

Introduction to Spark

Joe

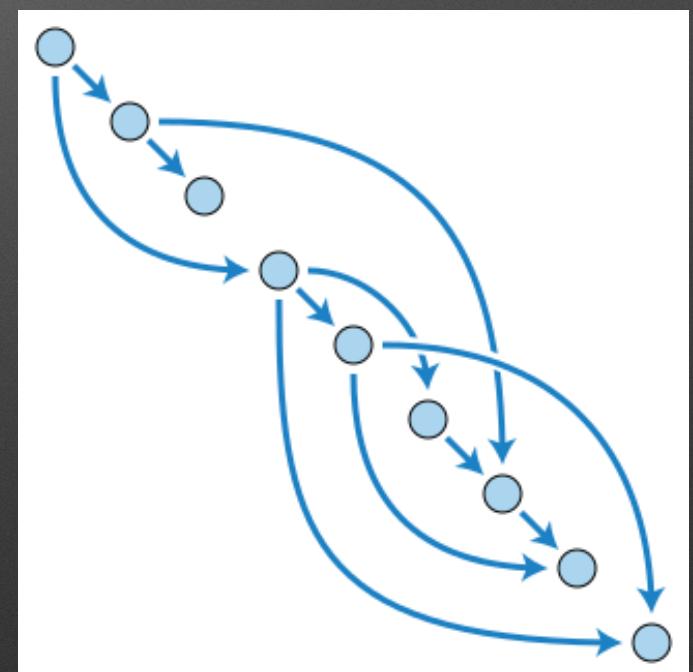
Introduction

Morning Objectives

1. Understand why Spark *can* be much faster, and discuss important tips to optimize performance.
2. Do the ‘hello world’ of distributed computing - perform a word count from a large corpus of documents.

Apache Spark

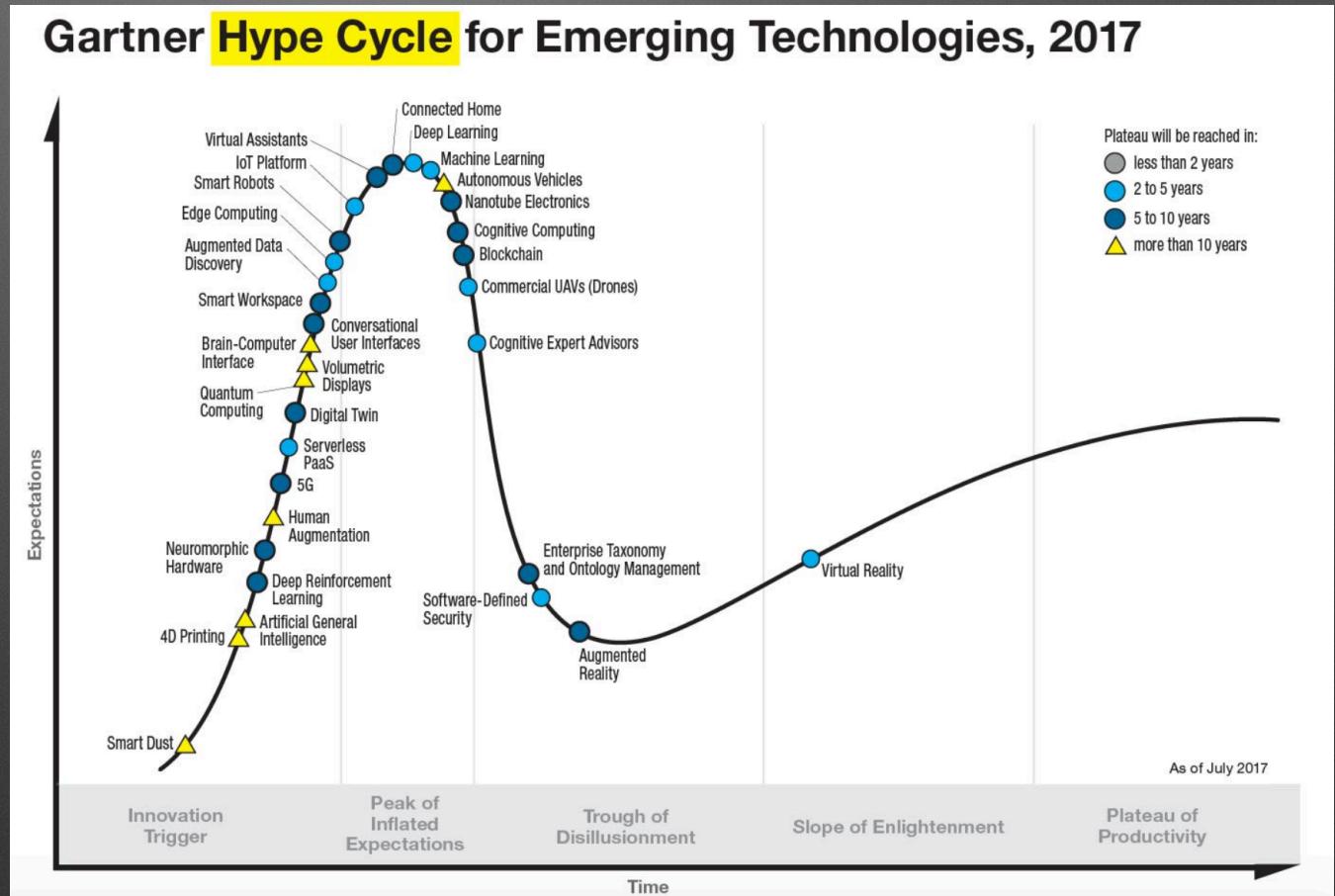
- Apache Spark is a fast and general engine for large-scale data processing
- Spark runs extremely fast because it performs in memory computations
- Spark has interfaces for Scala, Java, R, and Python
- Spark is an engine for building a directed analytic graph



A Note of Caution

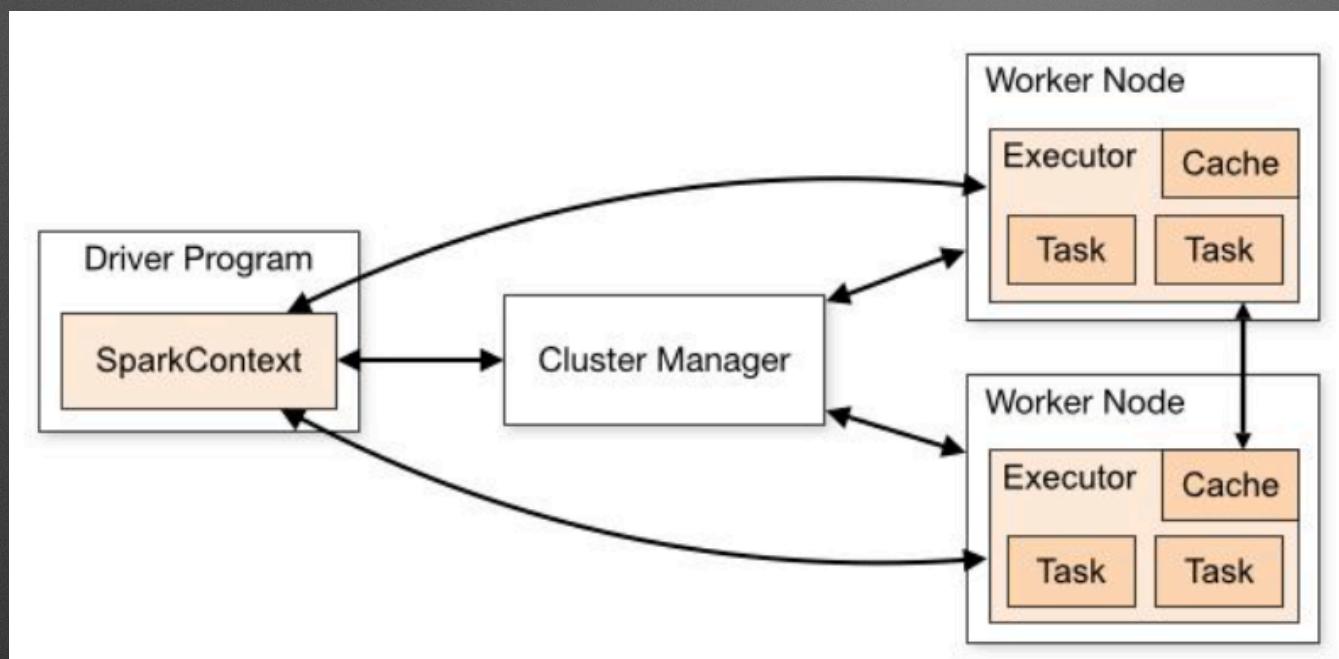
I love spark.

Strong fluency with
pandas/numpy/etc
are useful so do
not become overly
reliant on using
spark for data
manipulation!



Mechanics

Drivers & Workers

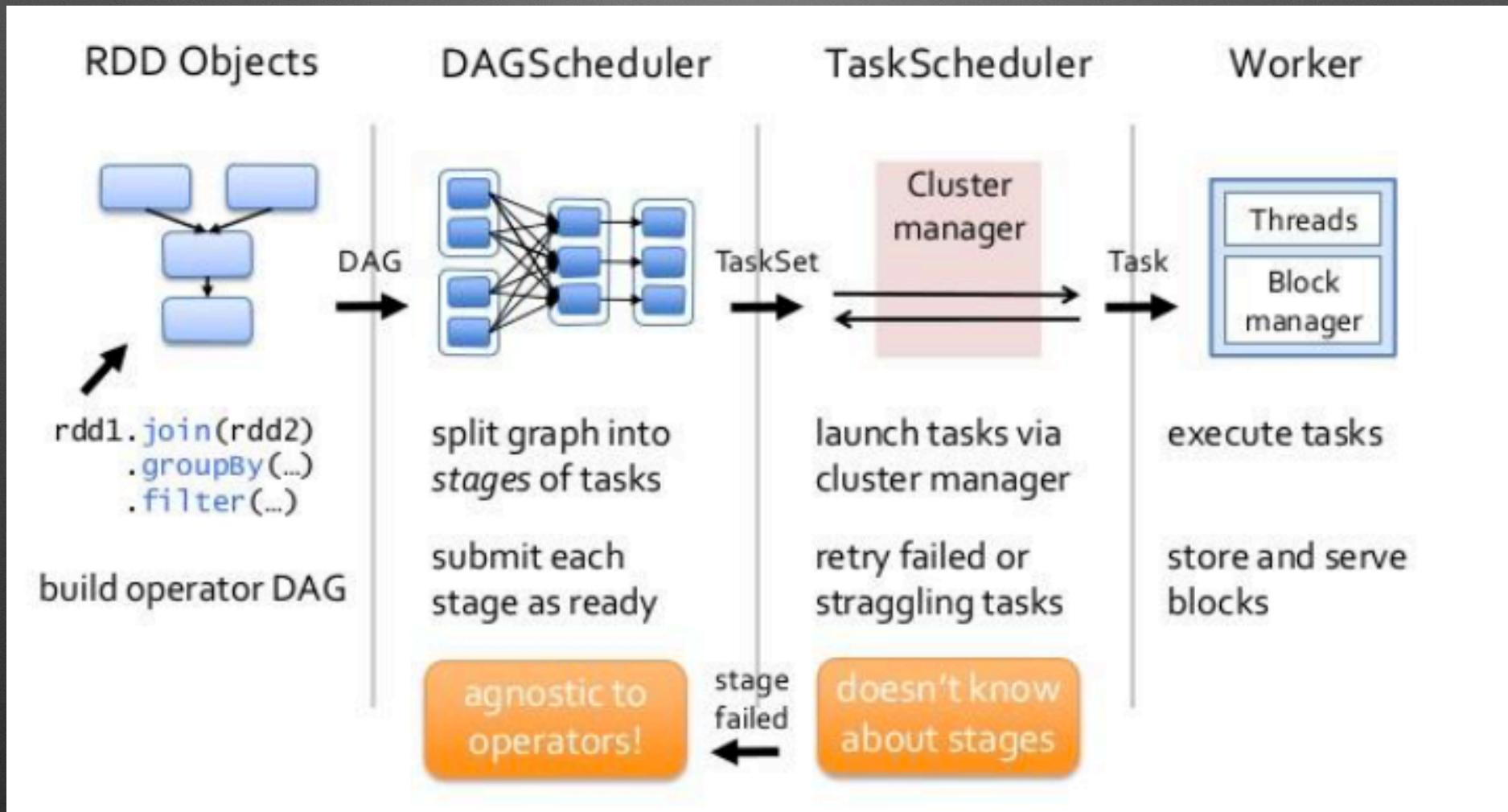


Application Properties

Property Name	Default
spark.app.name	(none)
spark.driver.cores	1
spark.driver.maxResultSize	1g
spark.driver.memory	1g
spark.executor.memory	1g

<https://spark.apache.org/docs/latest/configuration.html>

Scheduler Process

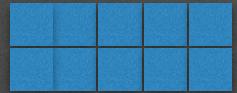


In Memory Computation

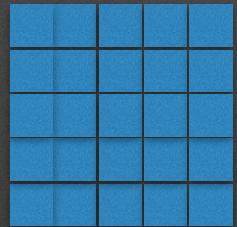
1 ns



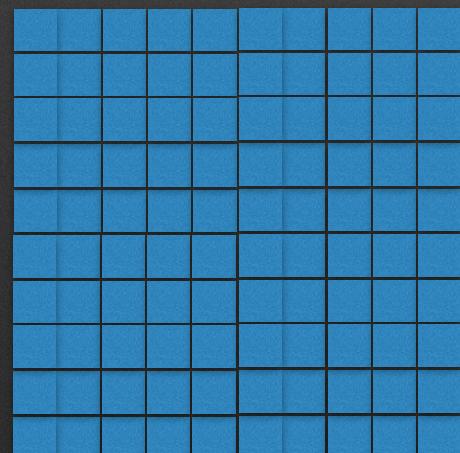
1 floating point operation



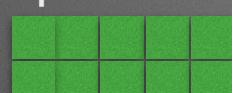
mutex lock



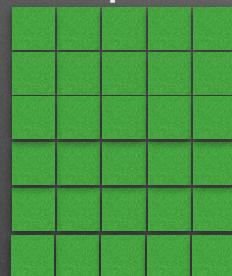
main memory ref



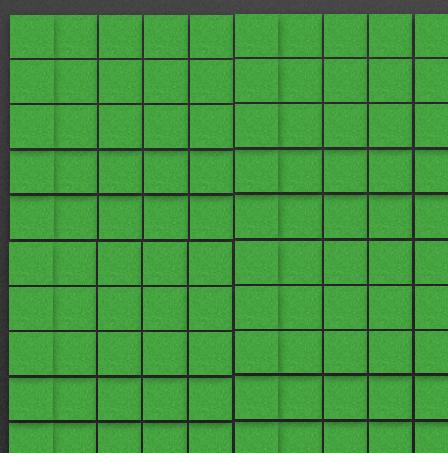
1μs



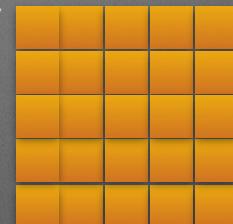
compress 1 KB



Send 1 KB over
1 Gbps network

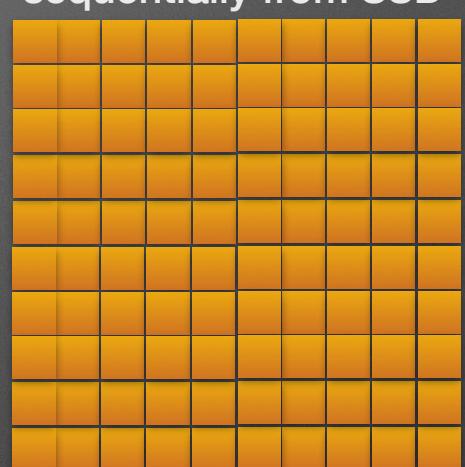


Read 1 MB
sequentially from
memory



Read 1 MB

sequentially from SSD



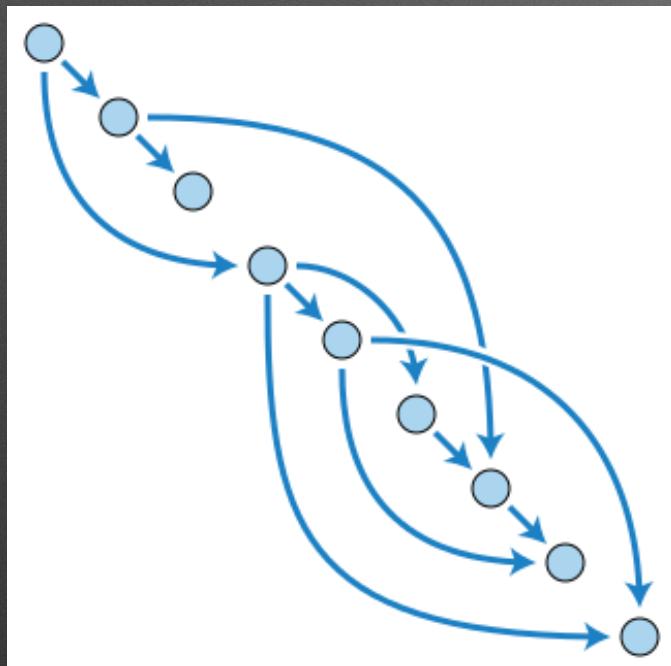
Read 1 MB
sequentially from disk



OK, time to get our hands
dirty

Using RDDs

RDDs



RDDs are the primary class introduced by Spark. It is a data container that allows for the construction of RDDs.

RDDs have the following properties:

- * Immutable
- * Lazily Evaluated
- * Cachable

RDD - Methods

```
rdd.flatMap(lambda text: text.split())\  
    .map(lambda word: (word.strip(".,-;?").lower(), 1))\  
    .reduceByKey(lambda a, b: a+b) \  
    .sortBy(lambda x: x[1], ascending=False) \  
    .take(10)
```

it did seem biased.

flatMap - one to many

it

did

seem

biased.

RDD - Methods

```
rdd.flatMap(lambda text: text.split())\  
    .map(lambda word: (word.strip(".,-;?").lower(), 1))\  
    .reduceByKey(lambda a, b: a+b) \  
    .sortBy(lambda x: x[1], ascending=False) \  
    .take(10)
```

biased.

map - one to one

(biased, 1)

RDD - Methods

```
rdd.flatMap(lambda text: text.split())\  
    .map(lambda word: (word.strip(".,-;?").lower(), 1))\  
    .reduceByKey(lambda a, b: a+b) .reduceByKey(lambda a, b: a+b) \  
    .sortBy(lambda x: x[1], ascending=False) \  
    .take(10)
```

(biased, 1)

reduceByKey - aggregate ordered, tuples with
hashable lead values

(biased, 375)

RDD - Methods

```
rdd.flatMap(lambda text: text.split())\  
    .map(lambda word: (word.strip(".,-;?")).lower(), 1))\  
    .reduceByKey(lambda a, b: a+b) \  
    .sortBy(lambda x: x[1], ascending=False) \  
    .take(10)
```

(biased, 375) (is, 2048) (yo, 12) (the, 3274)

sortBy - sort by a particular value

(the, 3274) (is, 2048) (biased, 375) (yo, 12)

RDD - Methods

```
rdd.flatMap(lambda text: text.split())\  
    .map(lambda word: (word.strip(",.-;?").lower(), 1))\  
    .reduceByKey(lambda a, b: a+b)\  
    .sortBy(lambda x: x[1], ascending=False) \  
    .take(10)
```

(the, 3274) (is, 2048) (biased, 375) (yo, 12)

take(n) - create a local list of length n whose entries are the RDD members (ex -> take(2))

[(the, 3274), (is, 2048)]

Morning Objectives

1. Understand why Spark *can* be much faster, and discuss important tips to optimize performance.
2. Do the ‘hello world’ of distributed computing - perform a word count from a large corpus of documents.

Afternoon Objectives

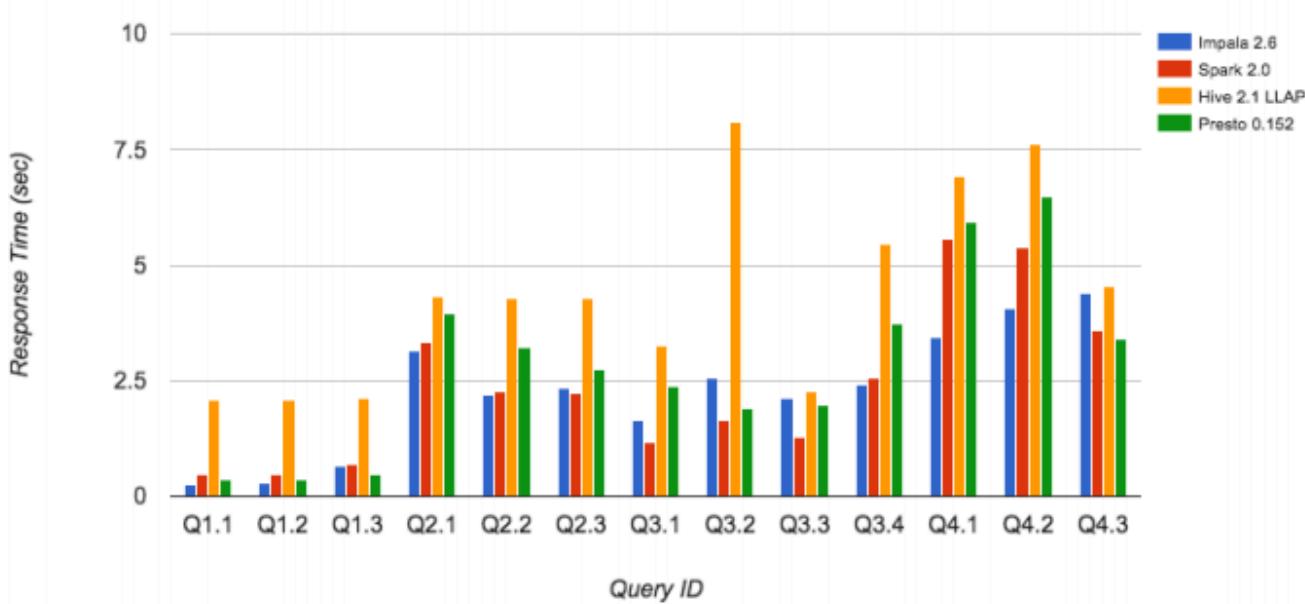
1. Learn to create spark dataframes
2. Learn to Query spark via SparkSQL

Afternoon Objectives

1. Learn to create spark dataframes
2. Learn to Query spark via SparkSQL

Alternative To SparkSQL - Impala

Figure 4: Benchmark Query Results for Cache Tables



Impala is an SQL engine for distributed tabular datasets

While it lacks many of the key features of spark, for information retrieval it can be superior

Notebook time!

Afternoon Objectives

1. Learn to create spark dataframes
2. Learn to Query spark via SparkSQL