INTRODUCTION TO SPARK

AGENDA

- Big Data Processing Challenges and Solutions
- How it is different from Hadoop MapReduce
- Introducing Spark
- Unified Platform of Spark
- Spark on Hadoop

Problems with Large scale data processing

- Data is Value. We must process it to extract value. How to store? How can we process all the data? Variety of data?
- Two Problems
  - ✓ Large-scale data storage
  - ✓ Large-scale data analysis

- Fortunately, the size and cost of storage has kept pace

Year	Capacity (GB)	Cost per GB (USD)
1997	2.1	157
2004	200	1.05
2014	3,000	0.036

- Unfortunately, transfer rates have not kept with capacity

Year	Cpacity (GB)	Transfer Rate (MB/s)	Disk Read Time
1997	2.1	16.6	126 seconds
2004	200	56.5	59 minutes
2014	3,000	210	3 hours, 58 minutes

Solutions and Trends

- Late 1990s - explosive growth e-commerce and machine data implied that workloads could not fit on a single computer anymore…

- Notable firms (Amazon, eBay, Yahoo, Google) led the shift to *horizontal scale-out* on clusters of commodity hardware, especially for machine learning use cases at scale.

- 2002 - Mitigate risk of large distributed workloads lost due to disk failures on commodity hardware…

- GFS (Google File System) and MapReduce and Hadoop

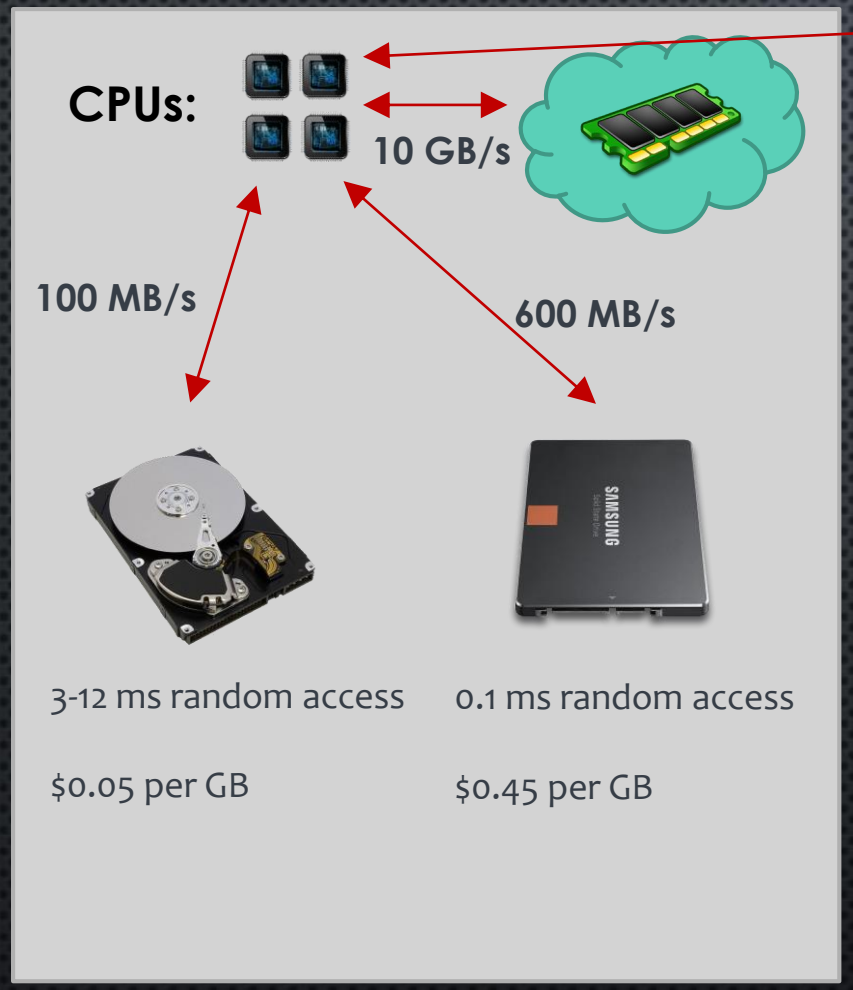
- 2010 - A unified engine for enterprise data workflows, based on commodity hardware, a decade later…

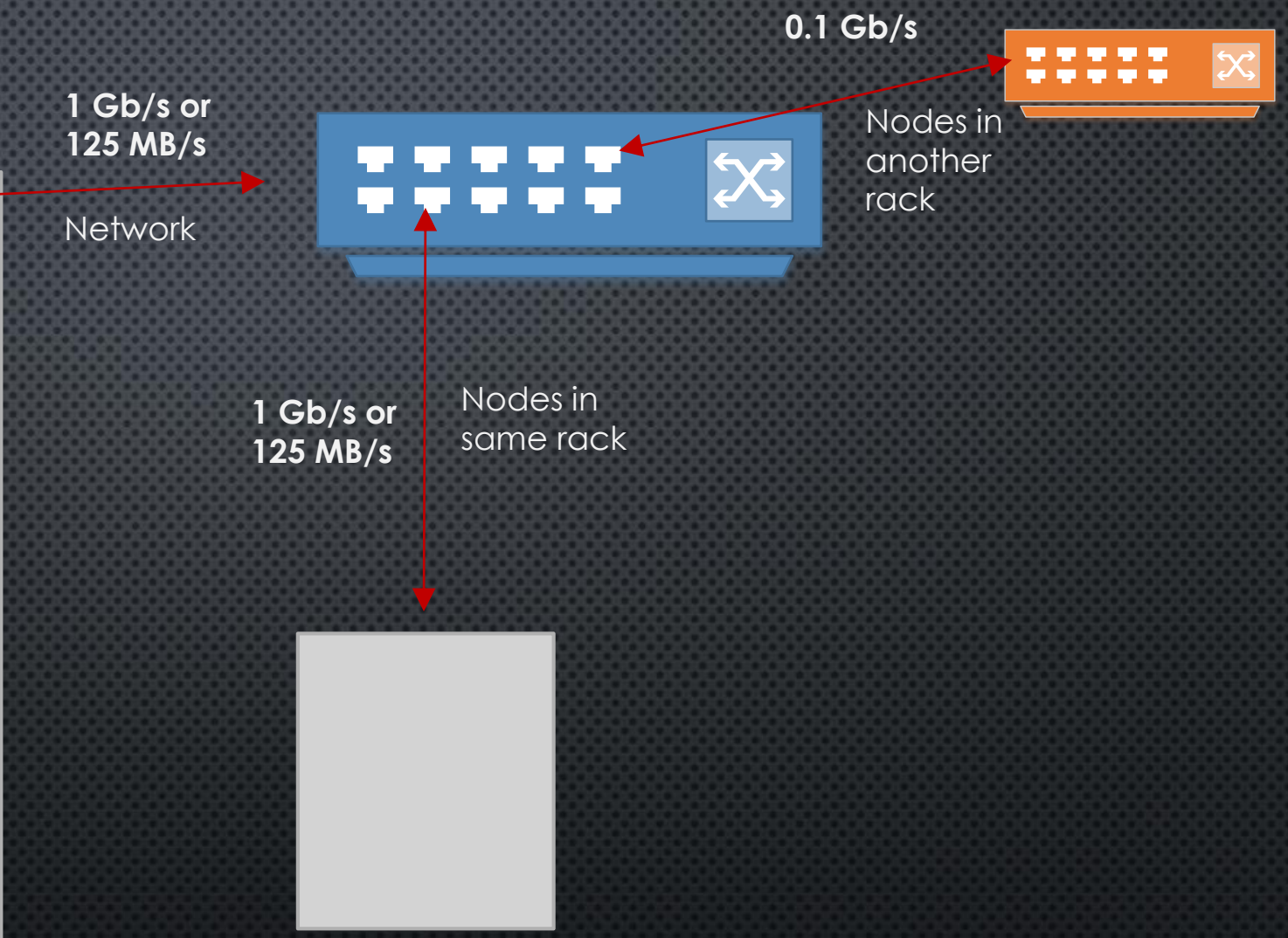
- Apache Spark and RDDs

Solutions and Trends

Cheap Network	Reference	URI
Cheap Storage	Replicate	DFS
Cheap Memory	Recompute	RDD

Why RAM?


CPUs:
10 GB/s
100 MB/s
600 MB/s
SAMSUNG
3-12 ms random access
0.1 ms random access
$0.05 per GB
$0.45 per GB


1 Gb/s or
125 MB/s
Network
0.1 Gb/s
Nodes in
another
rack
1 Gb/s or
125 MB/s
Nodes in
same rack

Solutions and Trends

- Decreasing storage costs have led to an explosion of big data
- Commodity cluster software, like Hadoop, has made it 10-20x cheaper to store and process large datasets
- Broadly available from multiple vendors

Implication

Big data storage is becoming commoditized, so how will organizations get an edge?

What matters now is what you can *do* with the data.

Two Factors

Speed: how quickly can you go from data to decisions?

Sophistication: can you run the best algorithms on the data?

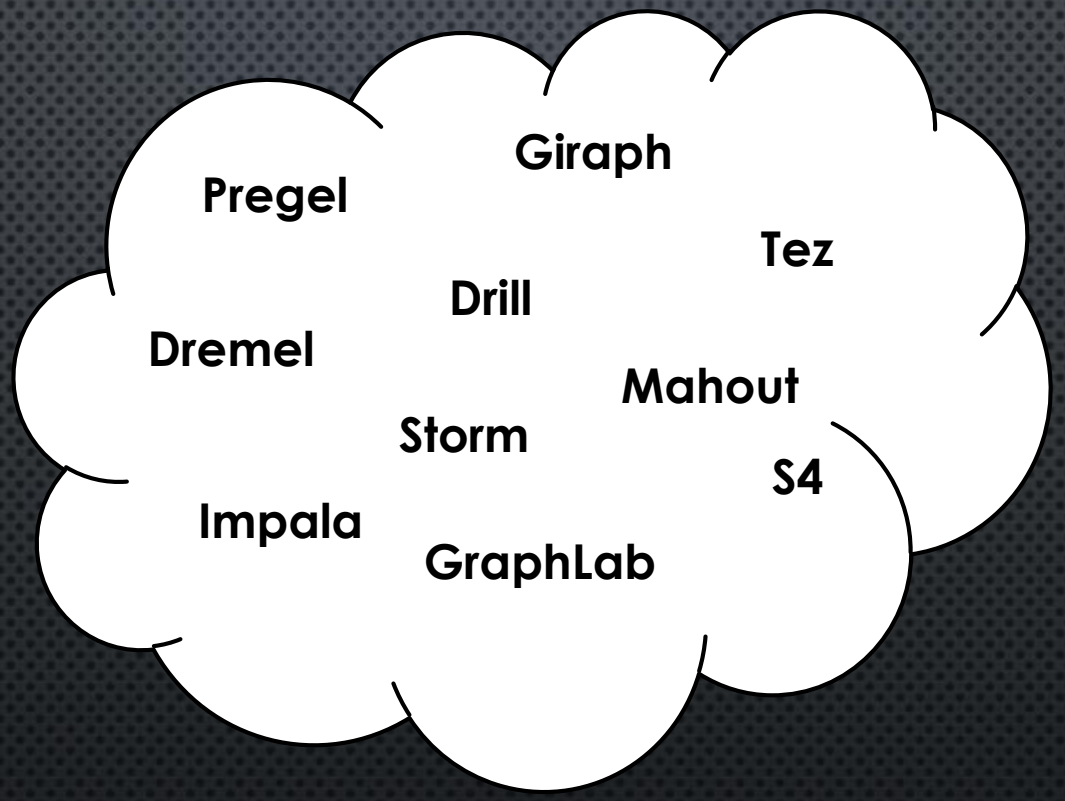
How easily you can develop apps?

How easily you can explore data?

These factors have usually required separate and non-commodity tools

(2004 – 2013)

(2007 – 2015?)



(2014 – ?)

Specialized Systems

(iterative, interactive, ML, streaming, graph, SQL, etc)

General Batch Processing

General Unified Engine

Introducing Apache Spark

- Developed in 2009 at UC Berkeley AMPLab
- Open sourced in 2010
- Feb 2014 - Top Level Project at the Apache Software Foundation
- Spark has since become one of the largest OSS communities in big data
- Over 250+ contributors in 50+ organizations

spark.apache.org

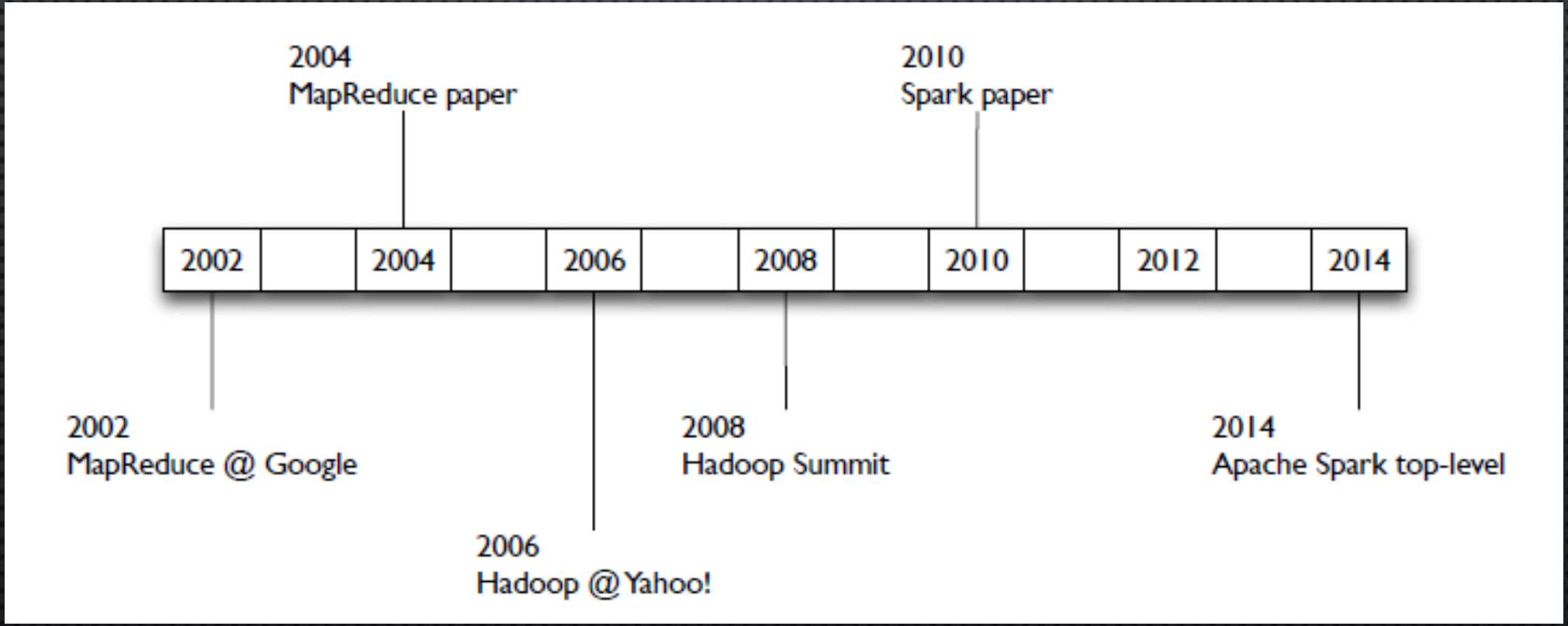
github.com/apache/spark

user@spark.apache.org

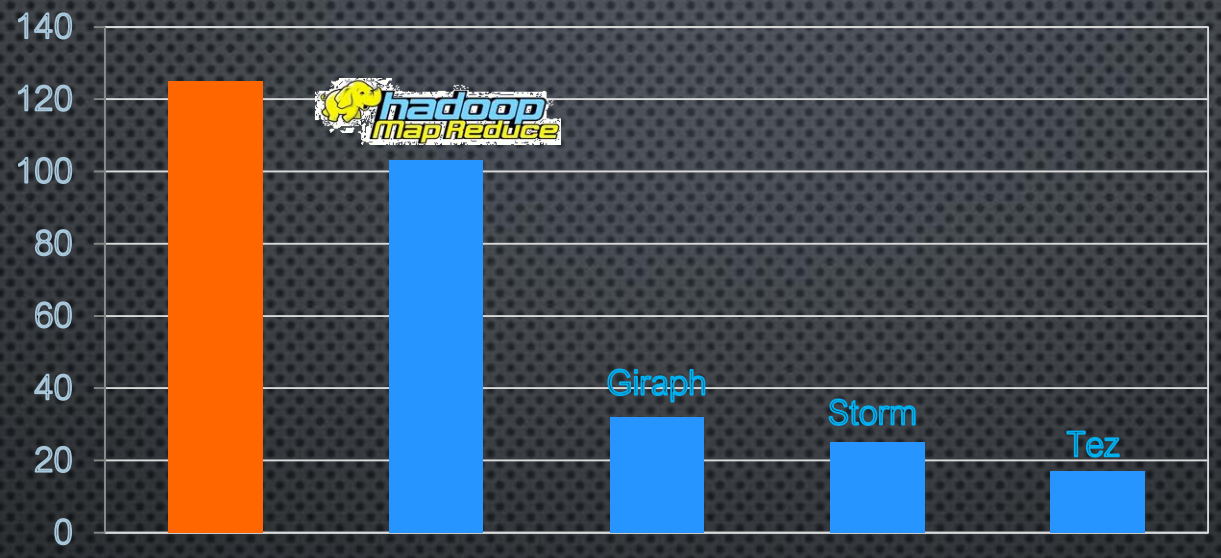
Introducing Apache Spark

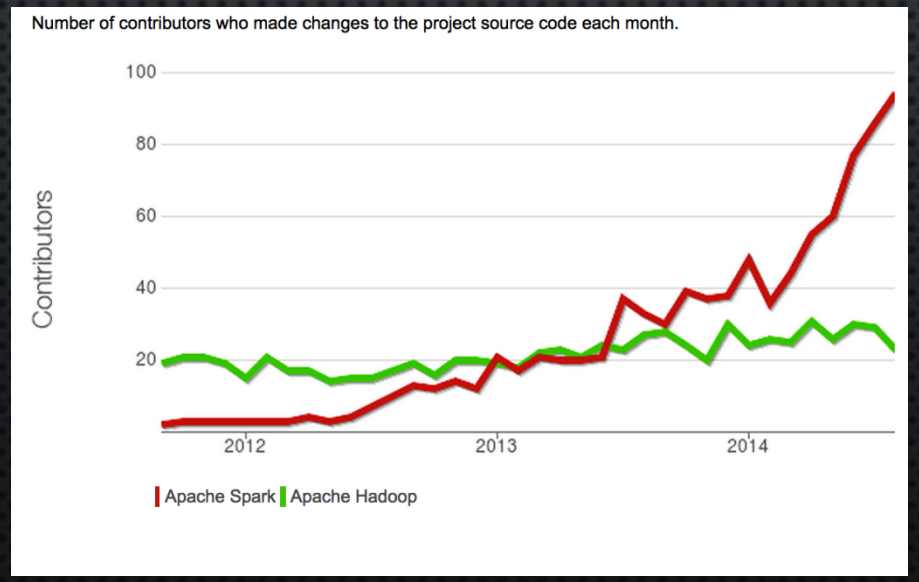
- Apache Spark is a fast and expressive cluster computing system interoperable with Apache Hadoop.
- **Written in Scala**
  - Functional programming language that runs in a JVM
- **100% Open Source**
- **Key Concepts**
  - ✓ Avoid the data bottleneck by distributing data when it is stored
  - ✓ Bring the processing to the data
  - ✓ In-memory computing capabilities deliver speed
  - ✓ General execution model supports wide variety of use cases
  - ✓ Ease of development – native APIs in Java, Scala, Python (+ SQL, Clojure, R)
  - ✓ Compatible with Hadoop's storage API's (HDFS, S3, Avro…)
  - ✓ Useful for large datasets and Iterative algorithms.
  - ✓ Up to 100x faster that Hadoop
  - ✓ Interactive Shell
  - ✓ Often 2-10× less code

Brief History



Most active open source project in Big Data





Hadoop MapReduce Issues

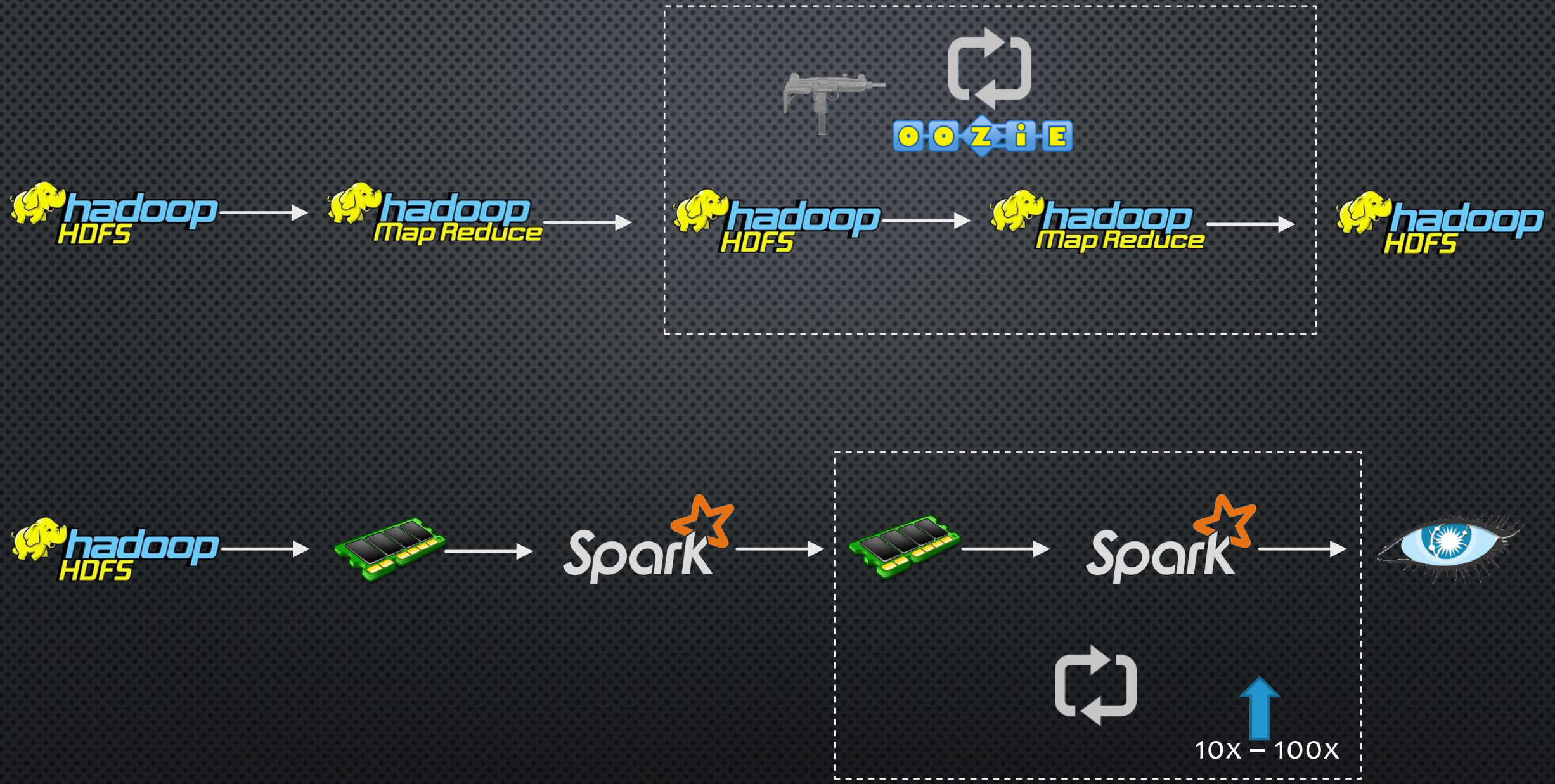
- Launching Mappers and Reducers takes time
- One MR job can rarely do a full computation
- Writing to disk (in triplicate!) between each job
- Going back to queue between jobs
- No in-memory caching
- No iterations
- Very high latency
- Not the greatest of APIs either

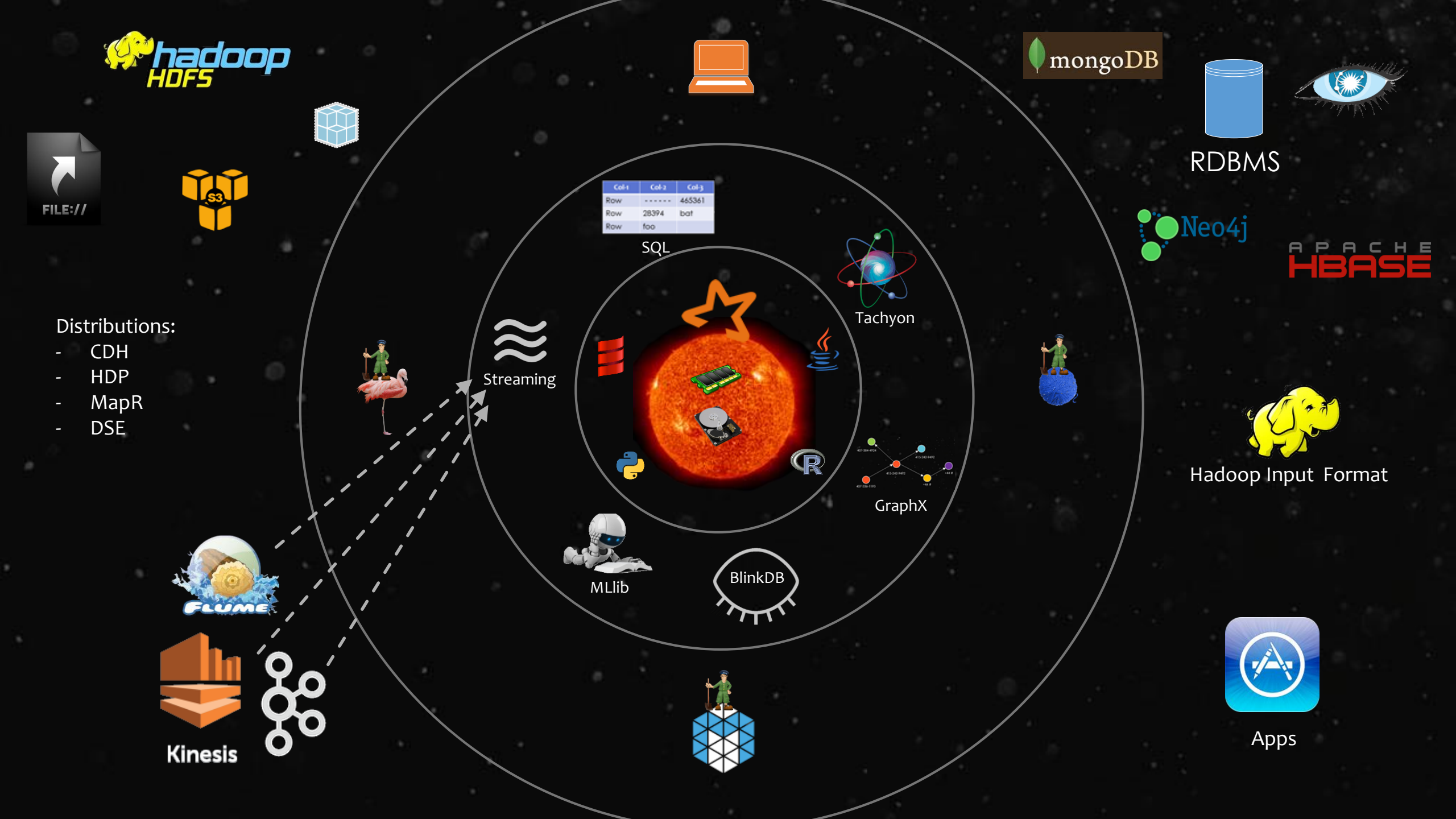
- MapReduce is dying….Hive, Pig and Mahout on Spark

https://cwiki.apache.org/confluence/display/Hive/Hive+on+Spark%3A+Getting+Started

http://blog.cloudera.com/blog/2014/09/pig-is-flying-apache-pig-on-apache-spark/

https://www.mapr.com/blog/making-mahout-fast-and-easy#.VHc1X4vF_DI


hadoop
HDFS
hadoop
Map Reduce
OOZIE
hadoop
HDFS
hadoop
Map Reduce
hadoop
HDFS
hadoop
HDFS
Spark
Spark
10x – 100x


hadoop
HDFS
FILE://
S3
mongoDB
RDBMS
Neo4j
APACHE
HBASE
Col-1 Col-2 Col-3
Row ------ 465361
Row 28394 bat
Row foo
SQL
Tachyon
Streaming
GraphX
MLlib
BlinkDB
Hadoop Input Format
Apps
FLUME
Kinesis
Distributions:
- CDH
- HDP
- MapR
- DSE

What it means to users?

Spark:

Interactive analysis

HDFS

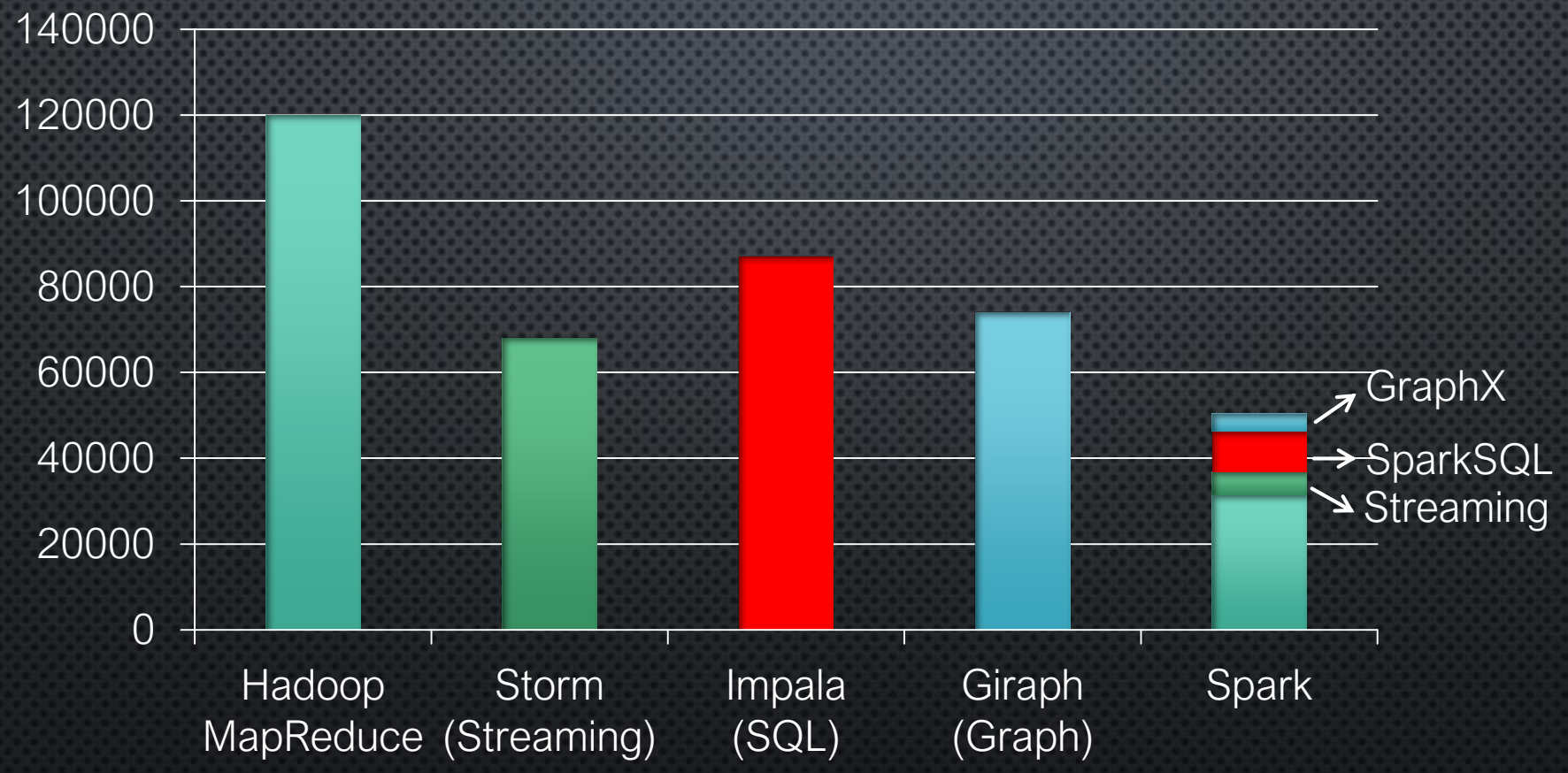
Combine Processing Types…

```
val points = sqlContext.sql(
  “SELECT latitude, longitude FROM historic_tweets”)

val model = KMeans.train(points, 10)

sc.twitterStream(...)
   .map(t => (model.closestCenter(t.location), 1))
   .reduceByWindow(“5s”, _ + _)
```

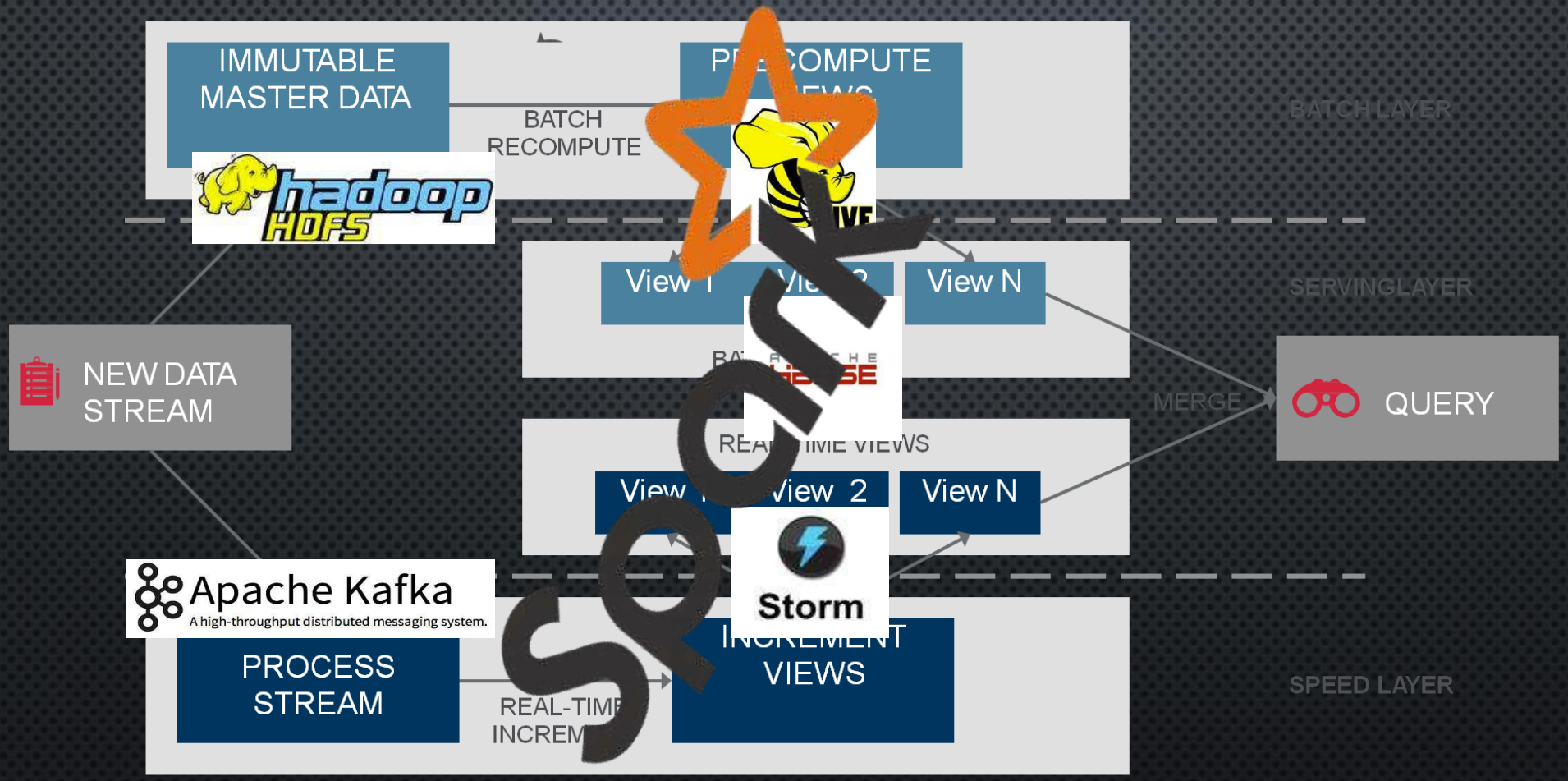
Benefits of Unification

- No need for copying or ETL of data between systems
- Combines processing types in one program
- Code reuse
- One system to learn
- One system to maintain

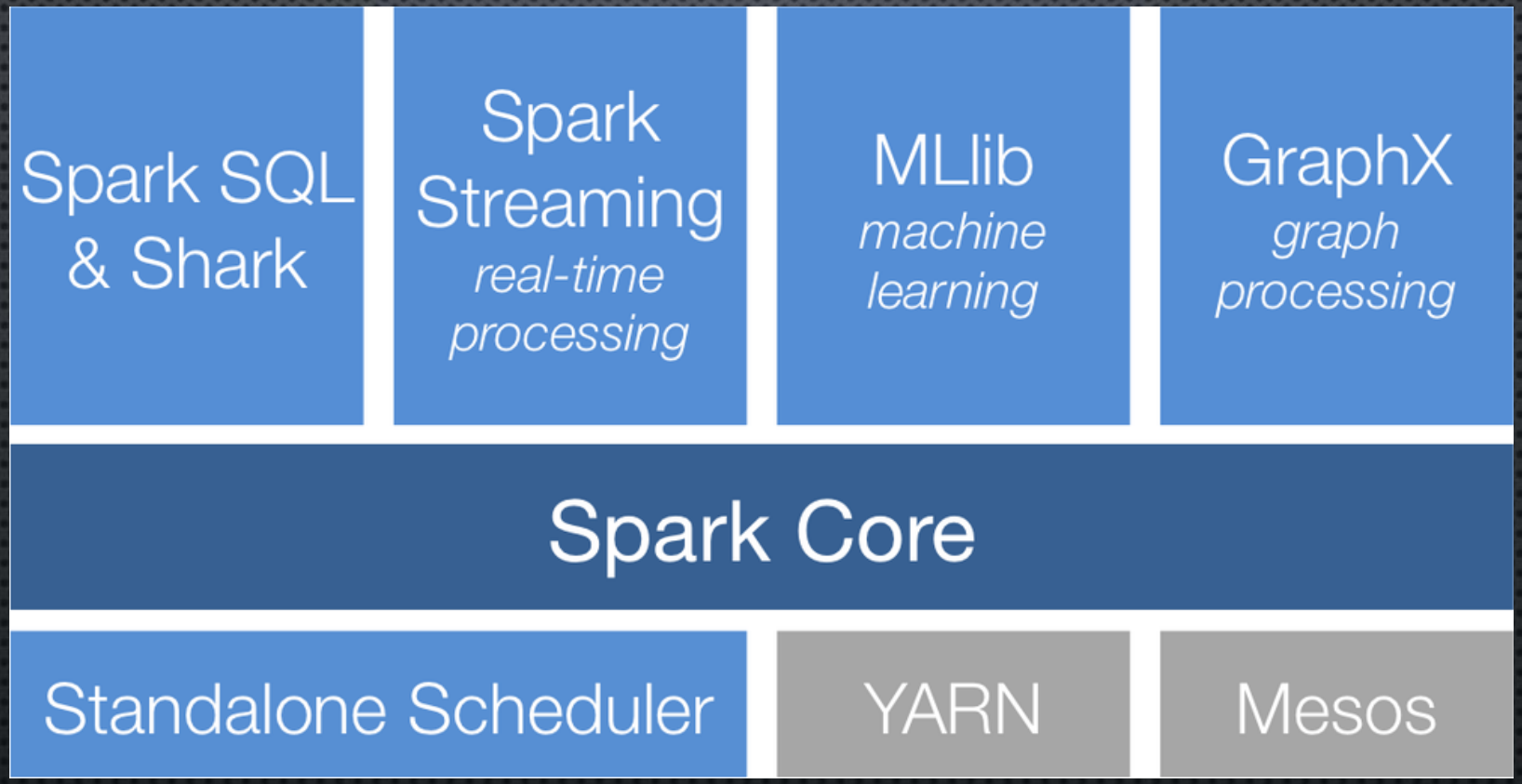
Benefits of Unification : Code Size



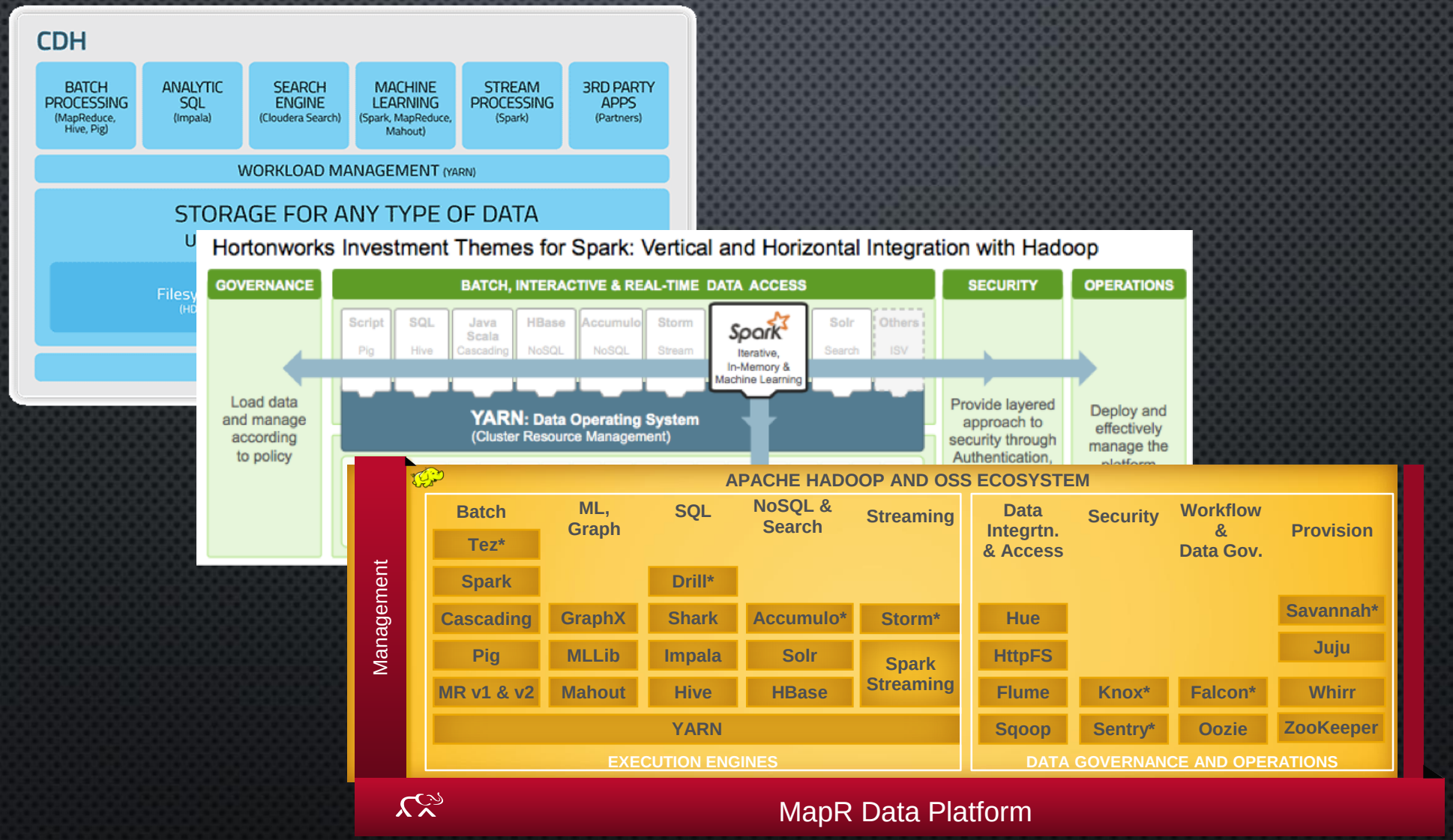
Building Lambda Architecture with Spark



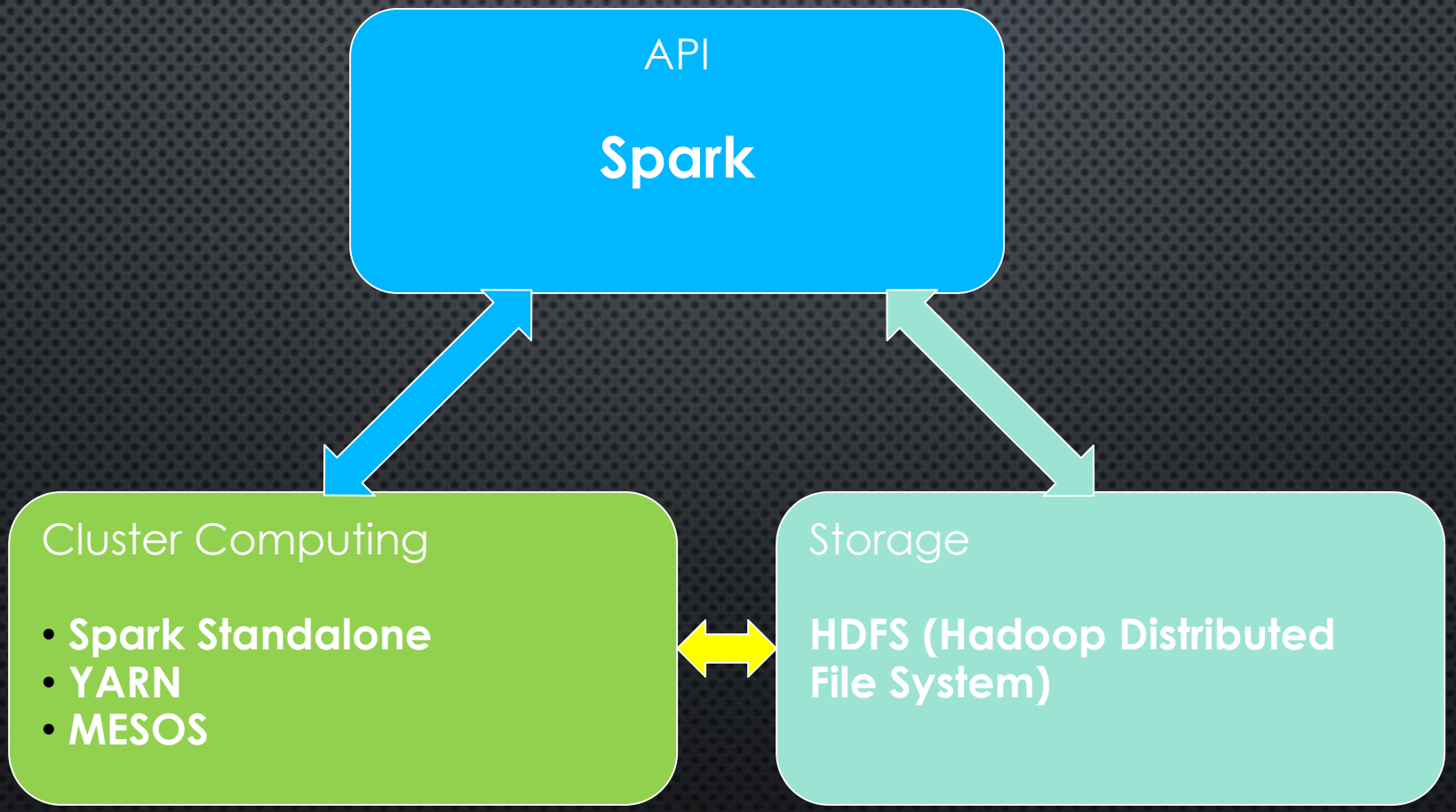
Spark Deployment Options



Spark on Hadoop Distributions



Distributed processing framework with Spark



Spark Advantages

- Easier APIs
- Python, Scala, Java

EASE OF DEVELOPMENT

IN-MEMORY PERFORMANCE

- RDDs
- DAGs Unify Processing

- SQL, ML, Streaming, Graph and Batch

COMBINE WORKFLOWS

Hadoop Advantages

UNLIMITED SCALE

- Multiple data sources
- Multiple applications
- Multiple users

ENTERPRISE PLATFORM

- Reliability
- Multi-tenancy
- Security

WIDE RANGE OF APPLICATIONS

- Files
- Databases
- Semi-structured

Hadoop + Spark Advantages

UNLIMITED SCALE

EASE OF DEVELOPMENT

IN-MEMORY PERFORMANCE

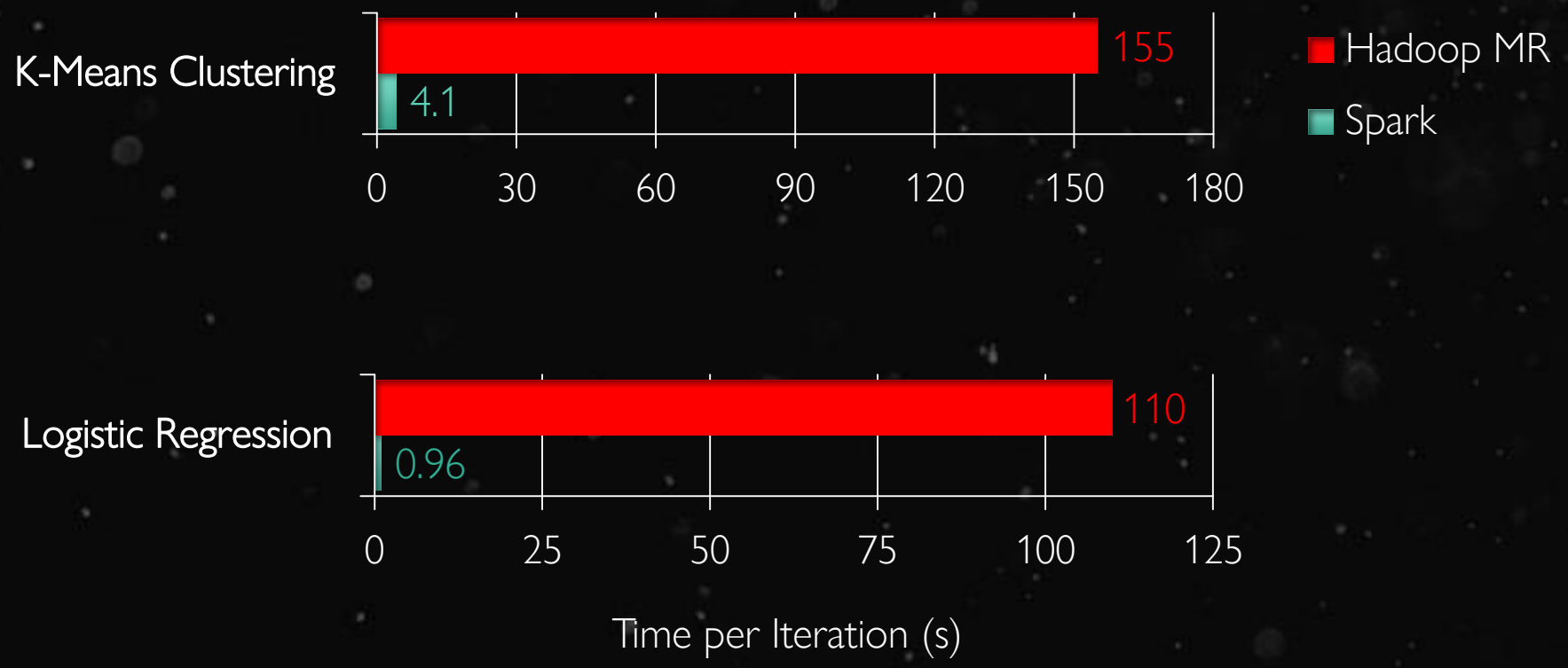
ENTERPRISE PLATFORM

WIDE RANGE OF APPLICATIONS

COMBINE WORKFLOWS

Operational Applications Augmented by In-Memory Performance

SPARK IS IN-MEMORY AND FAST



SPARK IS IN-MEMORY AND FAST

2013 Record: Hadoop

2100 machines

72 minutes

2014 Record: Spark

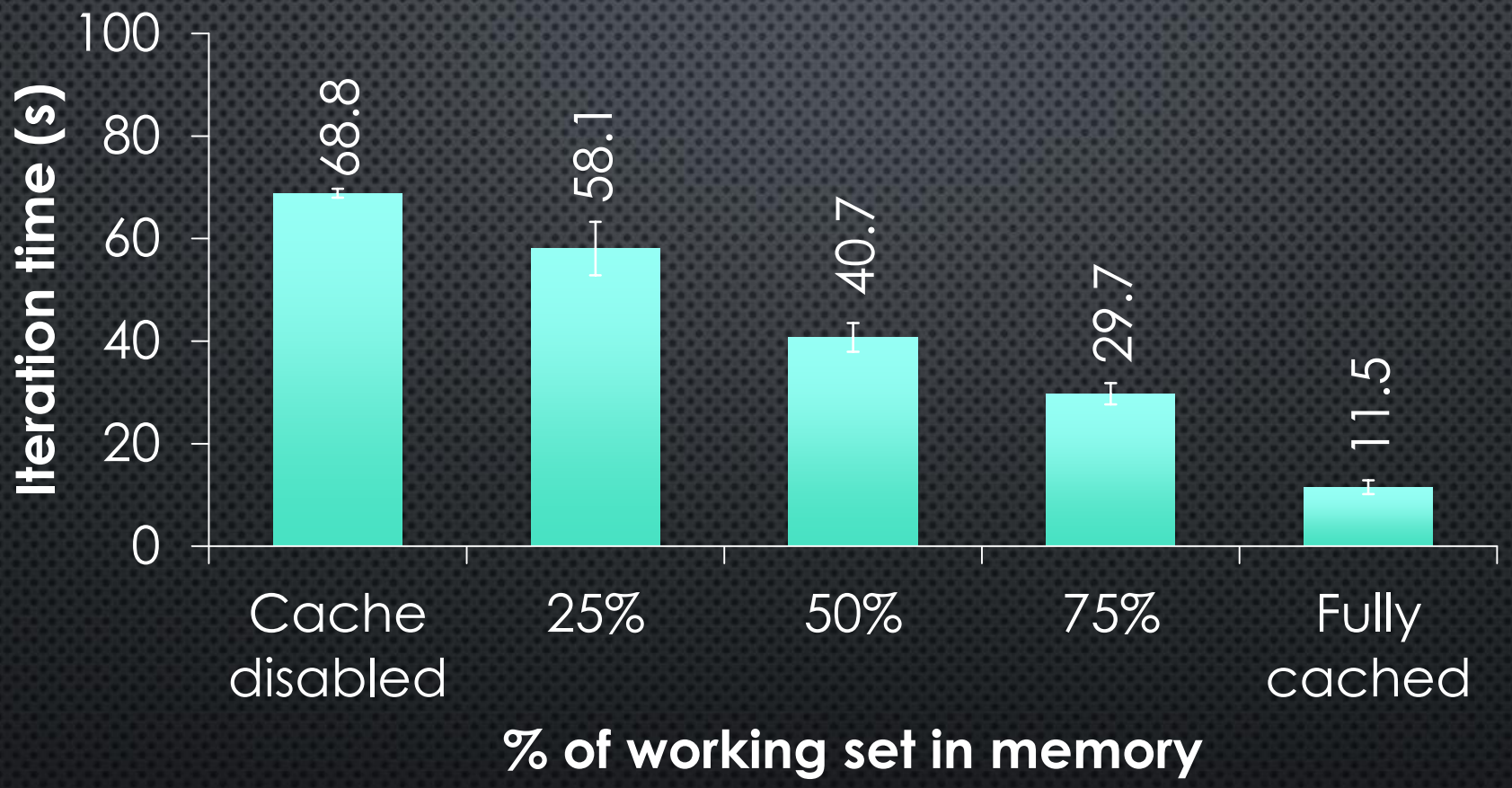
207 machines

23 minutes

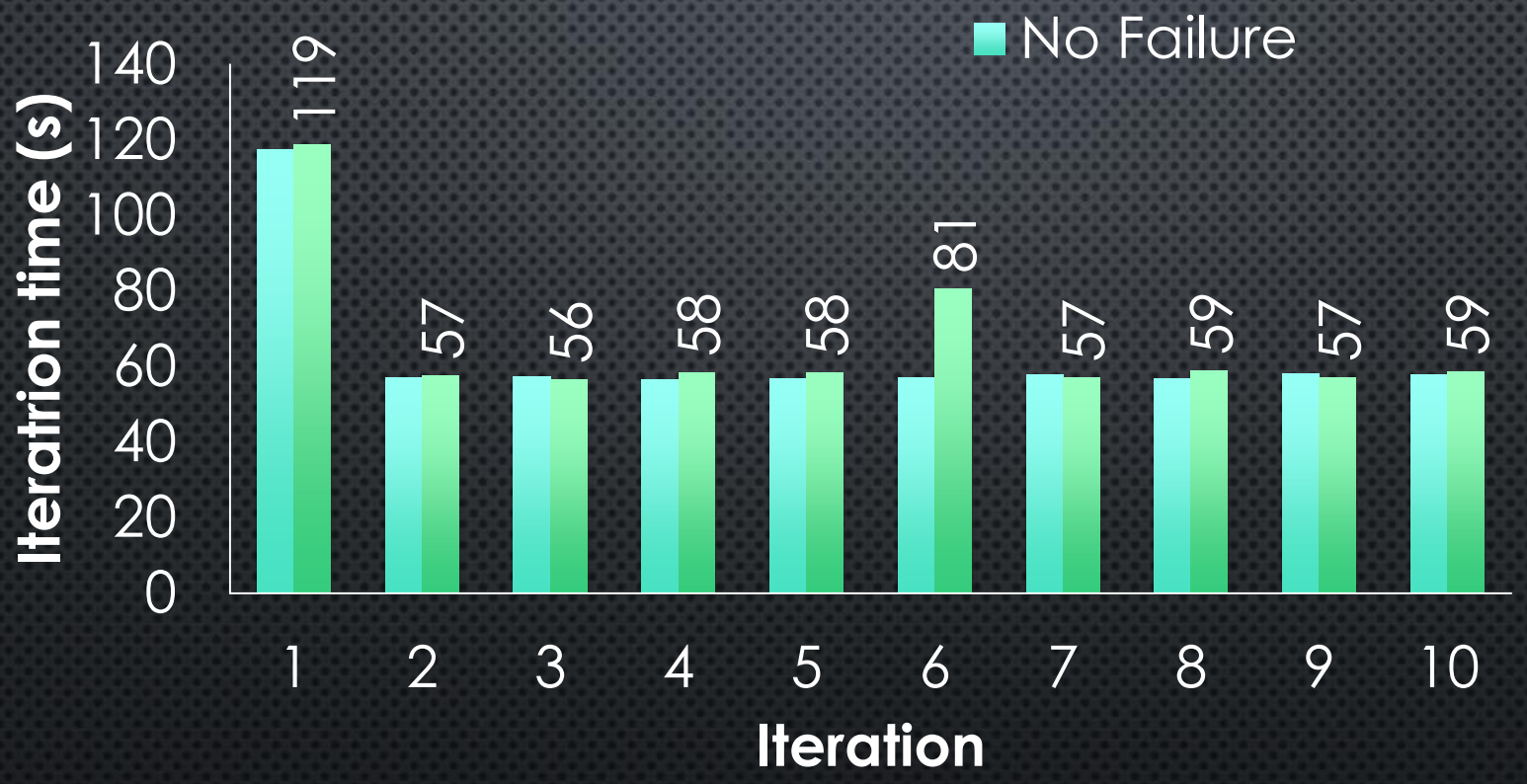
Also sorted 1PB in 4 hours

Source: Daytona GraySort benchmark, sortbenchmark.org

Not Enough RAM? – No Problem



Fault Recovery Results



Resources

Download and Documentation: https://spark.apache.org/

Slides/Videos: http://spark-summit.org/2013 http://spark-summit.org/2014

Source Code and examples : https://github.com/apache/spark/

Books: Learning Spark, Spark in Action and Fast Data Processing with Spark

Databricks Developer Resources: https://databricks.com/spark/developer-resources

Mailing List: **user@spark.apache.org**

MOOCs: **https://databricks.com/blog/2014/12/02/announcing-two-spark-based-moocs.html**

Events and Meetups: **https://spark.apache.org/community.html**

Spark Packages

spark-packages.org

Spark Packages

Feedback Register a package Login Find a package

A community index of packages for Apache Spark.

33 packages

databricks/spark-avro

Integration utilities for using Spark with Apache Avro data

@pwendell / Latest release: 0.1 (11/27/14) / Apache-2.0 / (5)

3 sql 3 input 2 library

dibbhatt/kafka-spark-consumer

Low Level Kafka-Spark Consumer

@dibbhatt / No release yet / (3)

2 streaming 1 kafka

sigmoidanalytics/spork

Pig on Apache Spark

Spark Packages is a community site hosting modules that are not part of Apache Spark. Your use of and access to this site is subject to the terms of use.
Apache Spark and the Spark logo are trademarks of the Apache Software Foundation. This site is maintained as a community service by Databricks.

NEW

Conclusions

- Big data will be standard: everyone will have it
- Organizations will gain an edge through speed of action and sophistication of analysis
- Apache Spark brings these to Hadoop clusters