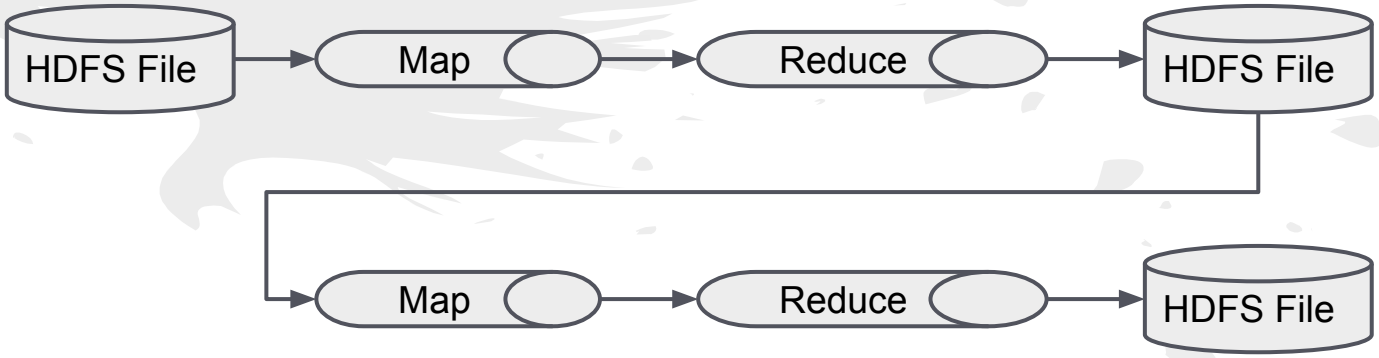
Introducing Apache Spark

Yuriy Taras

Agenda

- Why?
- Overview of Spark and it’s components
- Couple of Demos
- Q/A

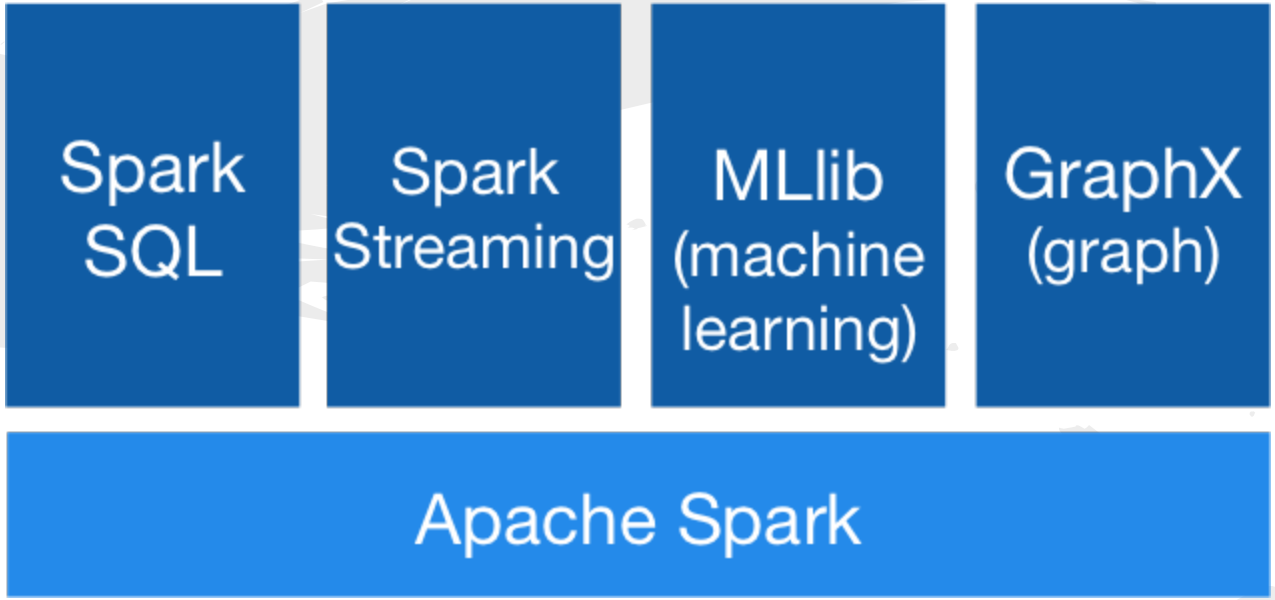
MapReduce



MapReduce paradigm flaws

- Counterintuitive
- Slow because of disk IO
- Imperative, not declarative

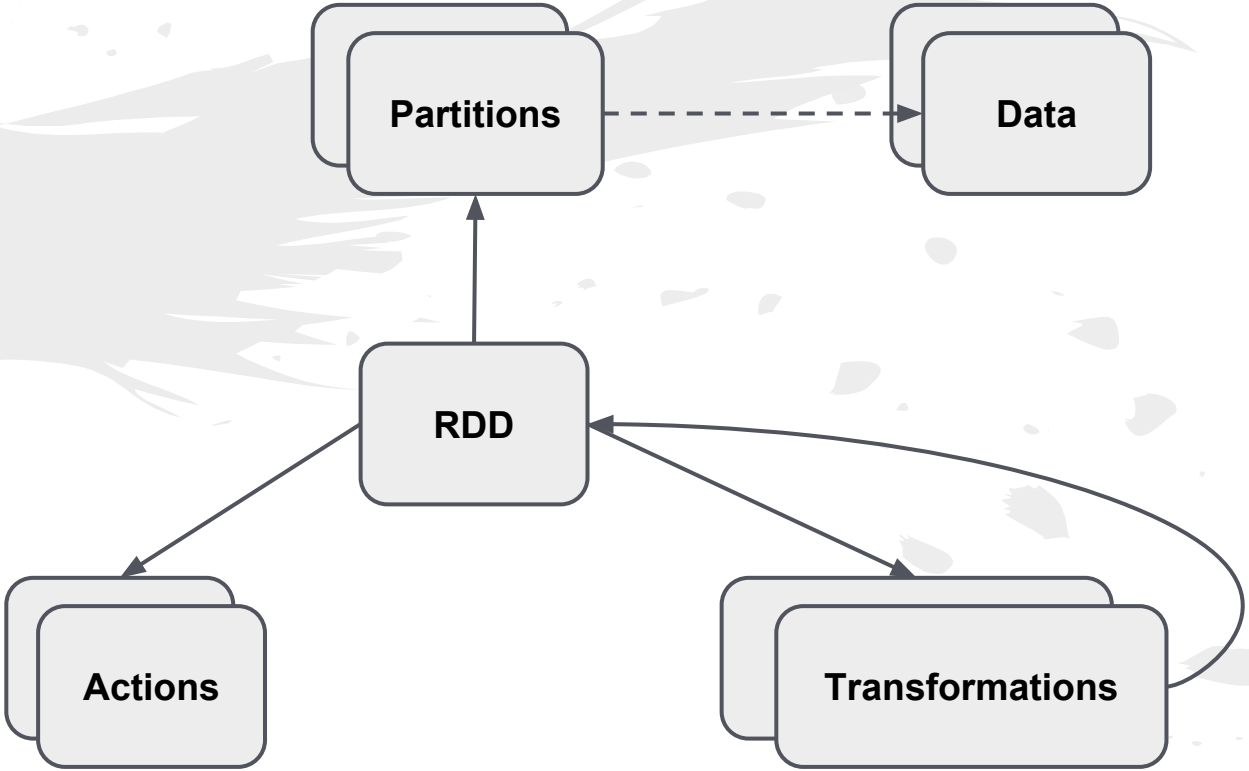
Apache Spark



Resilient Distributed Datasets

- Collections of objects spread across a cluster, stored in RAM or on Disk
- Built through parallel transformations
- Automatically rebuilt on failure

RDD - interesting parts



Available sources

Parallelized collections

Local file system

HDFS

Cassandra

HBase

Amazon S3

Transformations

map(*func*), **filter**(*func*), **flatMap**(*func*), **mapPartitions**(*func*), **mapPartitionsWithIndex**(*func*),

sample(*withReplacement*, *fraction*, *seed*), **union**(*otherDataset*), **intersection**(*otherDataset*),

distinct([*numTasks*])), **groupByKey**([*numTasks*]), **reduceByKey**(*func*, [*numTasks*]),

aggregateByKey(*zeroValue*)(*seqOp*, *combOp*, [*numTasks*]), **sortByKey**([*ascending*], [*numTasks*]),

join(*otherDataset*, [*numTasks*]), **cogroup**(*otherDataset*, [*numTasks*]), **cartesian**(*otherDataset*),
pipe(*command*, *[envVars]*),

coalesce(*numPartitions*), **repartition**(*numPartitions*), **repartitionAndSortWithinPartitions**

Actions

reduce(*func*), **collect**(), **count**(), **first**(), **take**(*n*), **takeSample**(*withReplacement*,*num*, [*seed*]),

takeOrdered(*n*, *[ordering]*), **saveAsTextFile**(*path*), **saveAsSequenceFile**(*path*),

saveAsObjectFile(*path*), **countByKey**(), **foreach**(*func*)

DEMO

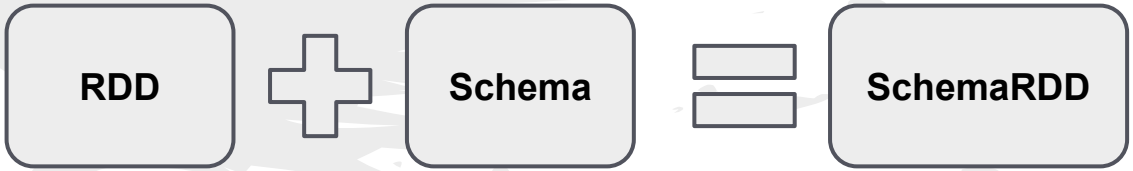
Spark SQL

Goals:

Provide ability to use SQL-like language on RDD

Loosely based on Hive

SchemaRDD




python
Scala
Java
HIVE QL
SQL-92
Spark SQL
SchemaRDD
Parquet
{JSON}
HIVE
cassandra

DEMO

Spark Streaming

Goals

Process large streams of data in realtime

Integrate with batch processing, use same programming model and computing capacity

Processing model

Example

```
val tweets  = TwitterUtils
  .createStream(ssc, None)
val hashTags =  tweets
  .flatMap(status =>
        getTags(status))
hashTags
  .saveAsHadoopFiles("hdfs://...")
```

Machine learning

Spark provides number of ML algorithms as part of MLib library.

classification: logistic regression, linear support vector machines (SVM), naive Bayes, decision trees

regression: linear regression, regression trees
collaborative filtering: alternating least squares (ALS)

clustering: k-means

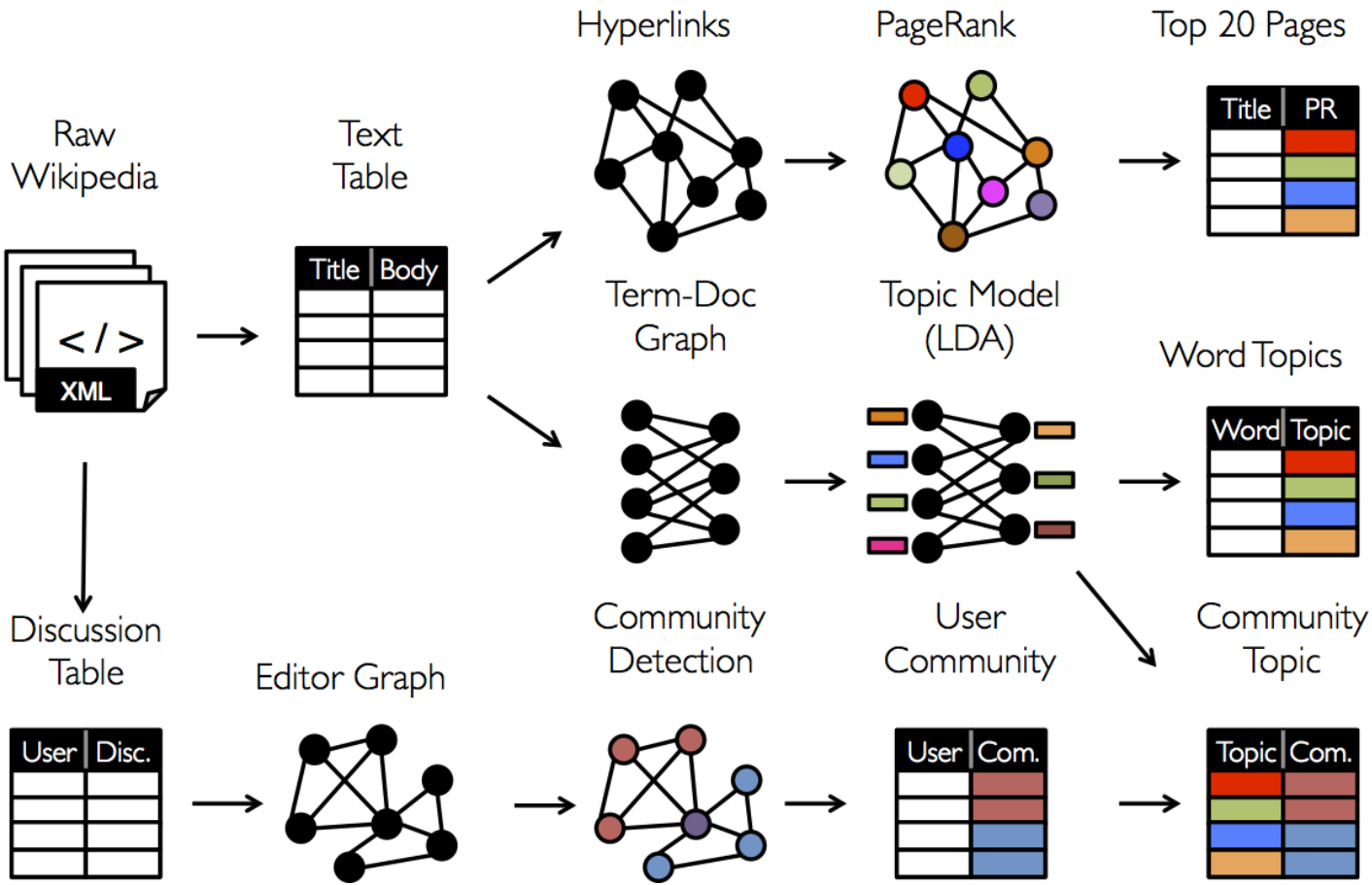
optimization: stochastic gradient descent (SGD), limitedmemory BFGS (L-BFGS)

dimensionality reduction: singular value decomposition (SVD), principal component analysis (PCA)

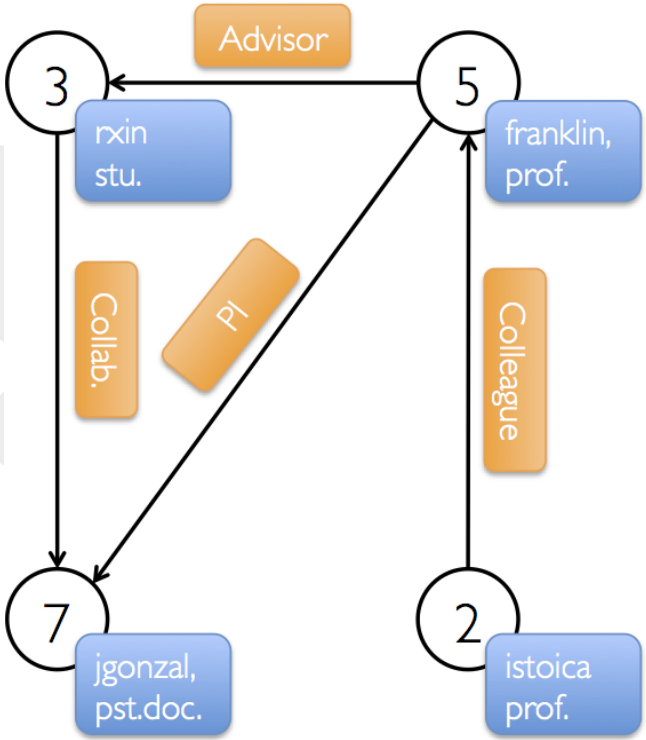
DEMO

GraphX

Provides distributed graph processing on top of Apache Spark. Integrates with other parts of the framework.


Raw Wikipedia
XML
Text Table
Title
Body
Hyperlinks
Term-Doc Graph
PageRank
Topic Model (LDA)
Top 20 Pages
Title
PR
Word Topics
Word
Topic
Discussion Table
User
Disc.
Editor Graph
Community Detection
User Community
User
Com.
Community Topic
Topic
Com.

Property Graph



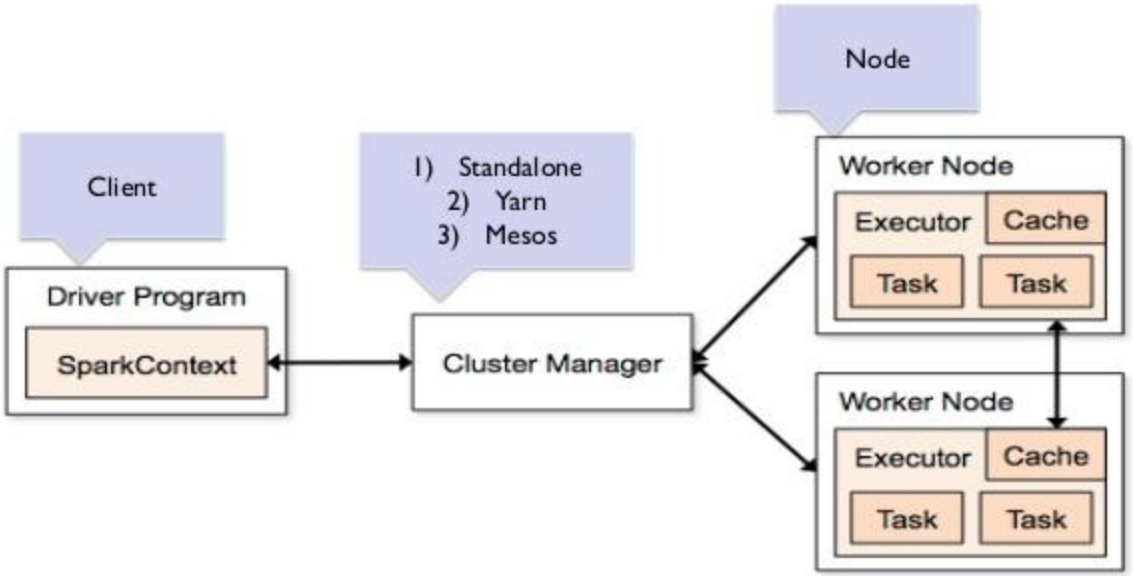
Vertex Table

Id	Property (V)
3	(rxin, student)
7	(jgonzal, postdoc)
5	(franklin, professor)
2	(istoica, professor)

Edge Table

SrcId	DstId	Property (E)
3	7	Collaborator
5	3	Advisor
2	5	Colleague
5	7	PI

Few words about deployment



Questions?