Apache Spark Tutorial

Bernhard Haslhofer
AIT - Austrian Institute of Technology

DIL breakfast talk
2016-04-21

Overview

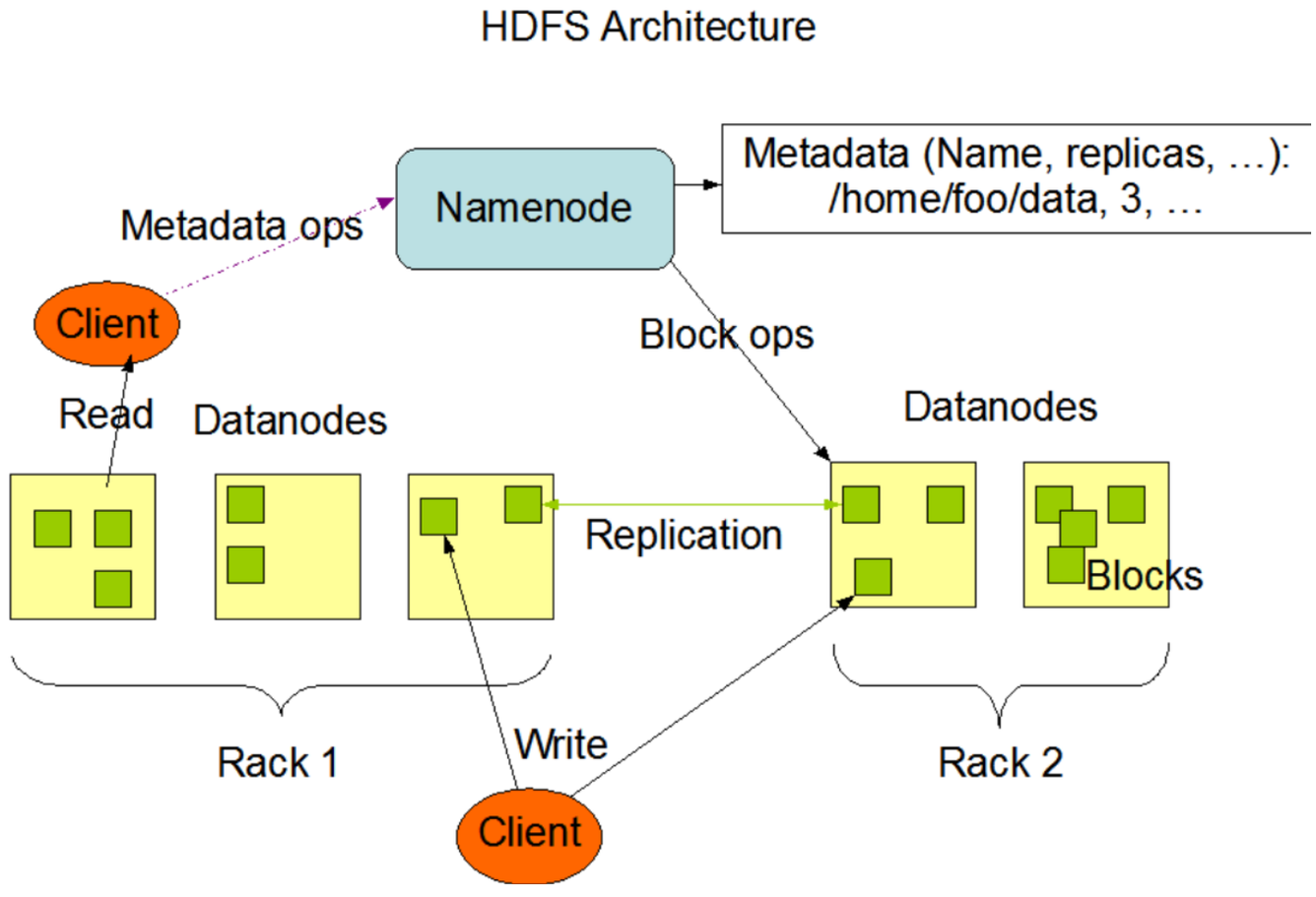
- What is Apache Spark?
- Common Operations
- SparkR - Bitcoin Demo
- Outlook and Discussion (cluster @ DIL)

What is Apache Spark?

- Technology used for Google file system (gs://…)
- Two key capabilities:
  - HDFS - Distributed Storage
  - MapReduce - Distributed Computing

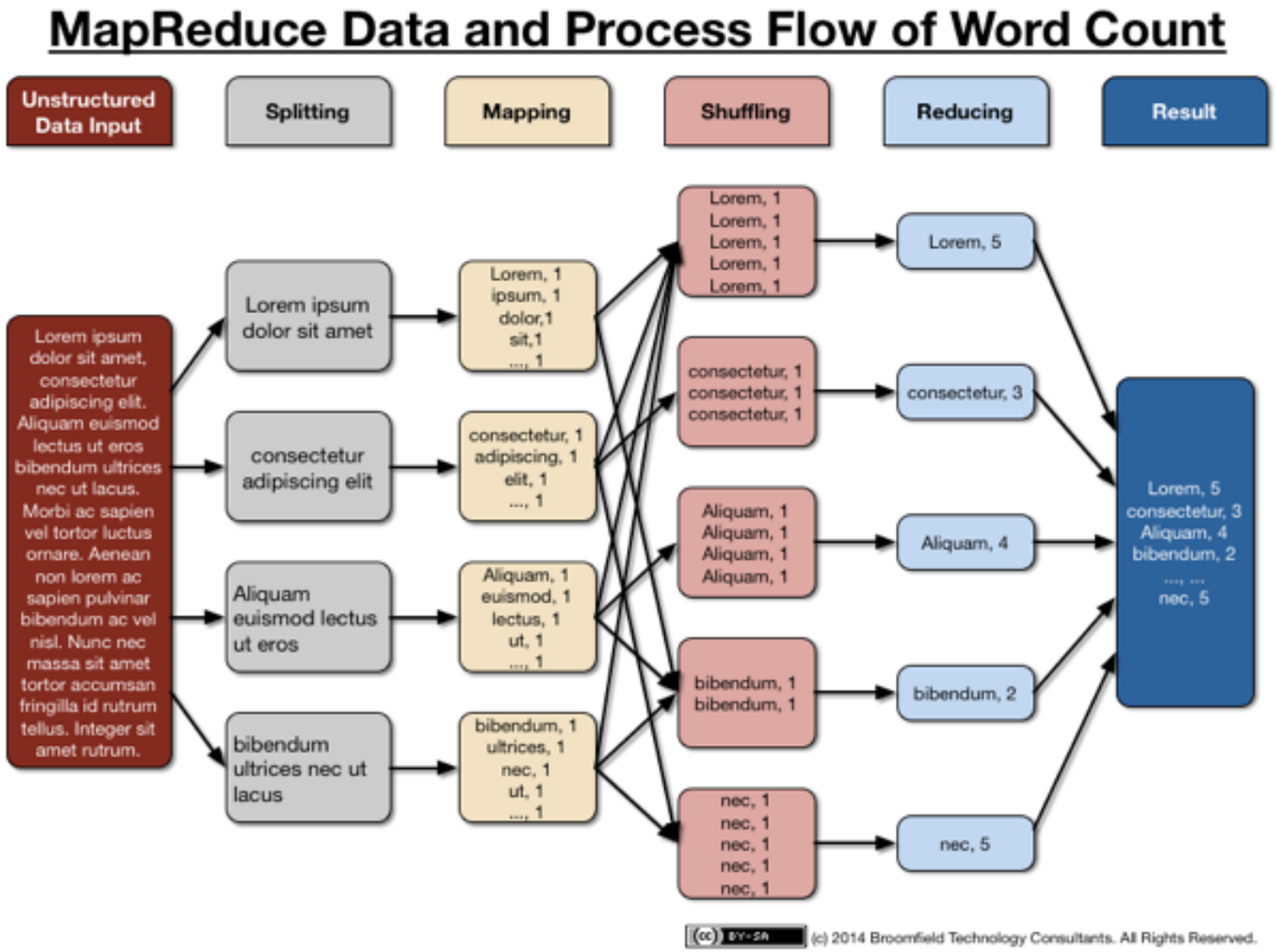
HDFS

- Distributed storage

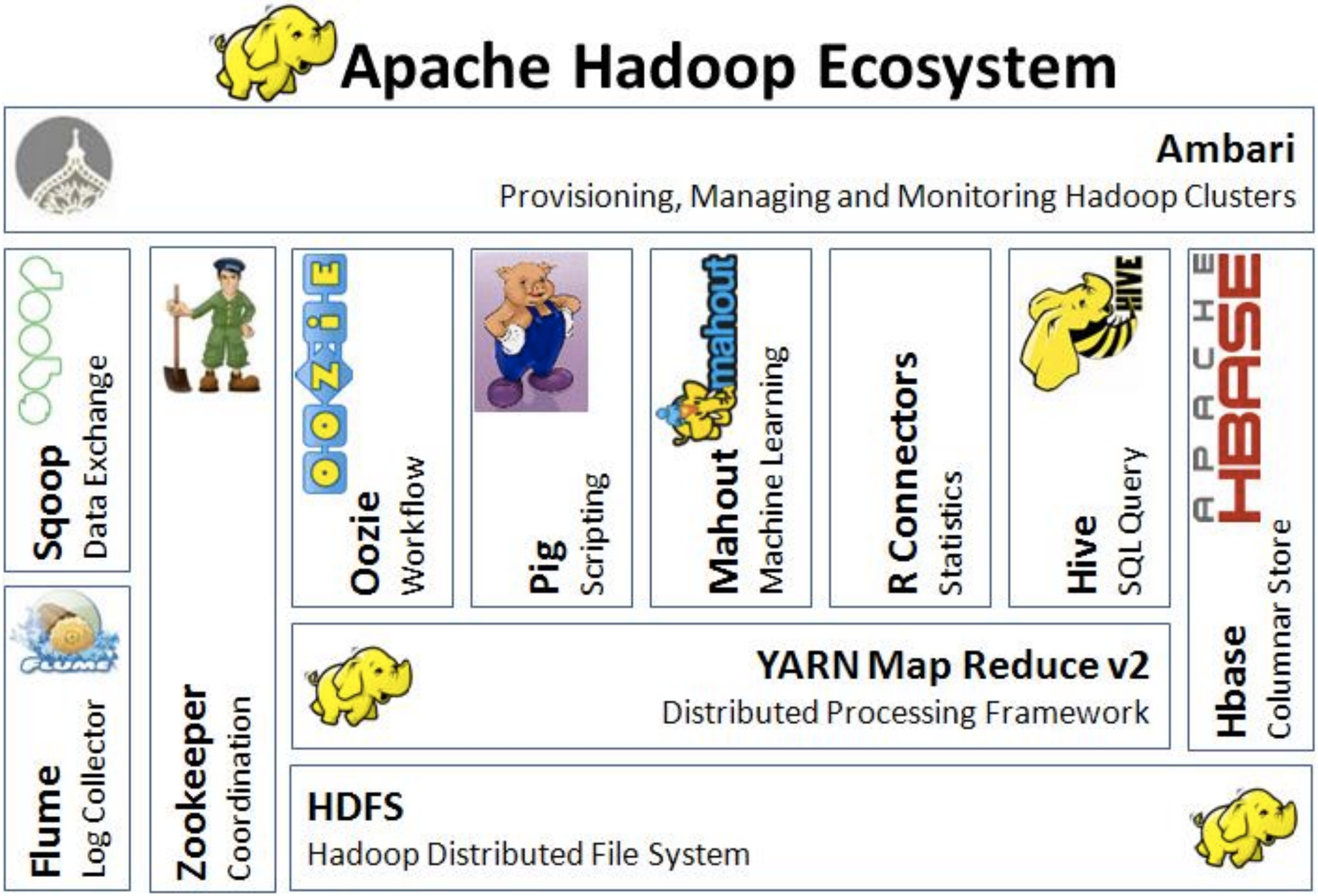


MapReduce

- Distributed compute



Ecosystem

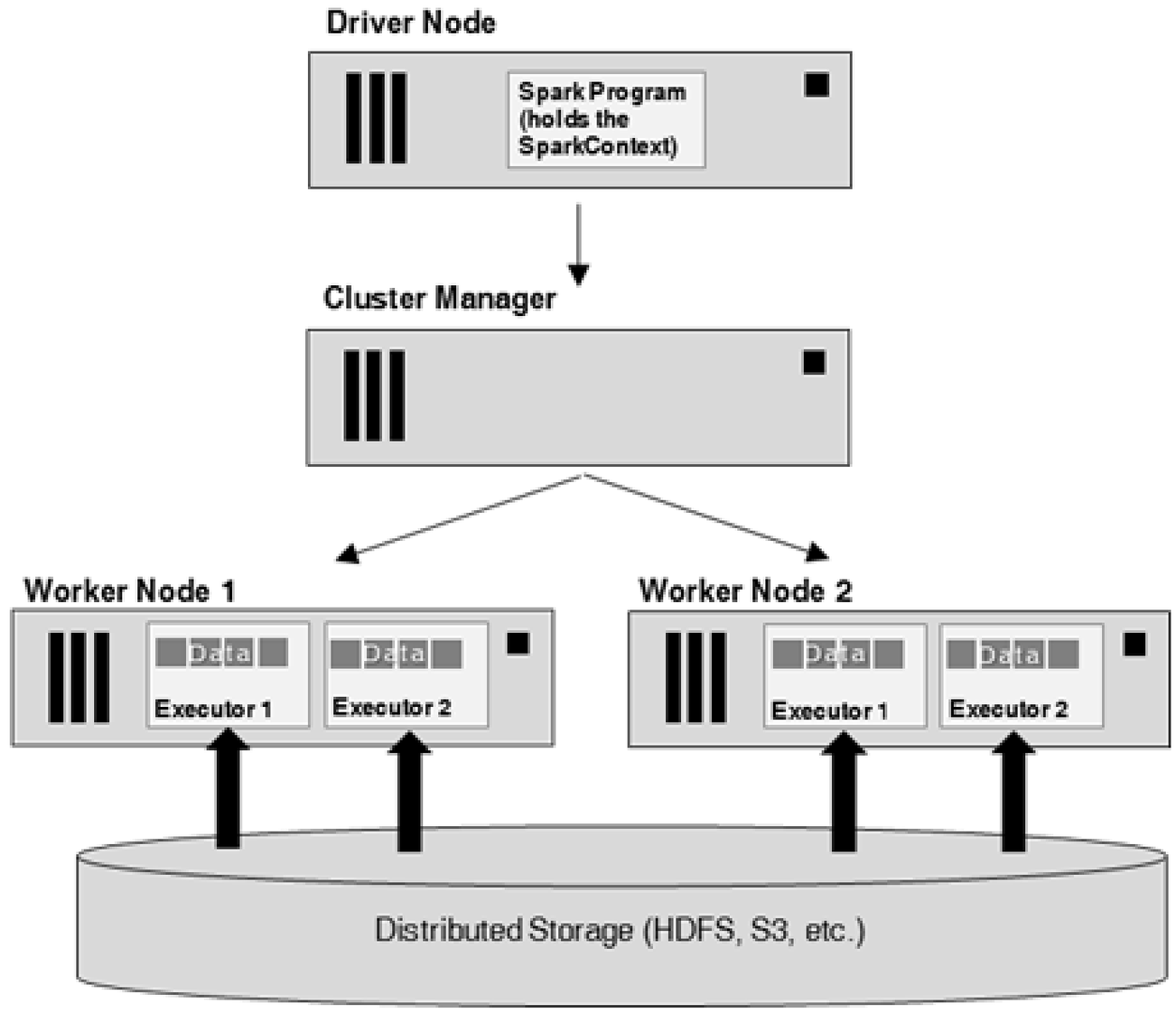

Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Flume
Log Collector
Zookeeper
Coordination
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

Hadoop Shortcomings

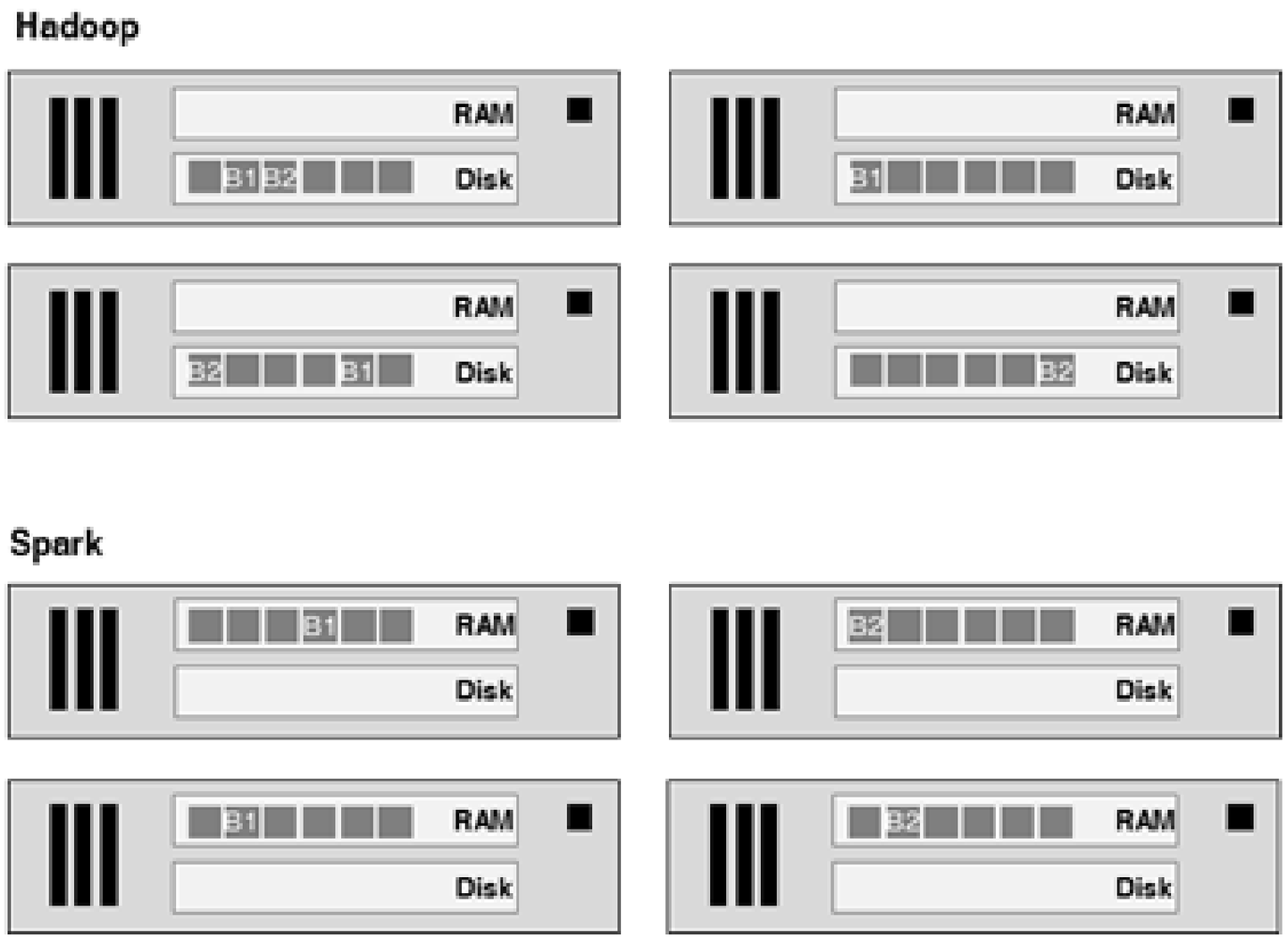
- Interactive querying
  - ask another question once we have an answer
  - requires reloading data from disk
- Iterative algorithms
  - e.g.: machine learning (Gradient descent)
  - requires series of MapReduce jobs
  - requires reloading data from disk

- Spark runs on a cluster of commodity hardware
- Key abstraction: **Resilient Distributed Dataset (RDD)**
  - distributed in-memory collections of elements
  - fault-tolerant
  - parallel operation

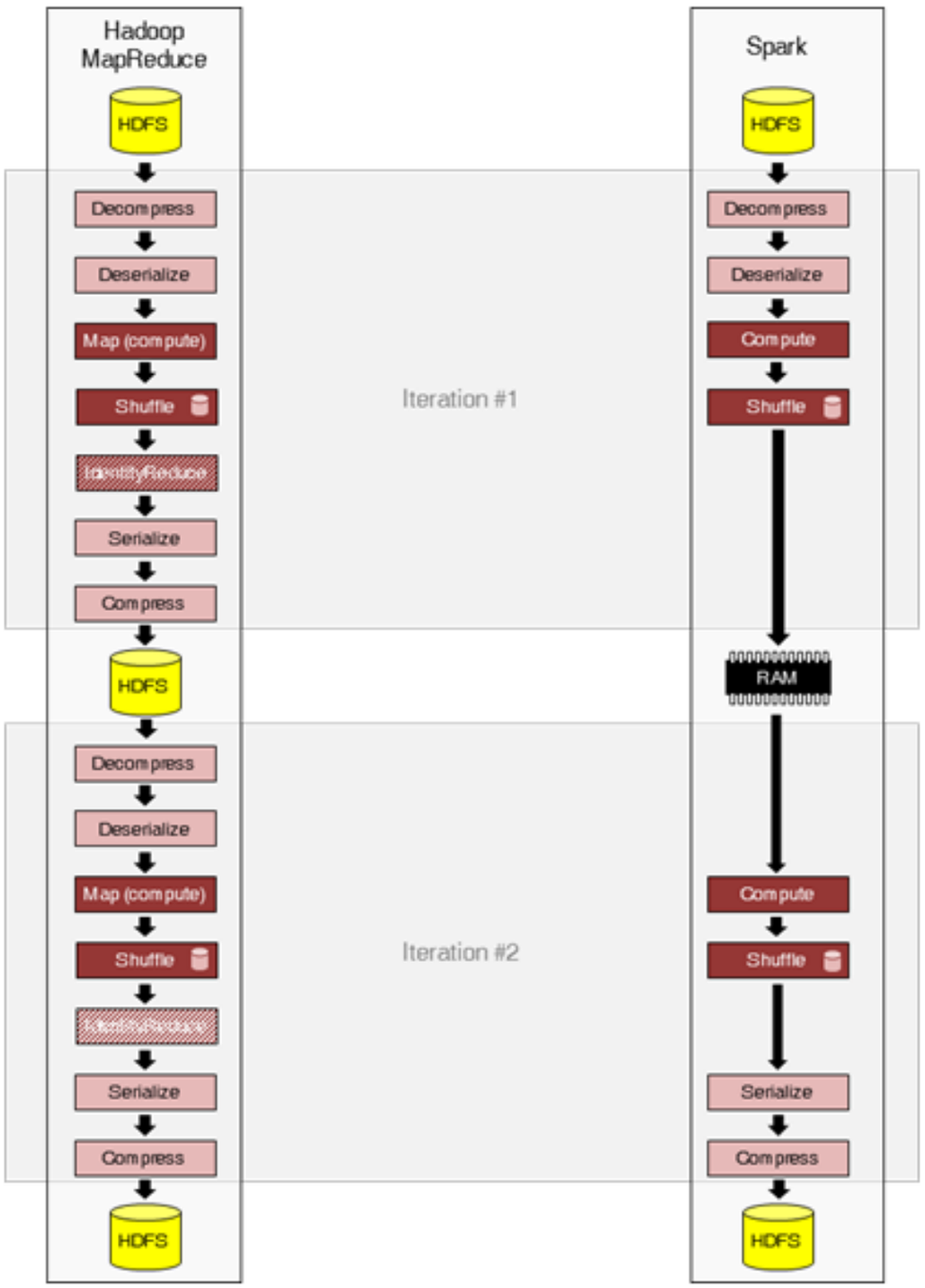
Architecture



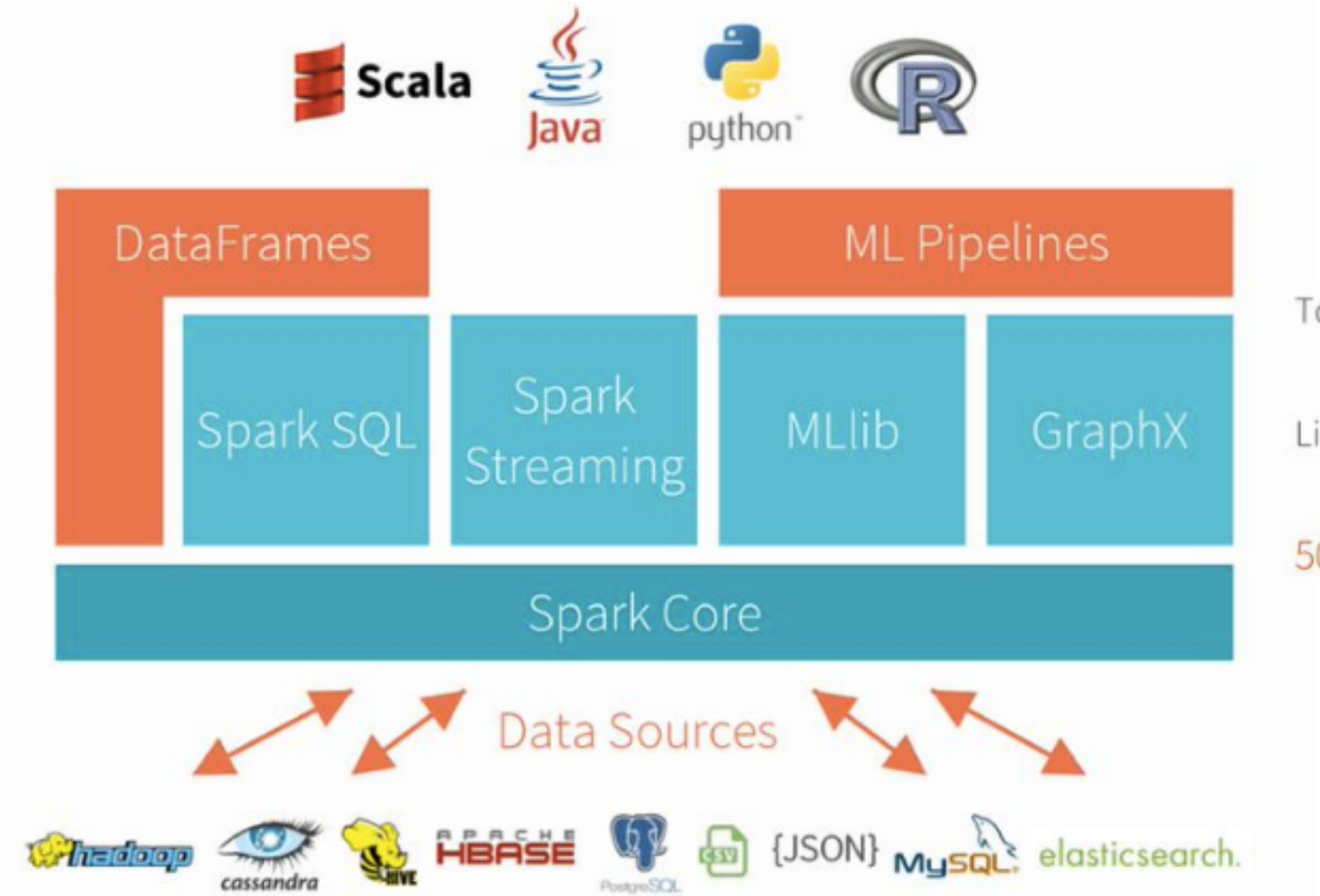
Diff Spark Hadoop



Diff Iterative Computing



Ecosystem



Total contributors: 150 → 500

Lines of code: 190K → 370K

500+ active production deployments

Common Operations

map / reduceByKey

```
val sentence = Array("A","soup", "can", "can", "can-can", ";", "can", "you", "?")

val r = sc.parallelize(sentence)

val m = r.map(x => (x, 1))

m.foreach(println(_))

val wordcount = m.reduceByKey((a, b) => a + b)

wordcount.foreach(println(_))
```

Compute term frequencies

map / reduce

```
wordcount.reduce((a,b) => if (a._2 > b._2) a else b)
```

Compute most frequent term

filter

```
r.filter(term => term.startsWith("c"))
```

Filter all terms starting with ‘c’

Transformations

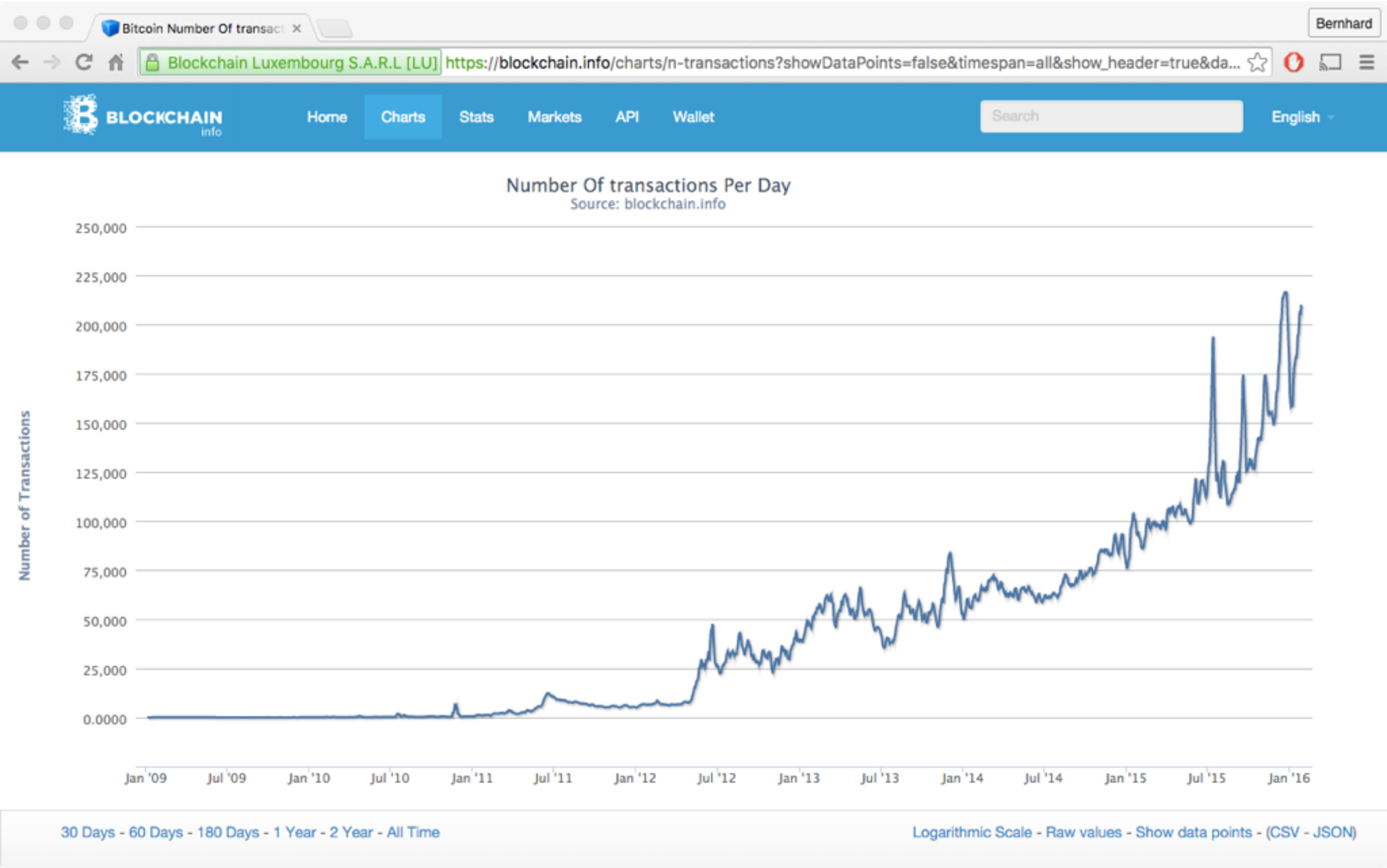
- map, flatMap, mapPartitions, mapPartitionsWithIndex
- filter, distinct
- sample
- union, intersection, cartesian
- groupByKey, aggregateByKey, join, cogroup
- coalesce, repartition
- sortByKey

Actions

- reduce
- collect
- count, countByKey
- first, take, takeSample, takeOrdered
- foreach
- saveAsTextFile, saveAsObjectFile

SparkR - Bitcoin Demo

Goal: plot such a figure…



Datasets

small blockchain dataset:

180 K blocks
3.14 M transactions

3 CSVs (~ 630 MB)

large blockchain dataset:

380 K blocks
95 M transactions

3 CSVs (~ 17 GB)

Data Structure

blocks.csv

|- block_hash (String)
|- height (Integer)
|- timestamp (Integer)

transactions.csv

|- tx_hash (String)
|- is_coinbase (Boolan)

rel_blocks_tx.csv

|- block_hash (String)
|- tx_hash (String)

Scenario

- Aggregate number of transactions per day and plot using standard R

- Aggregate number of transactions per day and plot using SparkR locally (single node)

- Aggregate number of transactions per day and plot using SparkR on a cluster

Outlook and Discussion

Status

- Latest release: Apache Spark 1.6.1
  - Apache Spark 2.0 expected in May 2016
- GraphX scalability issues for certain algorithms
  - connected components
  - Spark GraphFrames API coming soon