APACHE SPARK

By Josh Fermin

WHAT IS APACHE SPARK?

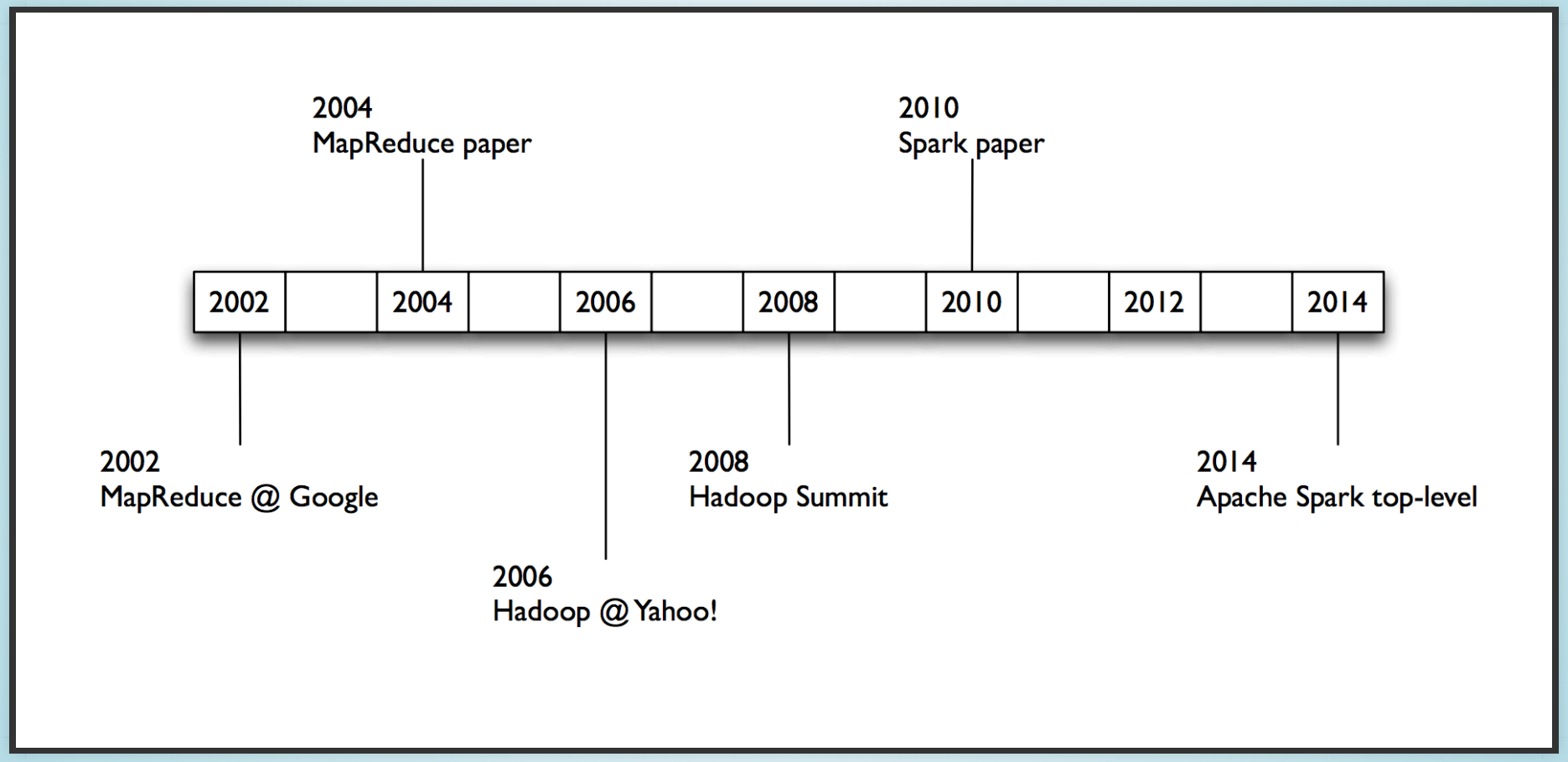
Open source data analytics cluster computing framework

Real-Time vs Interactive vs Batch processing

Write apps in Java, Scala or Python.

Map Reduce is only a part of supported capabilities.

HISTORY

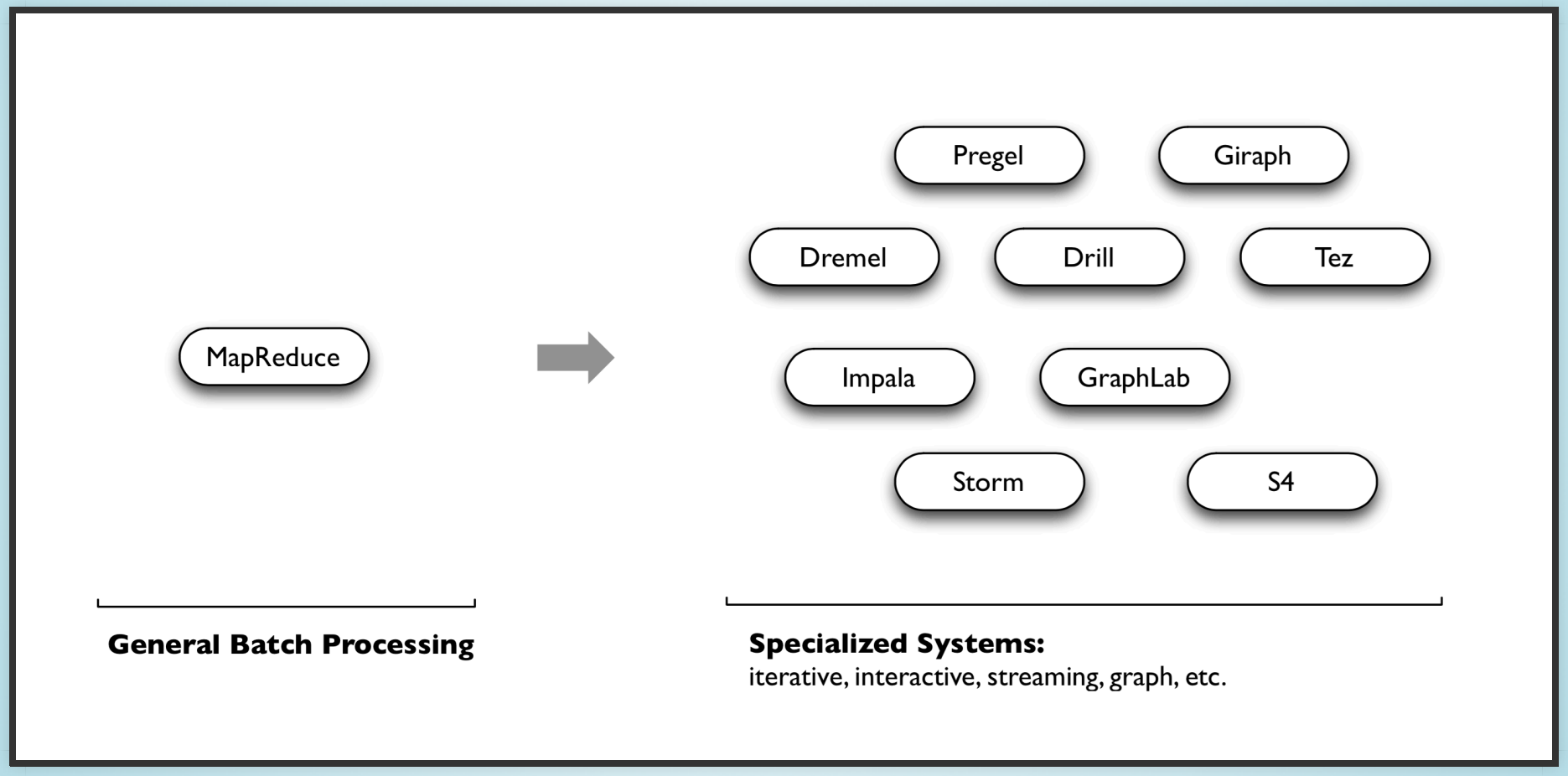

2004
MapReduce paper
2010
Spark paper
2002
2004
2006
2008
2010
2012
2014
2002
MapReduce @ Google
2006
Hadoop @ Yahoo!
2008
Hadoop Summit
2014
Apache Spark top-level

HISTORY - MAP REDUCE

Two major problems:

- difficulty programming in Map Reduce
- Batch style jobs not fitting all use cases

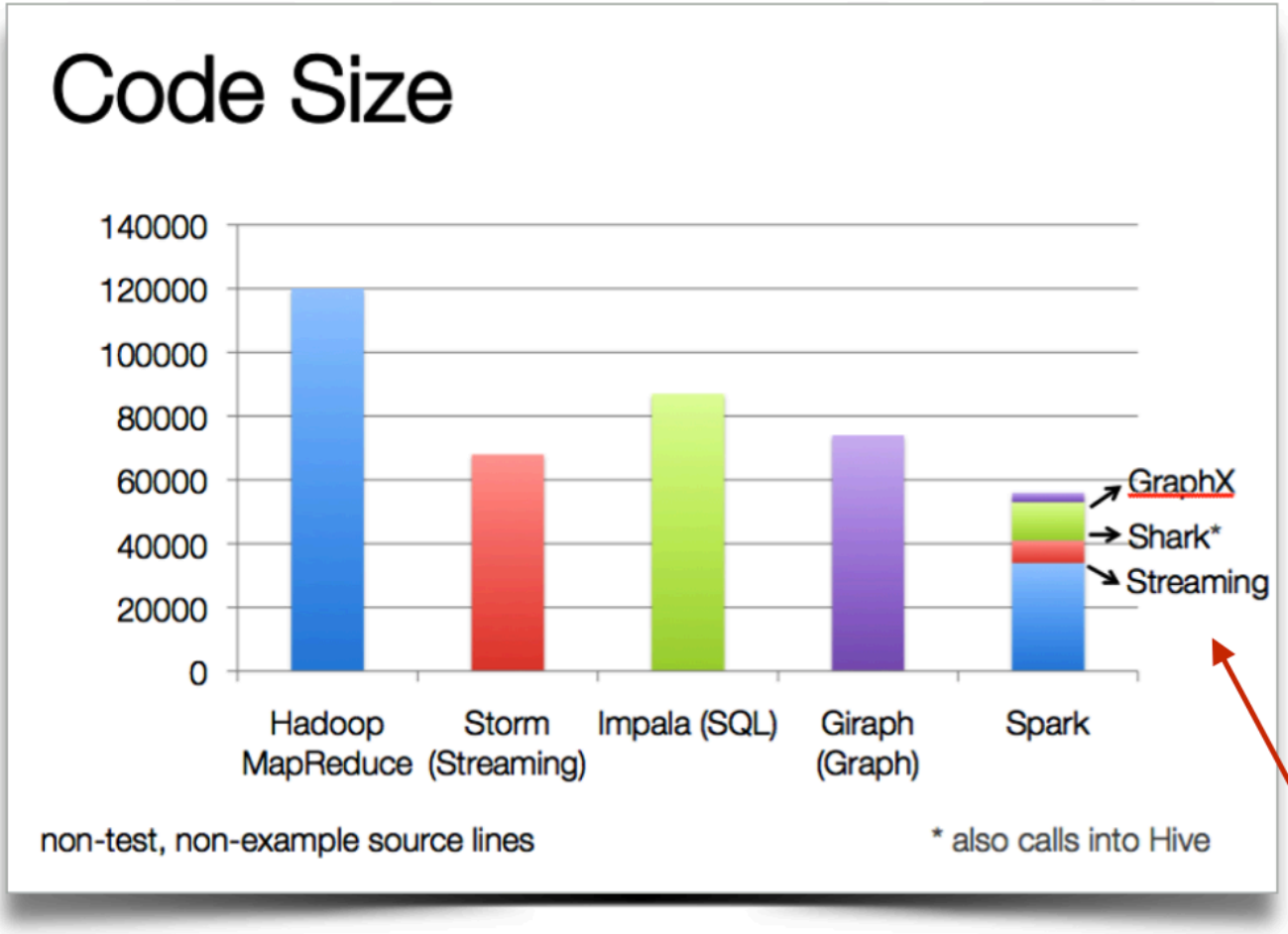
HISTORY - WORKAROUNDS



SPARK - GOALS

Generalize Map Reduce

- fast data sharing
- general DAGs (Directed Acyclic Graphs)



The State of Spark, and Where We're Going Next
Matei Zaharia
Spark Summit (2013)
youtu.be/nU6vO2EJAb4

used as libs, instead of specialized systems

LEARNING SPARK API

Problem: Count how many lines a word shows up in a document

SPARK SHELL - SCALA

Spark Download here (use prebuilt version)

```
./bin/spark-shell/
```

```
./bin/pyspark/
```

SPARK CONTEXT

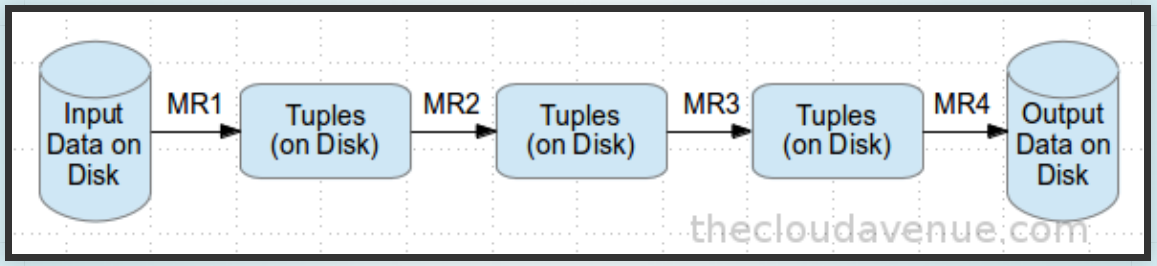
```
scala> val textFile = sc.textFile("README.md")
textFile: spark.RDD[String] = spark.MappedRDD@2ee9b6e3
```

- Start spark context
- Creates RDD (Resilient Distributed Dataset)

RDDS

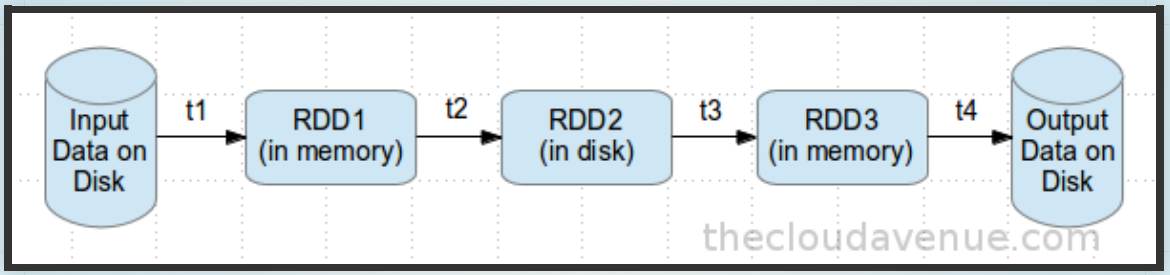
- Spark's primary abstraction
- Distributed collection of items
- Can be stored in the volatile memory or in a persistent storage

RDDS



- Reading and writing happens too often for each Map Reduce (MR)

RDDS



- Transformations from one to another are through memory and doesn't touch the disk (except for RDD2)
- When the memory runs out, it is usually moved over to persistent storage.

RDD ACTIONS

```
scala> textFile.count() // Number of items in this RDD
res0: Long = 126

scala> textFile.first() // First item in this RDD
res1: String = # Apache Spark
```

RDD TRANSFORMATIONS

```
val linesWithSpark = textFile.filter(line => line.contains("Spark"))
linesWithSpark: spark.RDD[String] = spark.FilteredRDD@7dd4af09
```

CHAINING ACTIONS AND TRANSFORMATIONS

```
// How many lines contain "Spark"?
textFile.filter(line => line.contains("Spark")).count()
res3: Long = 15
```

MORE RDD OPERATIONS

Finding the longest line in a text file:

```
textFile.map(line => line.split(" ").size)
    .reduce((a, b) => if (a > b) a else b)
res4: Long = 15
```

```
scala> import java.lang.Math
import java.lang.Math

textFile.map(line => line.split(" ").size)
    .reduce((a, b) => Math.max(a, b))
res5: Int = 15
```

MAP REDUCE

Finding word count

```
scala> val wordCounts = textFile.flatMap(line => line.split(" "))
         .map(word => (word, 1)).reduceByKey((a, b) => a + b)
wordCounts: spark.RDD[(String, Int)] = spark.ShuffledAggregatedRDD@71f02
```

```
scala> wordCounts.collect()
res6: Array[(String, Int)] = Array((means,1), (under,2),
 (this,3), (Because,1), (Python,2), (agree,1), (cluster.,1), ...)
```

CACHING

Cluster-wide in-memory cache

Useful for when data is accessed repeatedly (i.e. iterative algorithms or re-querying a small dataset)

CACHING EXAMPLE

```
scala> linesWithSpark.cache()
res7: spark.RDD[String] = spark.FilteredRDD@17e51082

scala> linesWithSpark.count()
res8: Long = 15
```

WHERE TO GO FROM HERE

- Spark Programming Gudie
- Deployment overview
- Spark Examples

WHO USES SPARK?

- SK Telecom - Analyze mobile usage patterns
- Freeman Lab - analyzing/visualizing patterns in large scale recordings of brain acitivty
- Yandex - Using spark to process islands identified from a search robot

RESOURCES

- Spark Quick Start / Spark Docs
- Intro To Apache Spark - Stanford
- Learning about RDDs