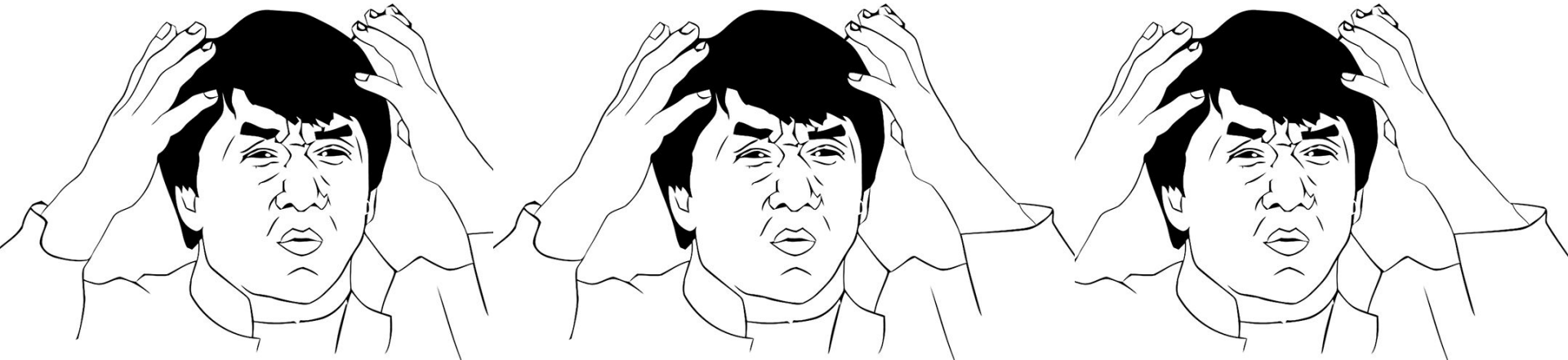
An approach to parallelize complex jobs

Hadoop is too low level

We need something more high level!

What is Spark?

It’s a “framework for big data”



Hadoop = basic low-level functions

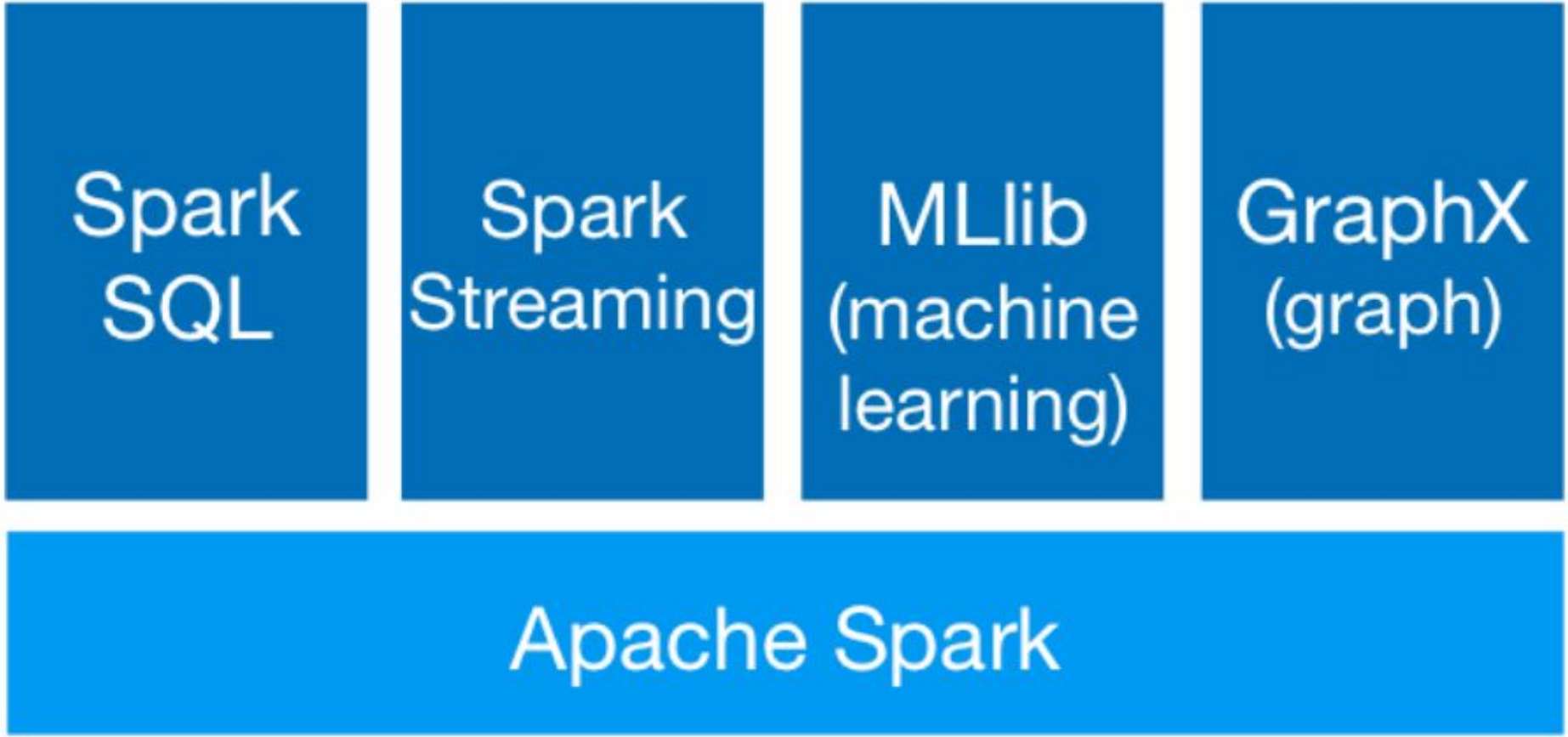
Spark = complex high-level functions

Hadoop = basic low-level functions

E.g. Numpy

E.g. Pandas

Spark = complex high-level functions


Spark SQL
Spark Streaming
MLlib (machine learning)
GraphX (graph)
Apache Spark

Spark+SQL = Hive

Hive = build tables on big data

We define structures on unstructured files

HiveSQL = query Hive tables

Spark+ML = MLib

Library to run ML models on Hadoop

Written in Scala

PySpark = Spark for python

We can't use Pandas

Spark RDD

Resilient Distributed Dataframe

Spark DataFrame

(similar to Pandas Dataframes)

Example: logistic regression in PySpark

```
from pyspark.ml.classification import LogisticRegression

# Load training data
training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")
```

```
from pyspark.ml.classification import LogisticRegression

# Load training data
training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)
```

```
from pyspark.ml.classification import LogisticRegression

# Load training data
training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

# Fit the model
lrModel = lr.fit(training)
```

```python
from pyspark.ml.classification import LogisticRegression

# Load training data
training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

# Fit the model
lrModel = lr.fit(training)

# Print the coefficients and intercept for logistic regression
print("Coefficients: " + str(lrModel.coefficients))
print("Intercept: " + str(lrModel.intercept))
```

Simple pandas operations can be complex in spark

Spark is always evolving

Summary

Hadoop = distributed FS + map-reduce

Spark = high level operations on Hadoop

Hive = SQL on big data

Spark + MLib = ML on big data