Real-Time Analytics with Spark Streaming

QCon São Paulo
2015-03-26

http://goo.gl/2M8uIf

Paco Nathan
@pacoid

Apache Spark, the elevator pitch

What is Spark?

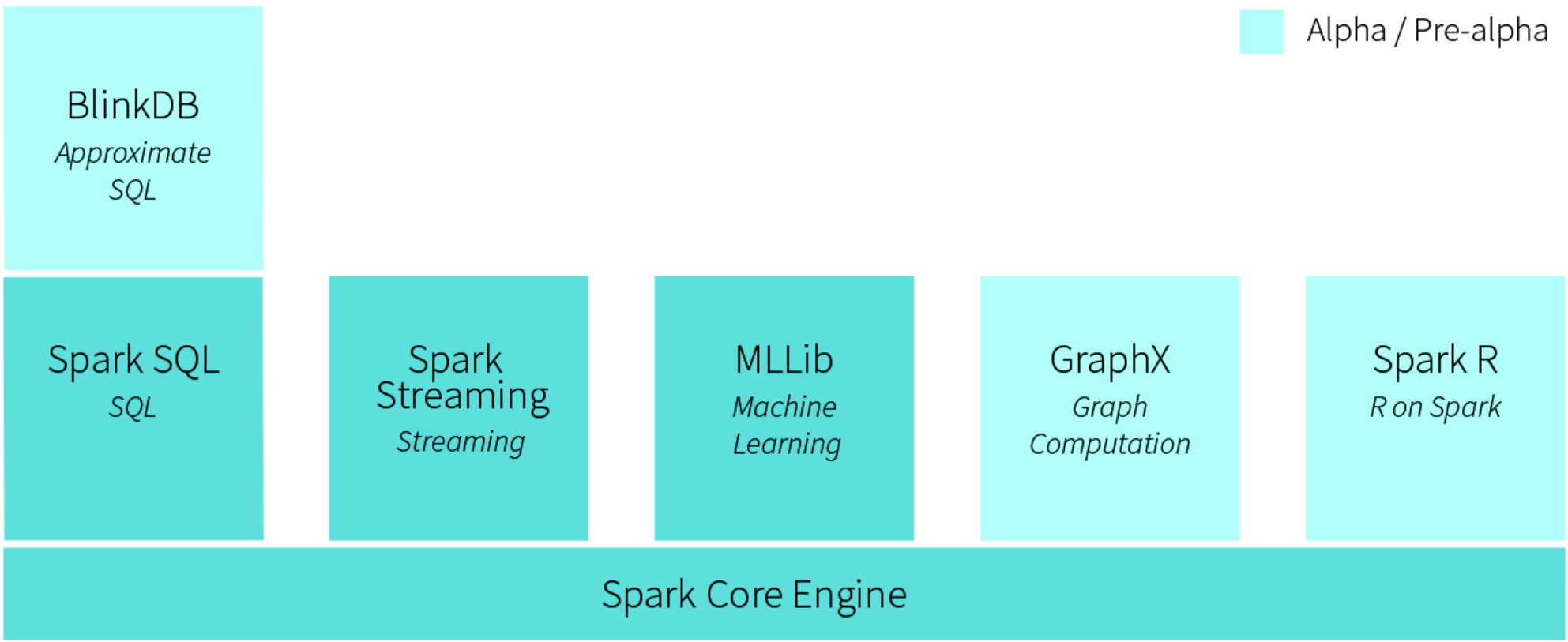
Developed in 2009 at UC Berkeley AMPLab, then open sourced in 2010, Spark has since become one of the largest OSS communities in big data, with over 200 contributors in 50+ organizations

"Organizations that are looking at big data challenges – including collection, ETL, storage, exploration and analytics – should consider Spark for its in-memory performance and the breadth of its model. It supports advanced analytics solutions on Hadoop clusters, including the iterative model required for machine learning and graph analysis."

Gartner, *Advanced Analytics and Data Science* (2014)

spark.apache.org

What is Spark?



What is Spark?

```
public class WordCount {
  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }

  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (otherArgs.length < 2) {
      System.err.println("Usage: wordcount <in> [<in>...] <out>");
      System.exit(2);
    }
    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    for (int i = 0; i < otherArgs.length - 1; ++i) {
      FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }
    FileOutputFormat.setOutputPath(job,
      new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

WordCount in 50+ lines of Java MR

```
val f = sc.textFile(inputPath)
val w = f.flatMap(l => l.split(" ")).map(word => (word, 1)).cache()
w.reduceByKey(_ + _).saveAsText(outputPath)
```

WordCount in 3 lines of Spark

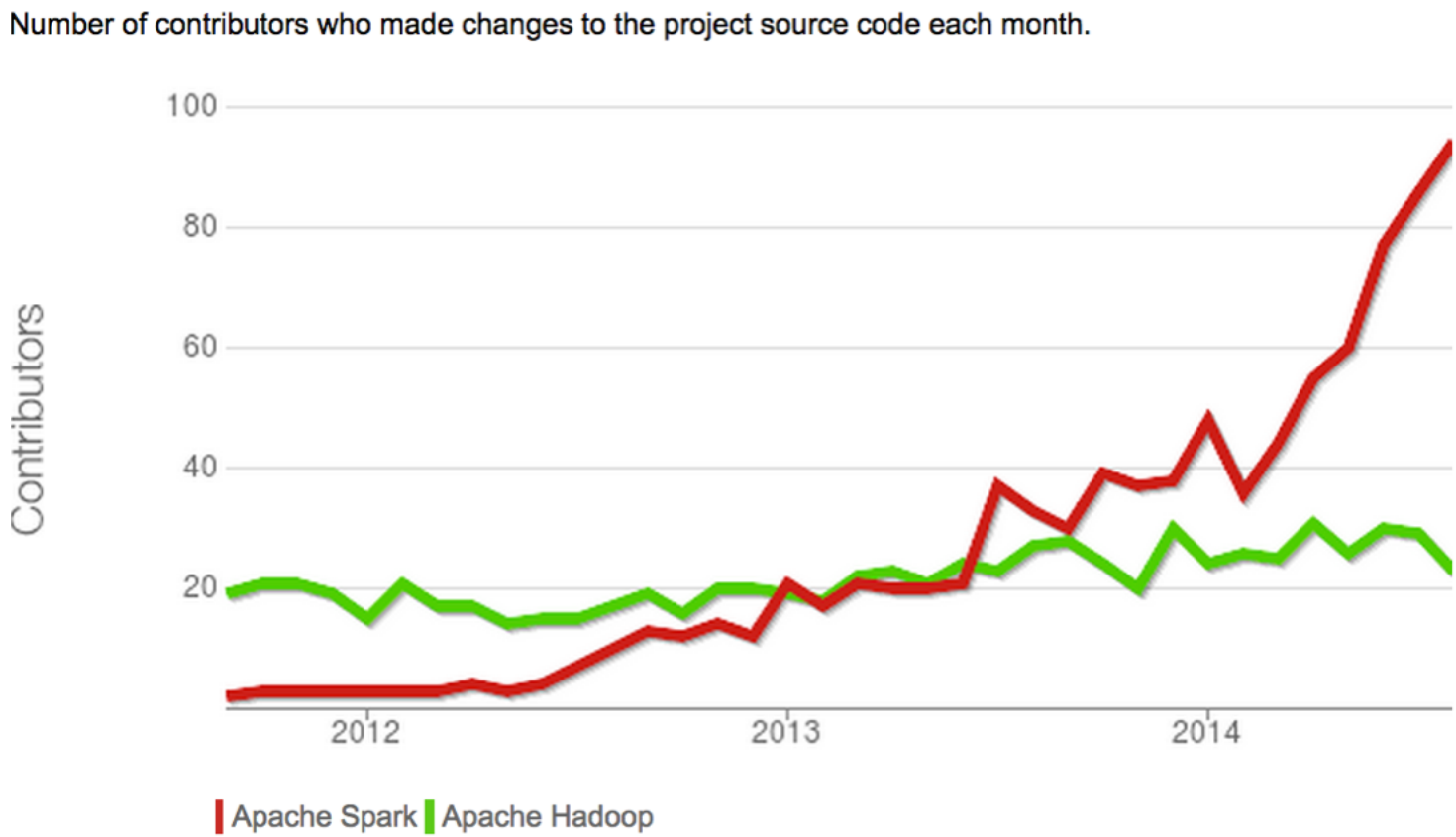
TL;DR: *Smashing The Previous Petabyte Sort Record*

databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html

	Hadoop MR Record	**Spark Record**	**Spark 1 PB**
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400 physical	6592 virtualized	6080 virtualized
Cluster disk throughput	3150 GB/s (est.)	618 GB/s	570 GB/s
Sort Benchmark Daytona Rules	Yes	Yes	No
Network	dedicated data center, 10Gbps	virtualized (EC2) 10Gbps network	virtualized (EC2) 10Gbps network
Sort rate	**1.42 TB/min**	**4.27 TB/min**	**4.27 TB/min**
Sort rate/node	**0.67 GB/min**	**20.7 GB/min**	**22.5 GB/min**

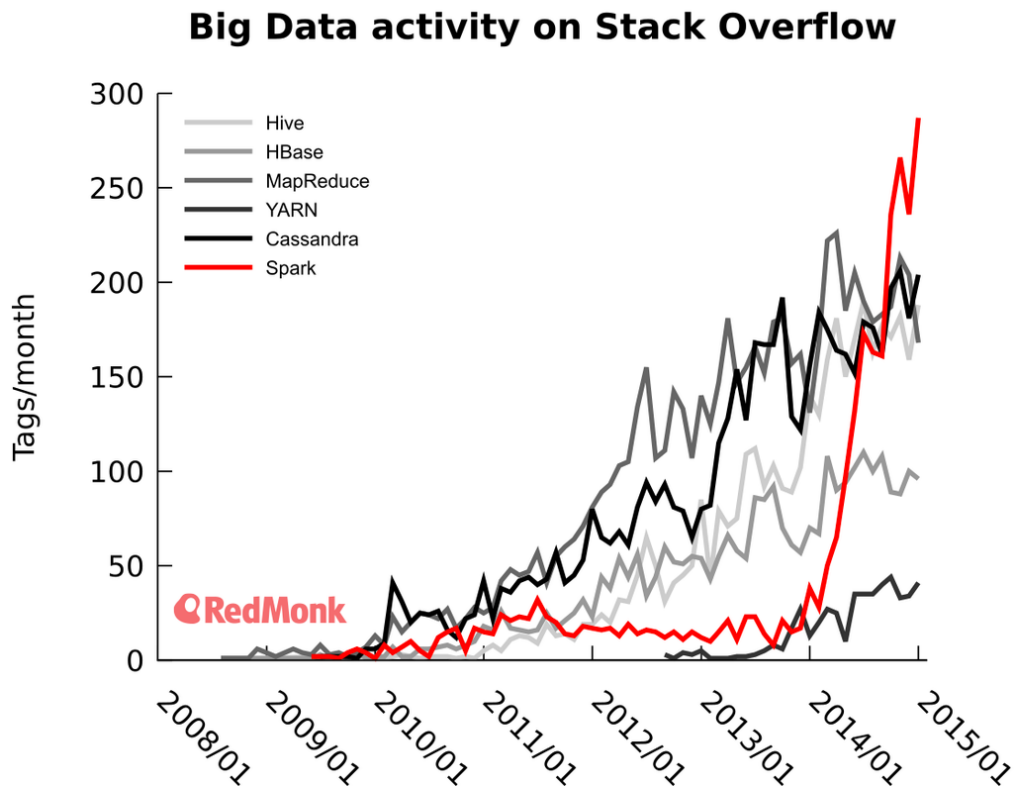
TL;DR: *Sustained Exponential Growth*

Spark is one of the most active Apache projects
ohloh.net/orgs/apache



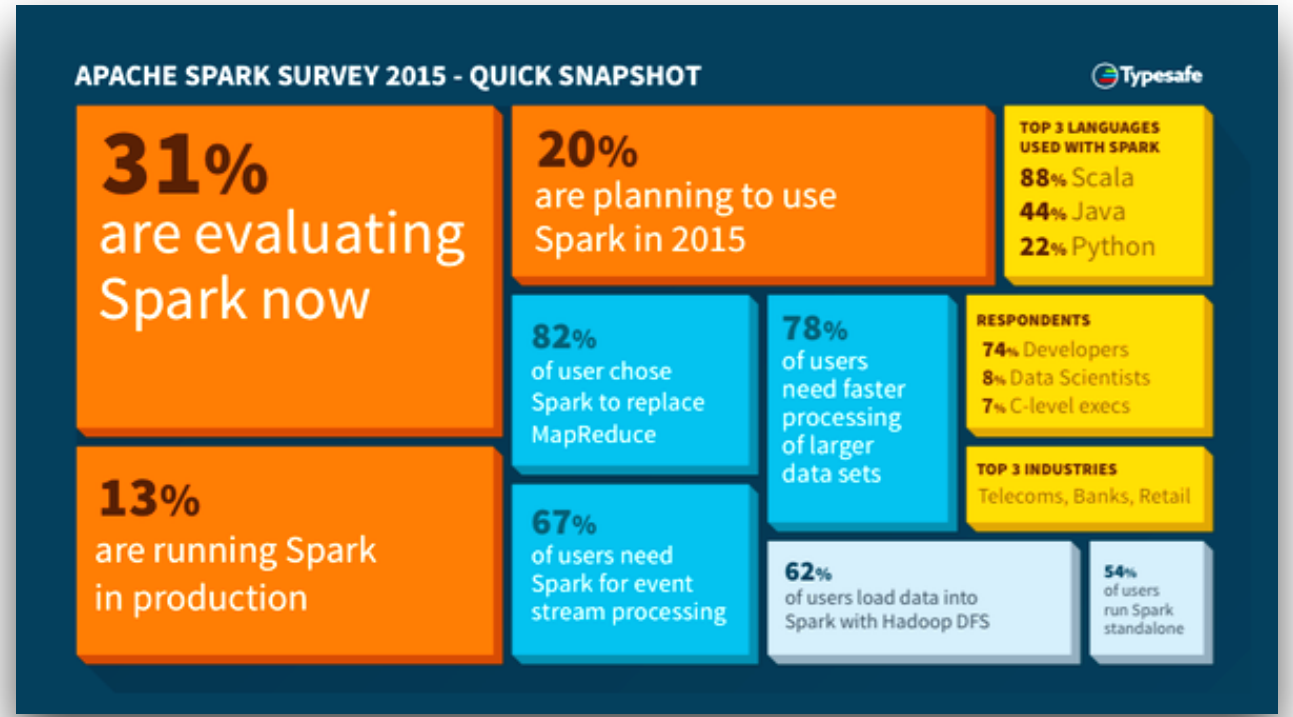
TL;DR: *Spark on StackOverflow*

twitter.com/dberkholz/status/568561792751771648



TL;DR: *Spark Survey 2015 by Databricks + Typesafe*

databricks.com/blog/2015/01/27/big-data-projects-are-hungry-for-simpler-and-more-powerful-tools-survey-validates-apache-spark-is-gaining-developer-traction.html



Streaming Analytics, the backstory

General Insights: *Business Drivers*

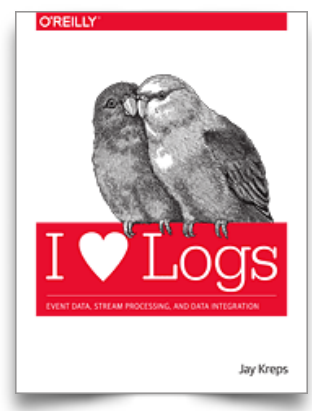
- batch windows were useful, but we need to obtain crucial insights faster and more globally
- probably don't need a huge cluster for real-time, but it'd be best to blend within a general cluster
- many use cases must consider data 2+ times: on the wire, subsequently as historical data
- former POV
  "secure data in DW, then OLAP ASAP afterward"
  gives way to current POV
  "analyze on the wire, write behind"

General Insights: *Use Cases*

early use cases: *finance*, *advertising*, *security*, *telecom*
– had similar risk/reward ratio for customers

transitioning to: *genomics*, *transportation*, *health care*, *industrial IoT*, *geospatial analytics*, *datacenter operations*, *education*, *video transcoding*, etc.
– large use cases

machine data keeps growing!



I <3 Logs
Jay Kreps
O'Reilly (2014)
shop.oreilly.com/product/0636920034339.do

General Insights: *Use Cases*

Because IoT! (*exabytes/day per sensor*)

bits.blogs.nytimes.com/2013/06/19/g-e-makes-the-machine-and-then-uses-sensors-to-listen-to-it/

Approach: *Complex Event Processing*

- 1990s R&D, mostly toward DB theory
- event correlation: query events co-located in time, geo, etc.
- RPC semantics
- relatively well-known in industry
- relatively heavyweight process

Approach: *Storm*

- Twitter, Yahoo!, etc., since 2011
- event processing, *at-least-once* semantics
- developers define topologies in terms of spouts and bolts – lots of coding required!
- abstraction layers (+ *complexity*, *overhead*, *state*)

  + **github.com/twitter/summingbird**
  + **storm.apache.org/documentation/Trident-tutorial.html**
  + **github.com/AirSage/Petrel**

Approach: *Micro-Batch*

Because Google!

MillWheel: Fault-Tolerant Stream Processing at Internet Scale
Tyler Akidau, **Alex Balikov**, **Kaya Bekiroglu**, **Slava Chernyak**, **Josh Haberman**, **Reuven Lax**, **Sam McVeety**, **Daniel Mills**, **Paul Nordstrom**, **Sam Whittle**
Very Large Data Bases (2013)
research.google.com/pubs/pub41378.html

Spark Streaming

Spark Streaming: *Requirements*

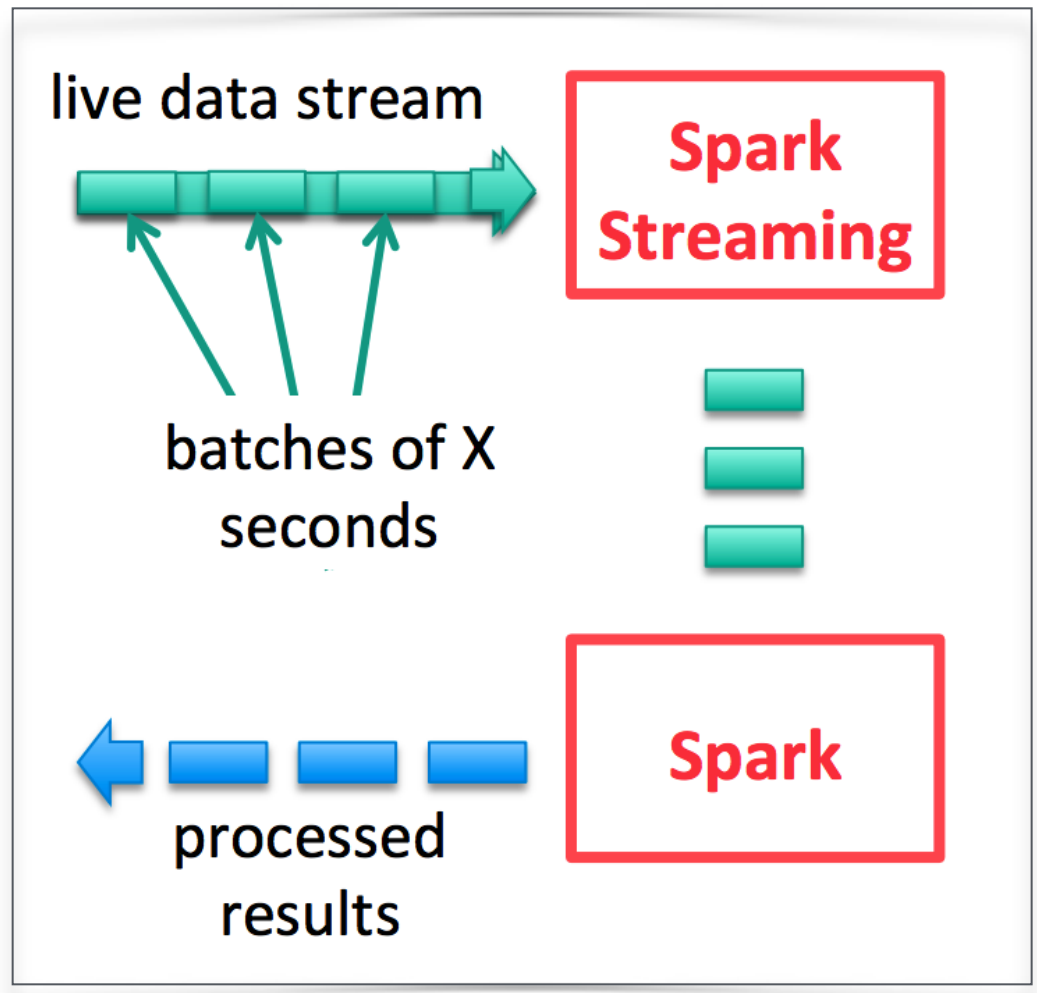
Consider our top-level requirements for a streaming framework:

- clusters scalable to 100's of nodes
- low-latency, in the range of seconds (meets 90% of use case needs)
- efficient recovery from failures (which is a hard problem in CS)
- integrates with batch: many co's run the same business logic both online+offline

Spark Streaming: *Requirements*

Therefore, run a streaming computation as:
a series of very small, deterministic batch jobs

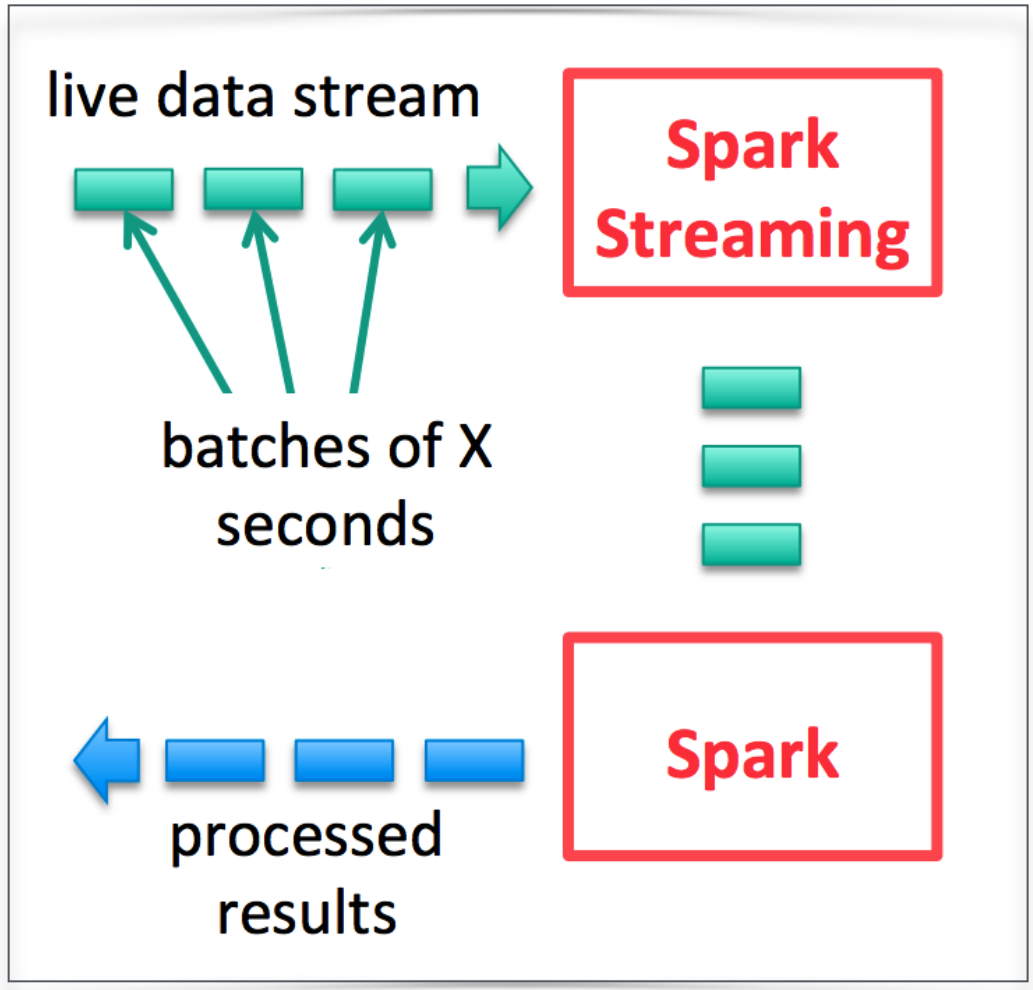
- *Chop up the live stream into batches of X seconds*
- *Spark treats each batch of data as RDDs and processes them using RDD operations*
- *Finally, the processed results of the RDD operations are returned in batches*



Spark Streaming: *Requirements*

Therefore, run a streaming computation as:
a series of very small, deterministic batch jobs

- *Batch sizes as low as ½ sec, latency of about 1 sec*
- *Potential for combining batch processing and streaming processing in the same system*



Spark Streaming: *Timeline*

2012 project started

2013 alpha release (Spark 0.7)

2014 graduated (Spark 0.9)

Discretized Streams: A Fault-Tolerant Model for Scalable Stream Processing
Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, Ion Stoica
Berkeley EECS (2012-12-14)
www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-259.pdf

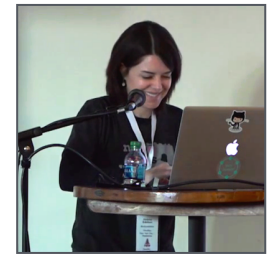
project lead:
Tathagata Das **@tathadas**

Spark Streaming: *Community – A Selection of Thought Leaders*

David Morales
Stratio
@dmoralesdf

Krishna Gade
Pinterest
@krishnagade

Claudiu Barbura
Atigeo
@claudiubarbura



Helena Edelson
DataStax
@helenaedelson

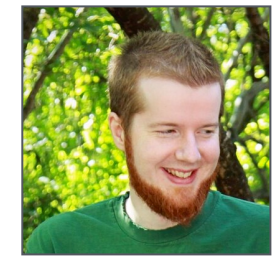
Eric Carr
Guavus
@guavus

Gerard Maas
Virdata
@maasg

Mayur Rustagi
Sigmoid Analytics
@mayur_rustagi

Russell Cardullo
Sharethrough
@russellcardullo

Jeremy Freeman
HHMI Janelia
@thefreemanlab



Cody Koeninger
Kixer
@CodyKoeninger

Antony Arokiasamy
Netflix
@aasamy

Dibyendu Bhattacharya
Pearson
@maasg

Mansour Raad
ESRI
@mraad

Spark Streaming: *Some Excellent Resources*

Programming Guide
spark.apache.org/docs/latest/streaming-programming-guide.html

Spark Streaming @Strata CA 2015
slideshare.net/databricks/spark-streaming-state-of-the-union-strata-san-jose-2015

Spark Reference Applications
databricks.gitbooks.io/databricks-spark-reference-applications/

Spark Streaming: *Quiz – Identify the code constructs…*

```
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext._

// create a StreamingContext with a SparkConf configuration
val ssc = new StreamingContext(sparkConf, Seconds(10))

// create a DStream that will connect to serverIP:serverPort
val lines = ssc.socketTextStream(serverIP, serverPort)

// split each line into words
val words = lines.flatMap(_.split(" "))

// count each word in each batch
val pairs = words.map(word => (word, 1))
val wordCounts = pairs.reduceByKey(_ + _)

// print a few of the counts to the console
wordCounts.print()

ssc.start()
ssc.awaitTermination()
```

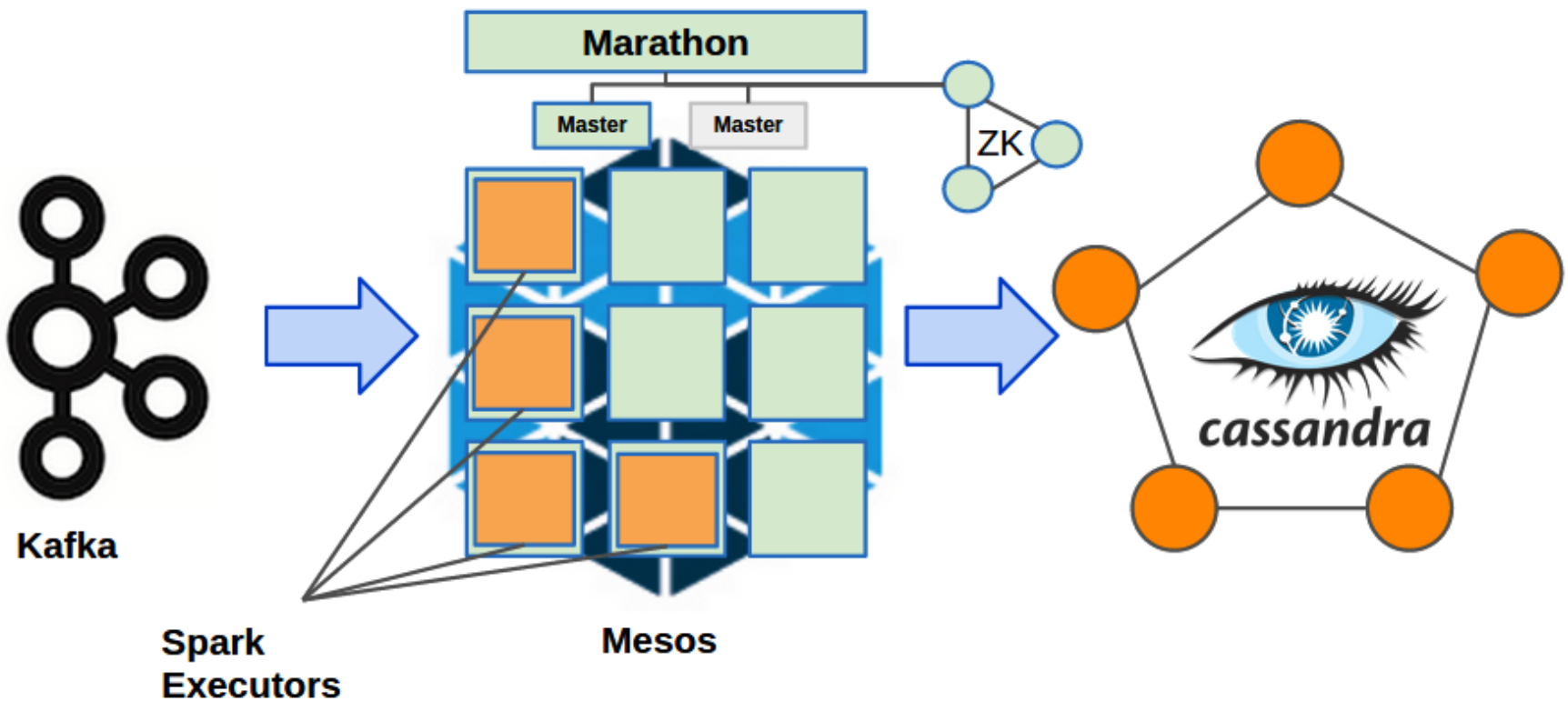
Details…

Tuning: *Virdata tutorial*

Tuning Spark Streaming for Throughput

Gerard Maas, 2014-12-22

virdata.com/tuning-spark/

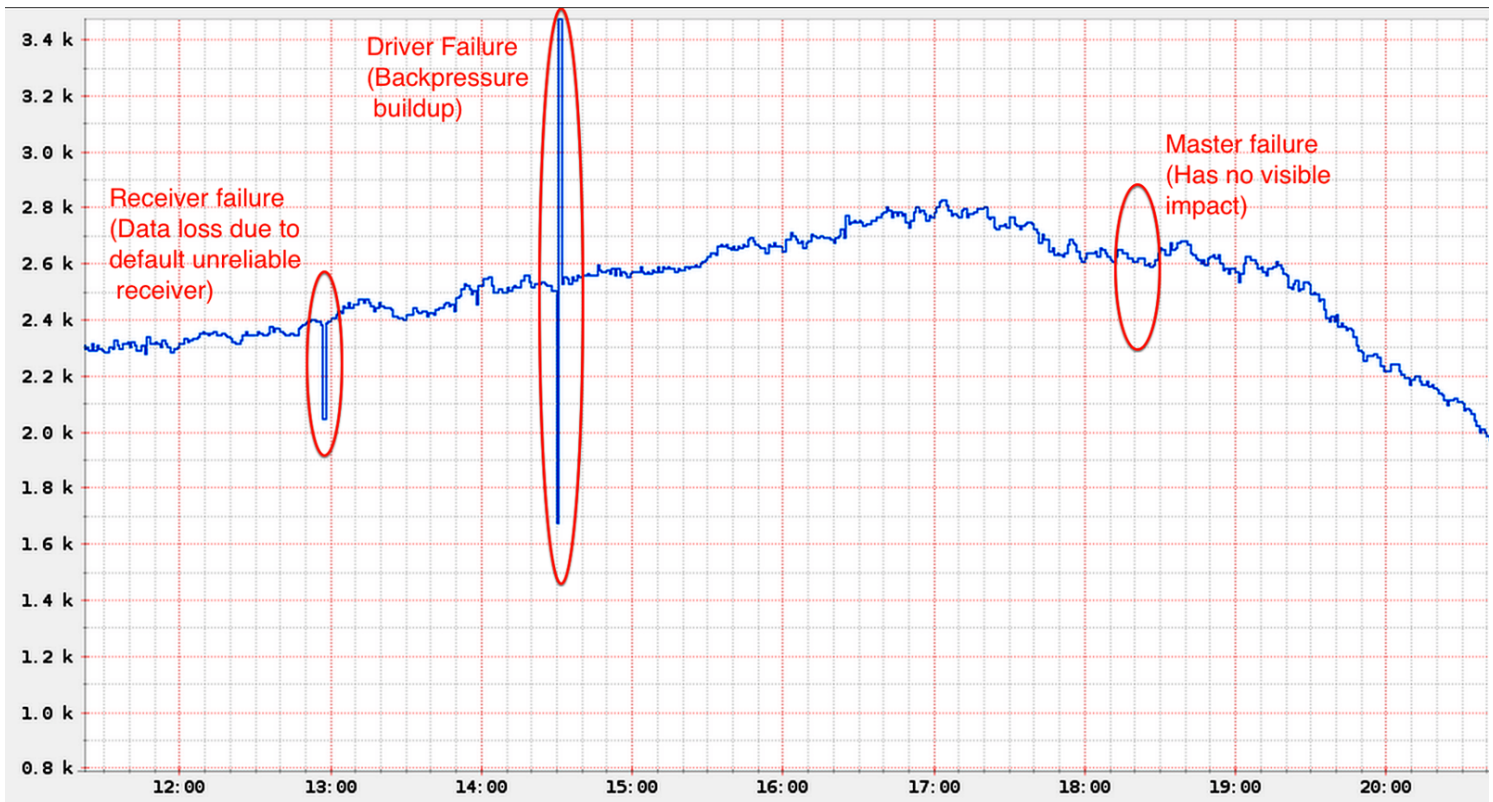


Resiliency: *Netflix tutorial*

Can Spark Streaming survive Chaos Monkey?

Bharat Venkat, Prasanna Padmanabhan, Antony Arokiasamy, Raju Uppalapati

techblog.netflix.com/2015/03/can-spark-streaming-survive-chaos-monkey.html



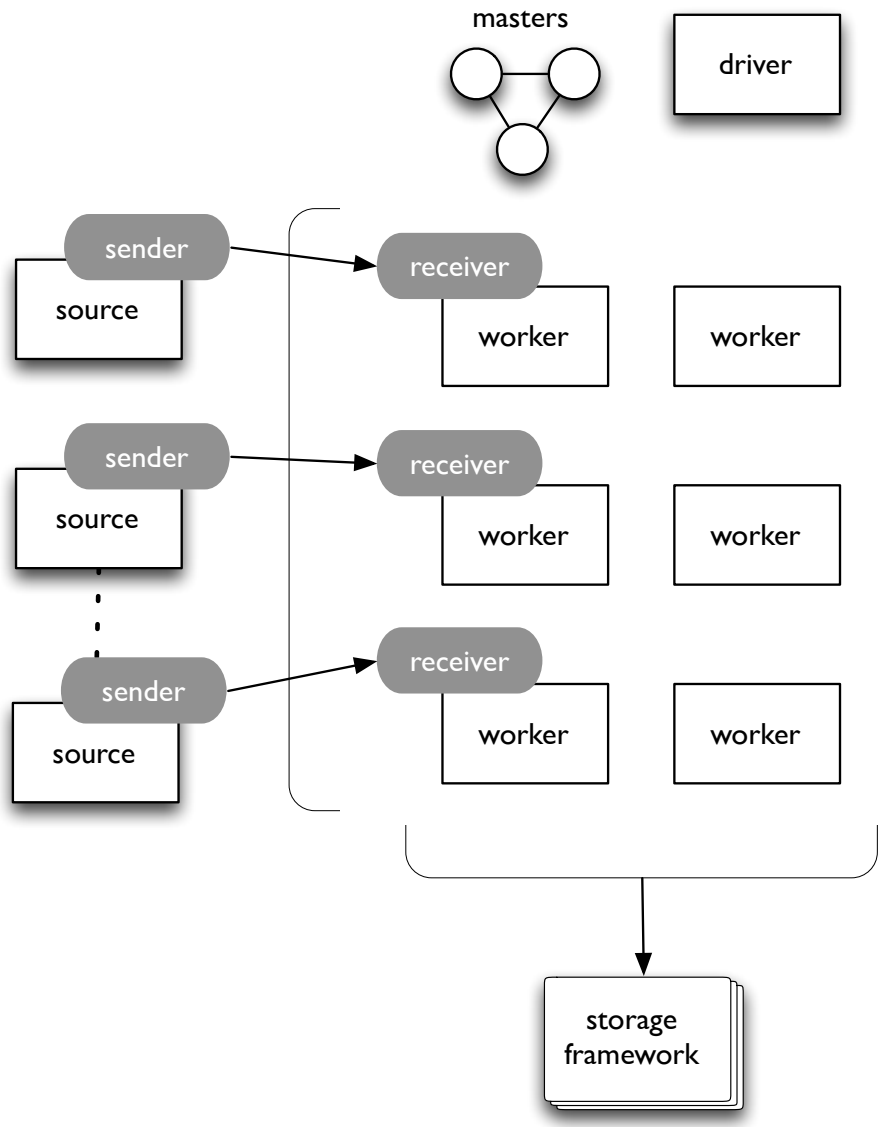
Resiliency: *other resources*

HA Spark Streaming defined:
youtu.be/jcJq3ZalXD8

excellent discussion of fault-tolerance (2012):
cs.duke.edu/~kmoses/cps516/dstream.html

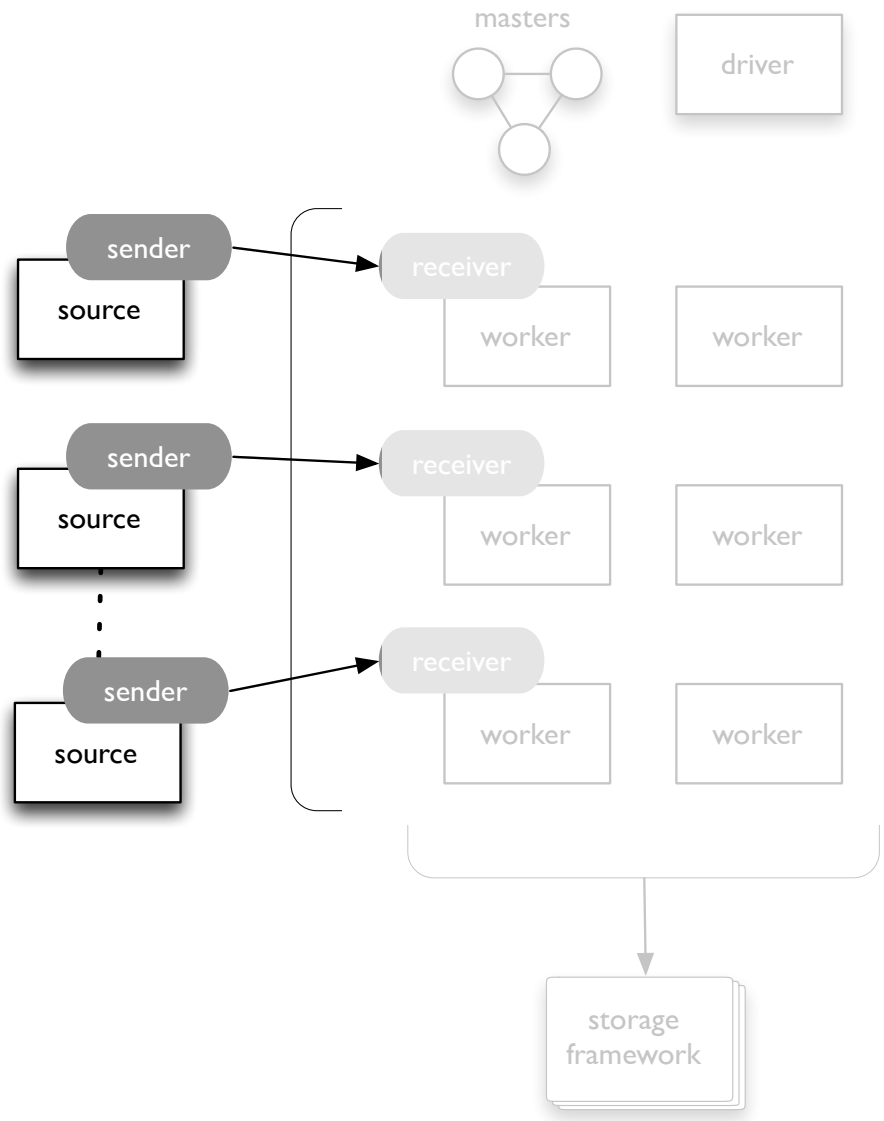
Improved Fault-tolerance and Zero Data Loss in Spark Streaming
Tathagata Das, 2015-01-15
databricks.com/blog/2015/01/15/improved-driver-fault-tolerance-and-zero-data-loss-in-spark-streaming.html

Resiliency: *illustrated*



(resiliency features)

Resiliency: *illustrated*



backpressure
(flow control is a hard problem)

reliable receiver

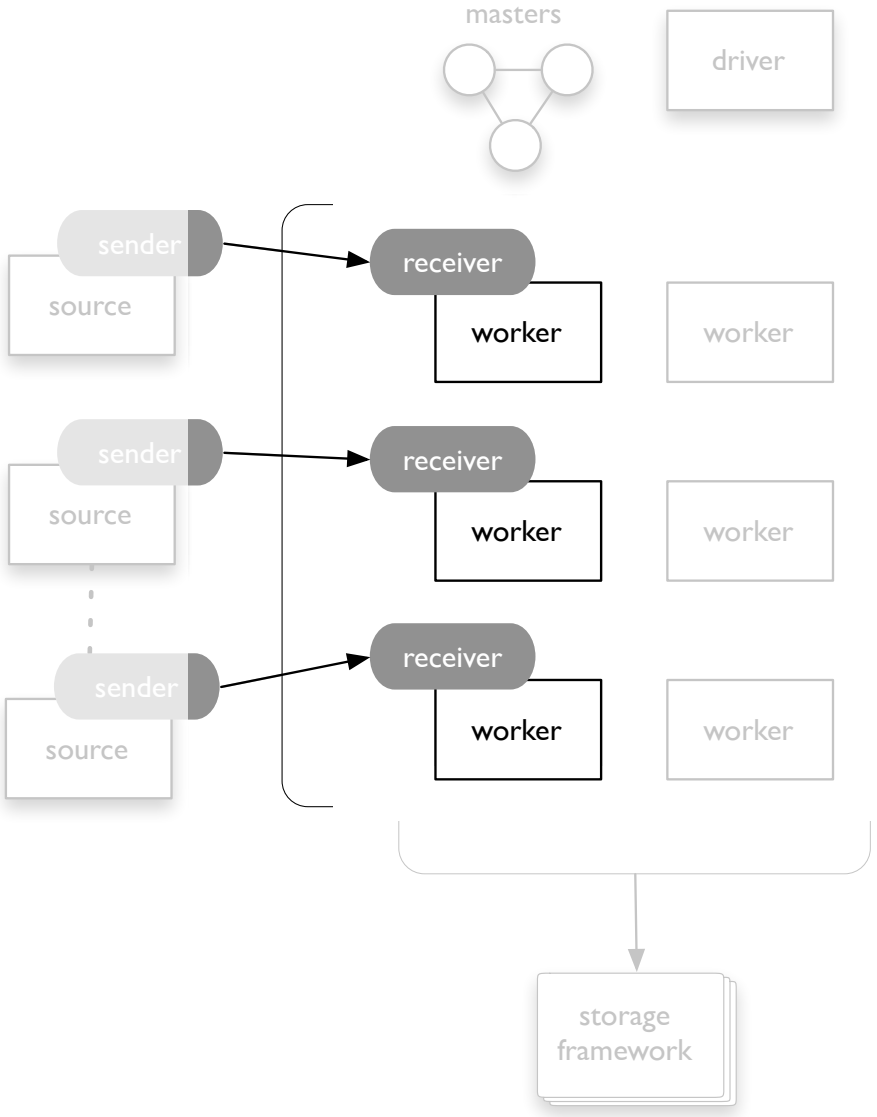
in-memory replication
write ahead log (data)

driver restart
checkpoint (metadata)

multiple masters

worker relaunch
executor relaunch

Resiliency: *illustrated*



backpressure
(flow control is a hard problem)

reliable receiver

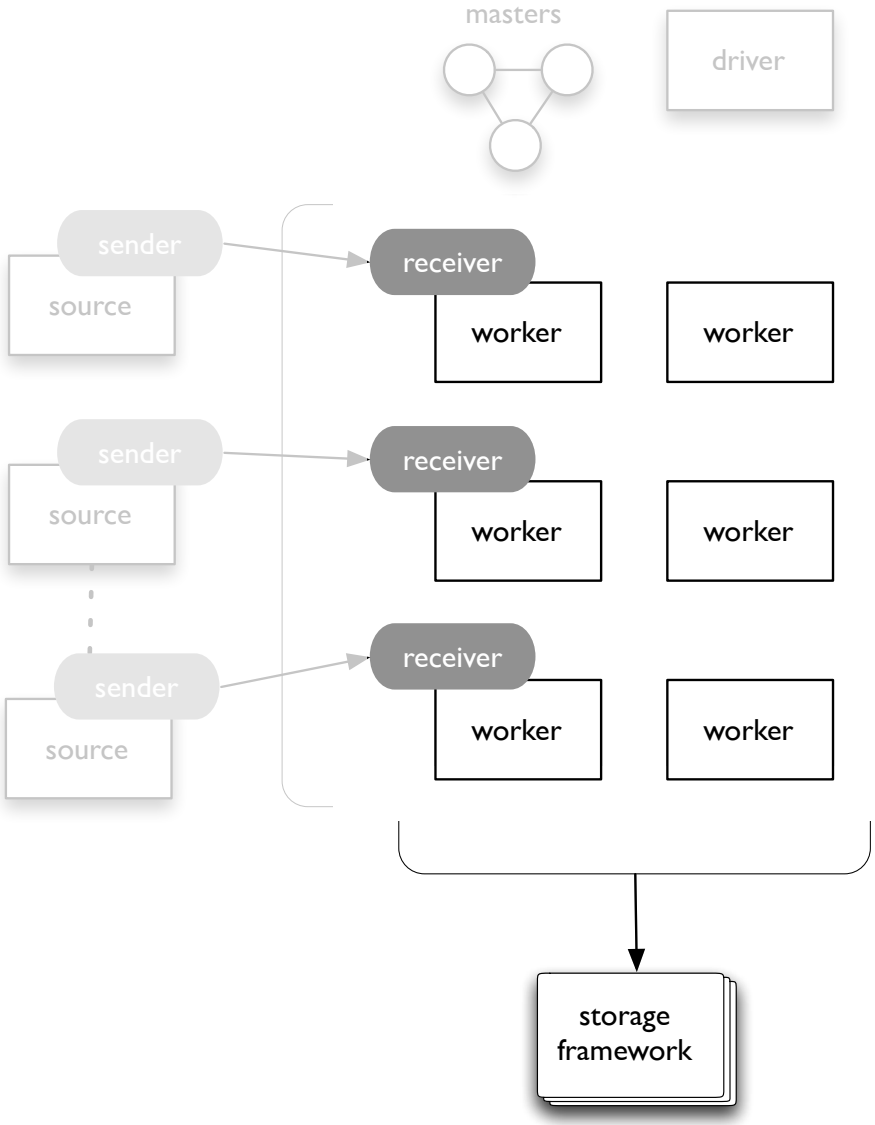
in-memory replication
write ahead log (data)

driver restart
checkpoint (metadata)

multiple masters

worker relaunch
executor relaunch

Resiliency: *illustrated*



backpressure
(flow control is a hard problem)

reliable receiver

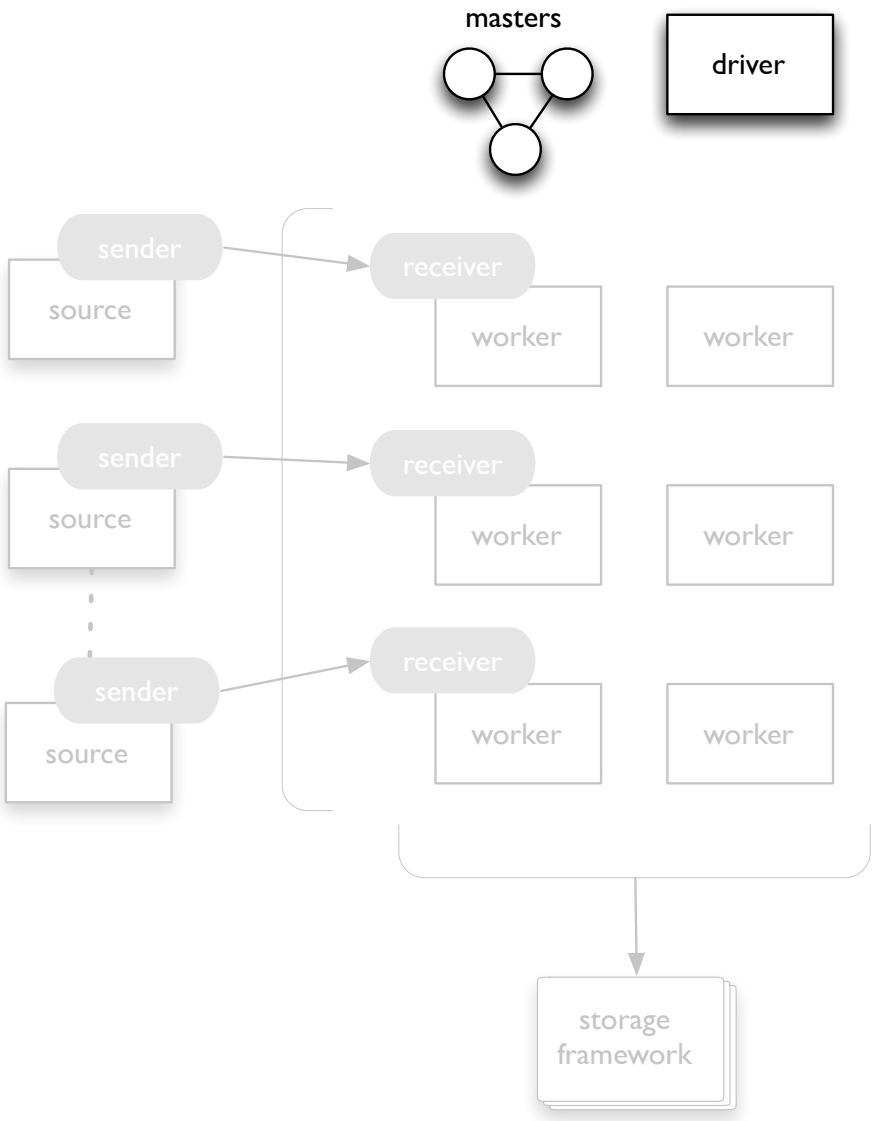
in-memory replication
write ahead log (data)

driver restart
checkpoint (metadata)

multiple masters

worker relaunch
executor relaunch

Resiliency: *illustrated*



backpressure
(flow control is a hard problem)

reliable receiver

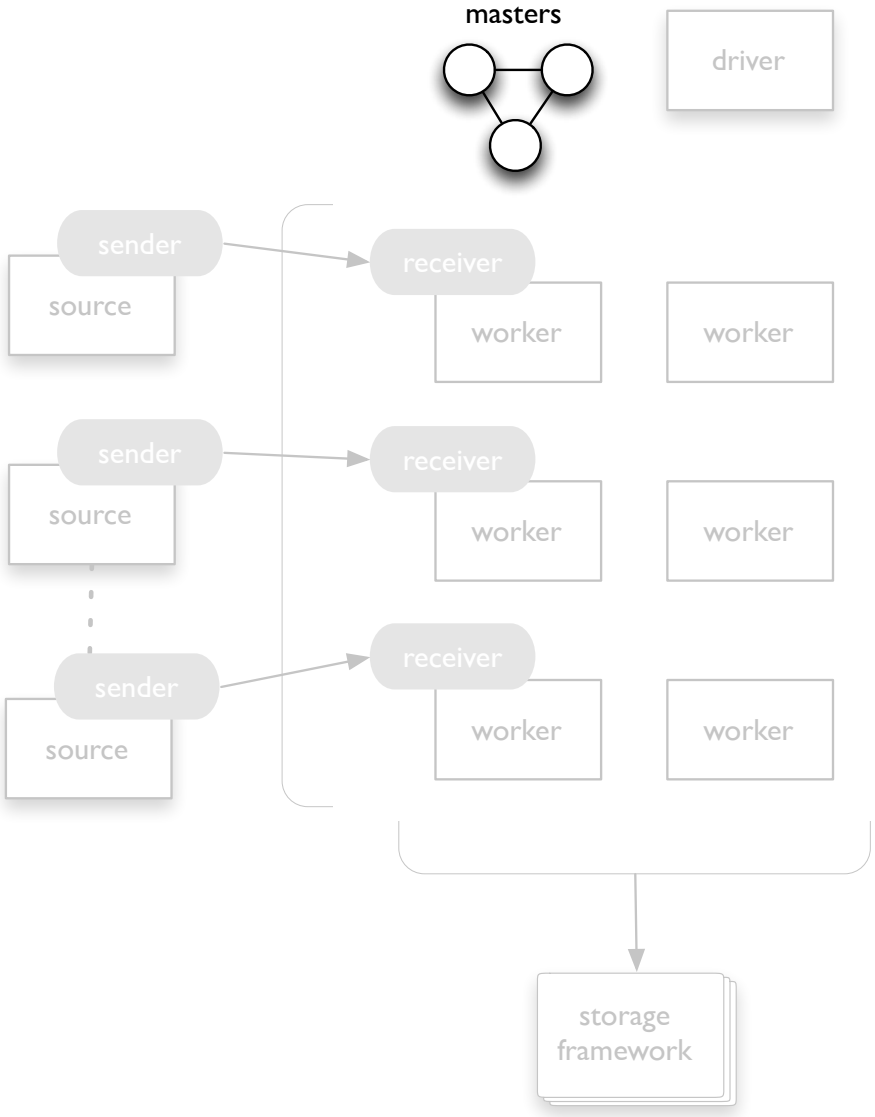
in-memory replication
write ahead log (data)

driver restart
checkpoint (metadata)

multiple masters

worker relaunch
executor relaunch

Resiliency: *illustrated*



backpressure
(flow control is a hard problem)

reliable receiver

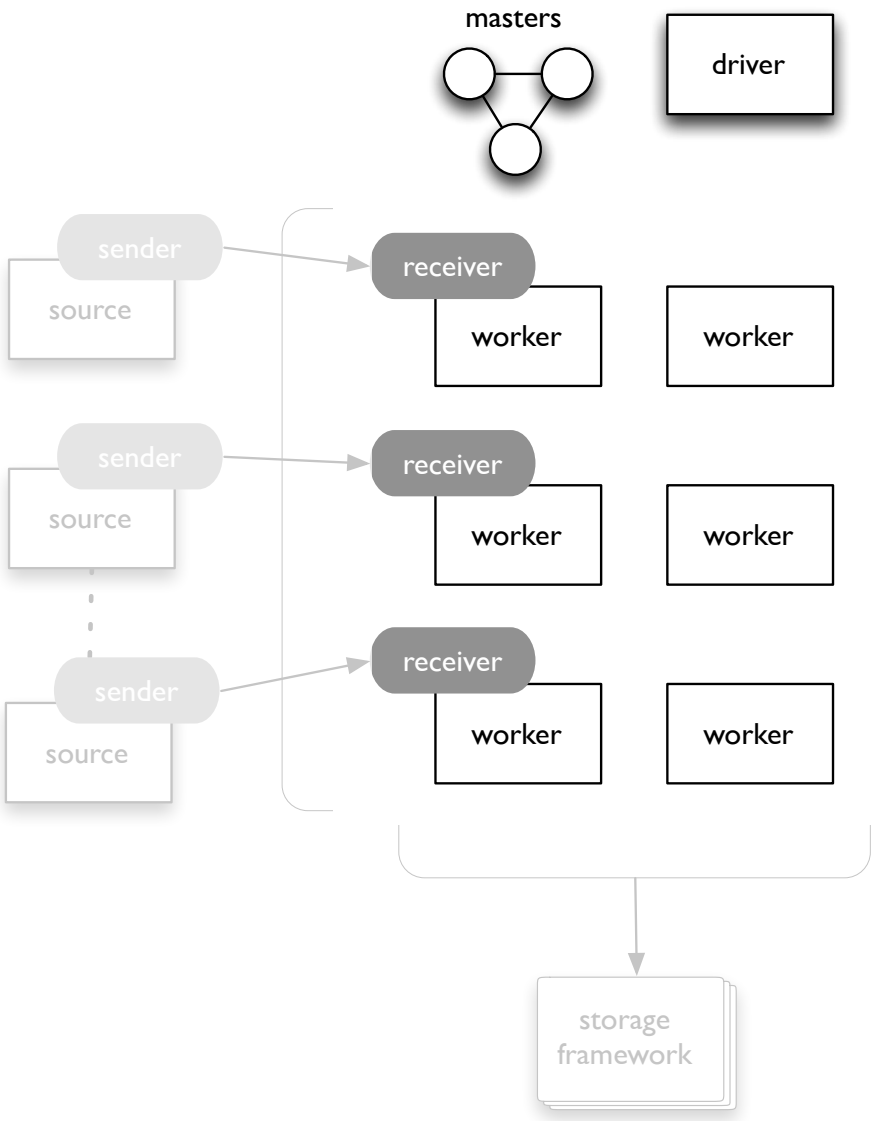
in-memory replication
write ahead log (data)

driver restart
checkpoint (metadata)

multiple masters

worker relaunch
executor relaunch

Resiliency: *illustrated*



backpressure
(flow control is a hard problem)

reliable receiver

in-memory replication
write ahead log (data)

driver restart
checkpoint (metadata)

multiple masters

worker relaunch
executor relaunch

Integrations: *architectural pattern deployed frequently in the field…*

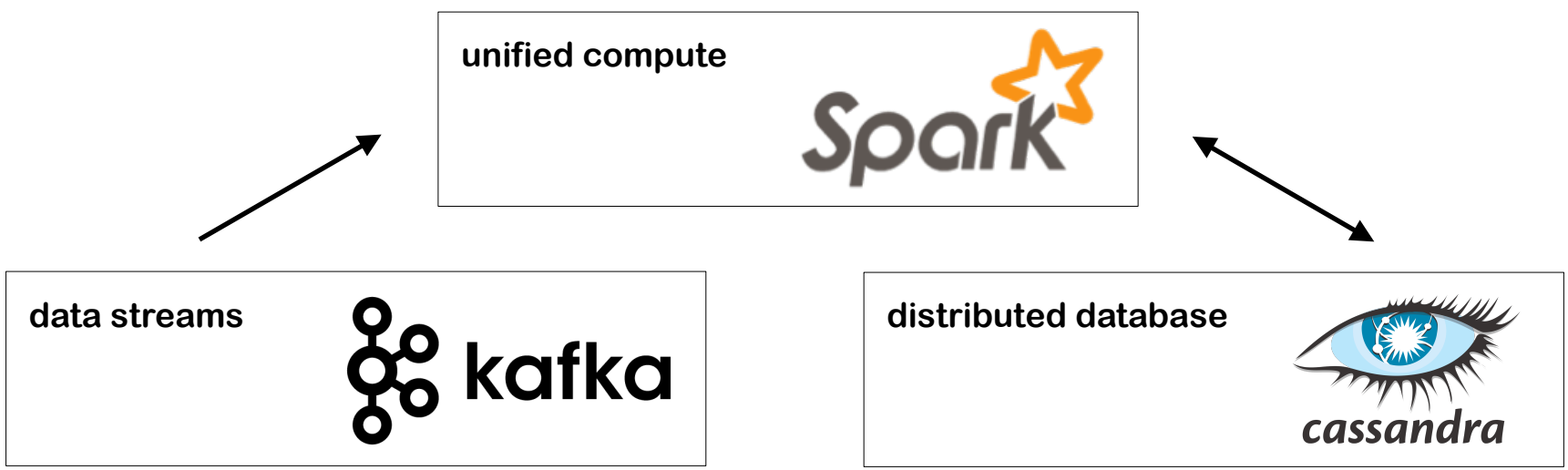
Kafka + Spark + Cassandra

datastax.com/documentation/datastax_enterprise/4.7/datastax_enterprise/spark/sparkIntro.html

http://helenaedelson.com/?p=991

github.com/datastax/spark-cassandra-connector

github.com/dibbhatt/kafka-spark-consumer



Integrations: *rich search, immediate insights*

Spark + ElasticSearch

databricks.com/blog/2014/06/27/application-spotlight-elasticsearch.html

elasticsearch.org/guide/en/elasticsearch/hadoop/current/spark.html

spark-summit.org/2014/talk/streamlining-search-indexing-using-elastic-search-and-spark

Because
Use Cases

Because Use Cases: *+80 known production use cases*

Because Use Cases: *Stratio*

Stratio Streaming: a new approach to Spark Streaming
David Morales, **Oscar Mendez**
2014-06-30
spark-summit.org/2014/talk/stratio-streaming-a-new-approach-to-spark-streaming

- Stratio Streaming is the union of a real-time messaging bus with a complex event processing engine using Spark Streaming
- allows the creation of streams and queries on the fly
- paired with Siddhi CEP engine and Apache Kafka
- added global features to the engine such as auditing and statistics
- use cases: large banks, retail, travel, etc.
- using Apache Mesos

Because Use Cases: *Pearson*

Pearson uses Spark Streaming for next generation adaptive learning platform
Dibyendu Bhattacharya
2014-12-08
databricks.com/blog/2014/12/08/pearson-uses-spark-streaming-for-next-generation-adaptive-learning-platform.html

- Kafka + Spark + Cassandra + Blur, on AWS on a YARN cluster
- single platform/common API was a key reason to replace Storm with Spark Streaming
- custom Kafka Consumer for Spark Streaming, using Low Level Kafka Consumer APIs
- handles: Kafka node failures, receiver failures, leader changes, committed offset in ZK, tunable data rate throughput

Because Use Cases: *Guavus*

Guavus Embeds Apache Spark
into its Operational Intelligence Platform
Deployed at the World's Largest Telcos
Eric Carr
2014-09-25
databricks.com/blog/2014/09/25/guavus-embeds-apache-spark-into-its-operational-intelligence-platform-deployed-at-the-worlds-largest-telcos.html

- 4 of 5 top mobile network operators, 3 of 5 top Internet backbone providers, 80% MSOs in NorAm
- analyzing 50% of US mobile data traffic, +2.5 PB/day
- latency is critical for resolving operational issues before they cascade: 2.5 MM transactions per second
- "analyze first" not "store first ask questions later"

Because Use Cases: *Sharethrough*

Spark Streaming for Realtime Auctions
Russell Cardullo
2014-06-30
slideshare.net/RussellCardullo/russell-cardullo-spark-summit-2014-36491156

- the profile of a 24 x 7 streaming app is different than an hourly batch job…
- data sources from RabbitMQ, Kinesis
- ingest ~0.5 TB daily, mainly click stream and application logs, 5 sec micro-batch
- feedback based on click stream events into auction system for model correction
- monoids… using **Algebird**
- using Apache Mesos on AWS

Because Use Cases: *Freeman Lab, Janelia*

hhmi | janelia Research Campus

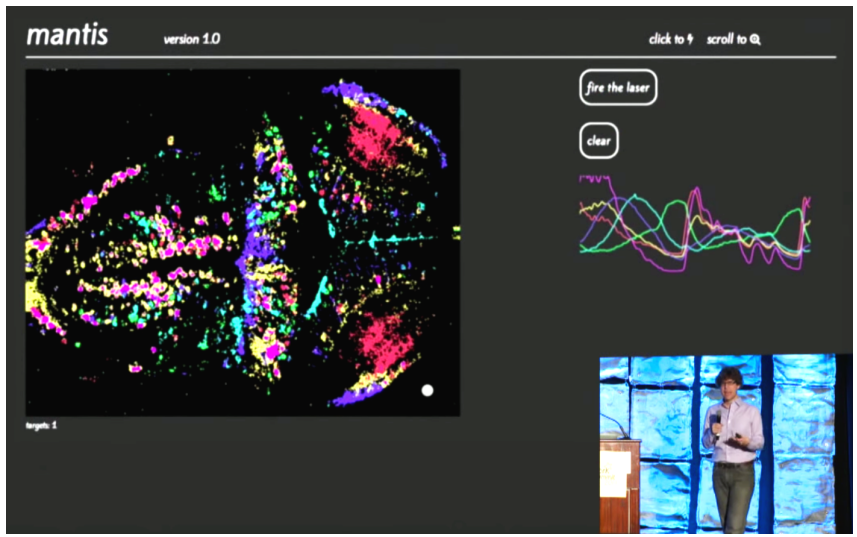
Analytics + Visualization for Neuroscience: Spark, Thunder, Lightning

Jeremy Freeman

2015-01-29

youtu.be/cBQm4LhHn9g?t=28m55s

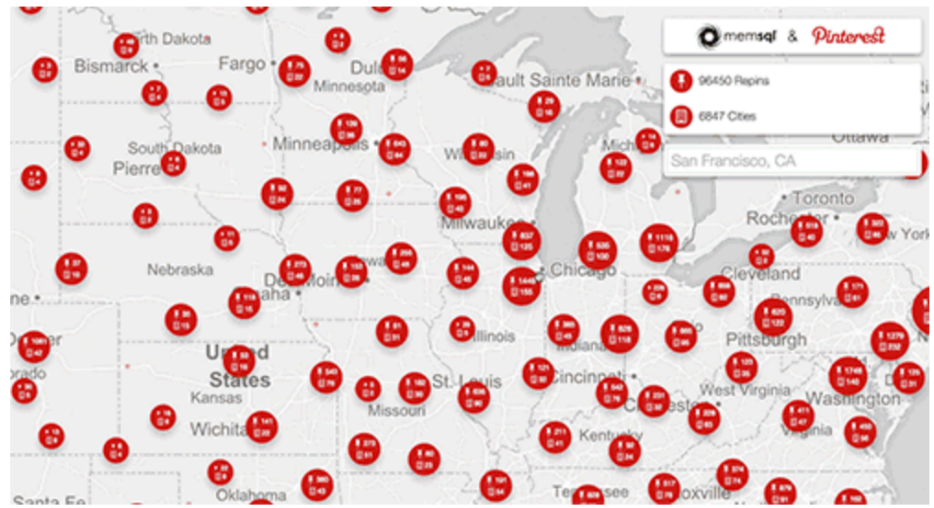
- genomics research – zebrafish neuroscience studies
- real-time ML for laser control
- 2 TB/hour per fish
- 80 HPC nodes



Because Use Cases: *Pinterest*

Real-time analytics at Pinterest
Krishna Gade
2015-02-18
engineering.pinterest.com/post/111380432054/real-time-analytics-at-pinterest

- higher performance event logging
- reliable log transport and storage
- faster query execution on real-time data
- integrated with MemSQL



Because Use Cases: *Ooyala*

Productionizing a 24/7 Spark Streaming service on YARN

Issac Buenrostro, Arup Malakar

2014-06-30

spark-summit.org/2014/talk/productionizing-a-247-spark-streaming-service-on-yarn

- state-of-the-art ingestion pipeline, processing over two billion video events a day
- how do you ensure 24/7 availability and fault tolerance?
- what are the best practices for Spark Streaming and its integration with Kafka and YARN?
- how do you monitor and instrument the various stages of the pipeline?

A Big Picture

A Big Picture...

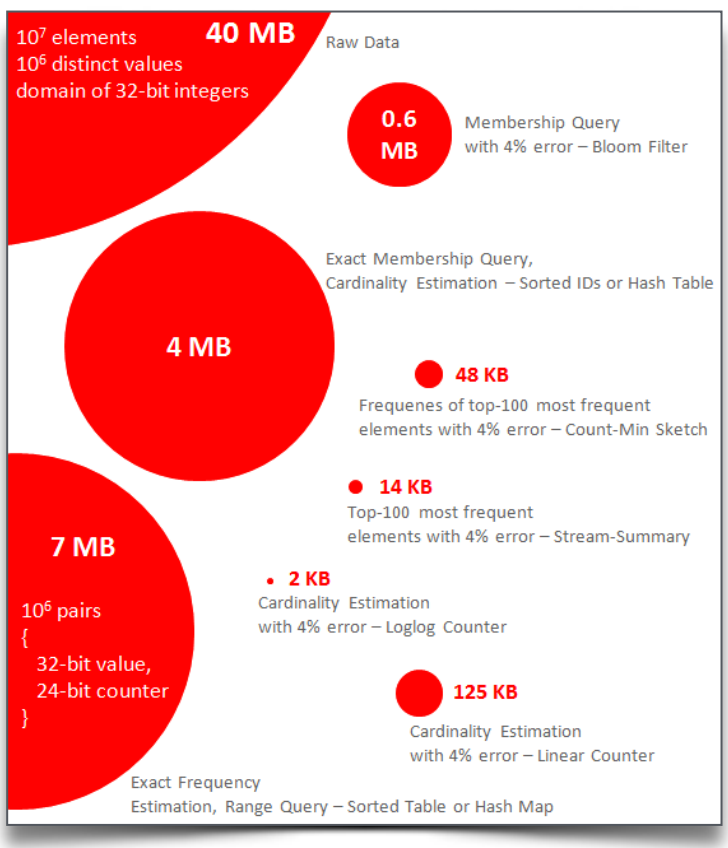
19-20c. statistics emphasized *defensibility* in lieu of *predictability*, based on analytic variance and goodness-of-fit tests

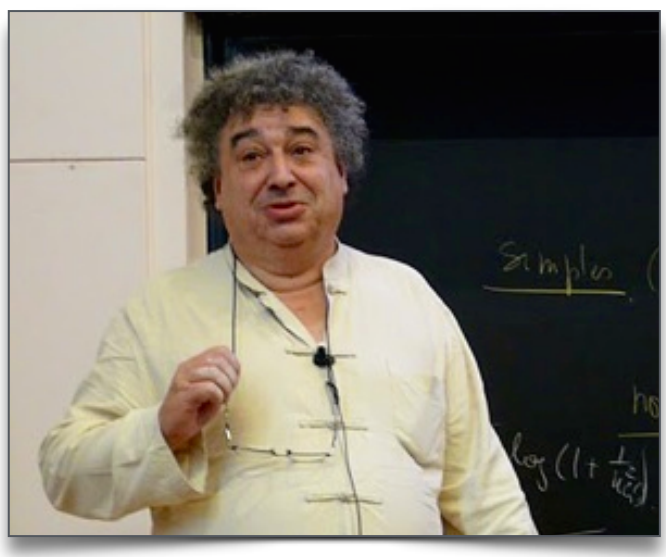
That approach inherently led toward a manner of computational thinking based on **batch windows**

They missed a subtle point…

A Big Picture... The view in the lens has changed

21c. shift towards modeling based on probabilistic approximations: trade bounded errors for greatly reduced resource costs

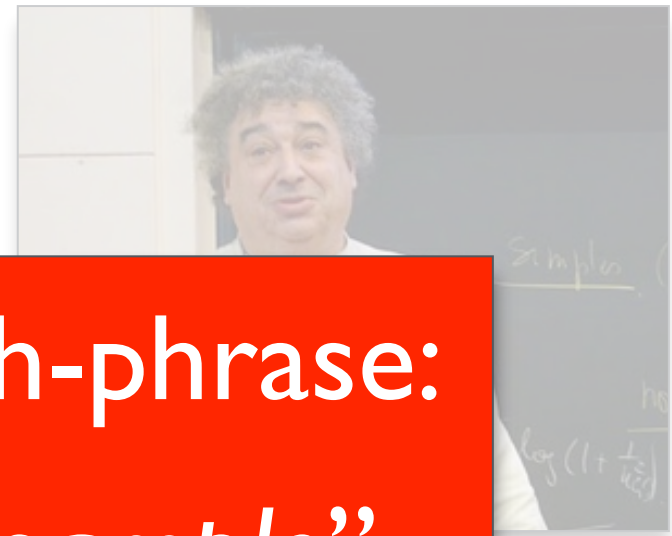




highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/

A Big Picture... The view in the lens has changed

21c. shift towards modeling based on probabil approximations: trade bounded errors for greatly reduced resource costs



Twitter catch-phrase:

"Hash, don't sample"

highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/

Probabilistic Data Structures:

a fascinating and relatively new area, pioneered by relatively few people – e.g., **Philippe Flajolet**

provides *approximation*, with error bounds – in general uses significantly less resources (RAM, CPU, etc.)

many algorithms can be constructed from combinations of read and write *monoids*

aggregate different ranges by composing hashes, instead of repeating full-queries

Probabilistic Data Structures: *Some Examples*

algorithm	*use case*	*example*
Count-Min Sketch	frequency summaries	**code**
HyperLogLog	set cardinality	**code**
Bloom Filter	set membership	
MinHash	set similarity	
DSQ	streaming quantiles	
SkipList	ordered sequence search	

Probabilistic Data Structures: *Some Examples*

algorithm	*use case*	*example*
Count-Min Sketch	frequency summaries	**code**
HyperLogLog	set cardinality	**code**
Bloom Filter		
MinHash		
DSQ	streaming quantiles	
SkipList	ordered sequence search	

suggestion: consider these as quintessential collections data types at scale

Probabilistic Data Structures: *Performance Bottlenecks*

Add ALL the Things:
Abstract Algebra Meets Analytics

infoq.com/presentations/abstract-algebra-analytics

Avi Bryant, Strange Loop (2013)

Avi Bryant
@avibryant

- *grouping doesn't matter (associativity)*
- *ordering doesn't matter (commutativity)*
- *zeros get ignored*

In other words, while partitioning data at scale is quite difficult, you can let the math allow your code to be flexible at scale

Probabilistic Data Structures: *Industry Drivers*

- sketch algorithms: trade bounded errors for orders of magnitude less required resources, e.g., fit more complex apps in memory
- multicore + large memory spaces (off heap) are increasing the resources per node in a cluster
- containers allow for finer-grain allocation of cluster resources and multi-tenancy
- monoids, etc.: guarantees of associativity within the code allow for more effective distributed computing, e.g., partial aggregates
- less resources must be spent sorting/windowing data prior to working with a data set
- real-time apps, which don't have the luxury of anticipating data partitions, can respond quickly

Probabilistic Data Structures: *Recommended Reading*

Probabilistic Data Structures for Web Analytics and Data Mining
Ilya Katsov (2012-05-01)

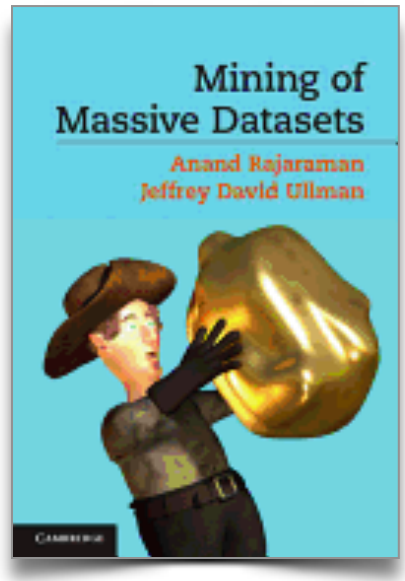
A collection of links for streaming algorithms and data structures
Debasish Ghosh

Aggregate Knowledge blog (now Neustar)
Timon Karnezos, **Matt Curcio**, et al.

Probabilistic Data Structures and Breaking Down Big Sequence Data
C. Titus Brown, O'Reilly (2010-11-10)

Algebird
Avi Bryant, **Oscar Boykin**, et al. Twitter (2012)

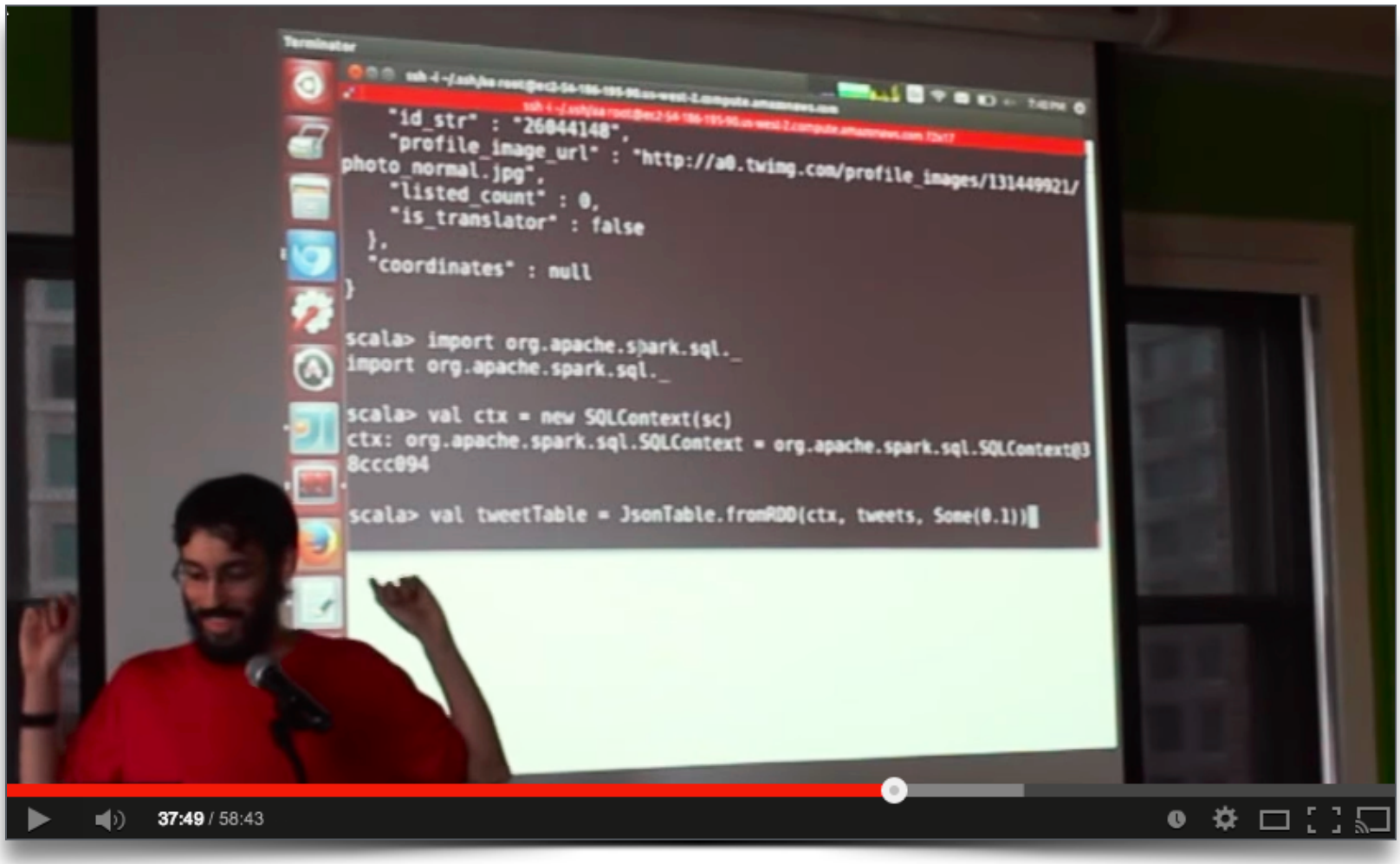
Mining of Massive Datasets
Jure Leskovec, **Anand Rajaraman**, **Jeff Ullman**, Cambridge (2011)



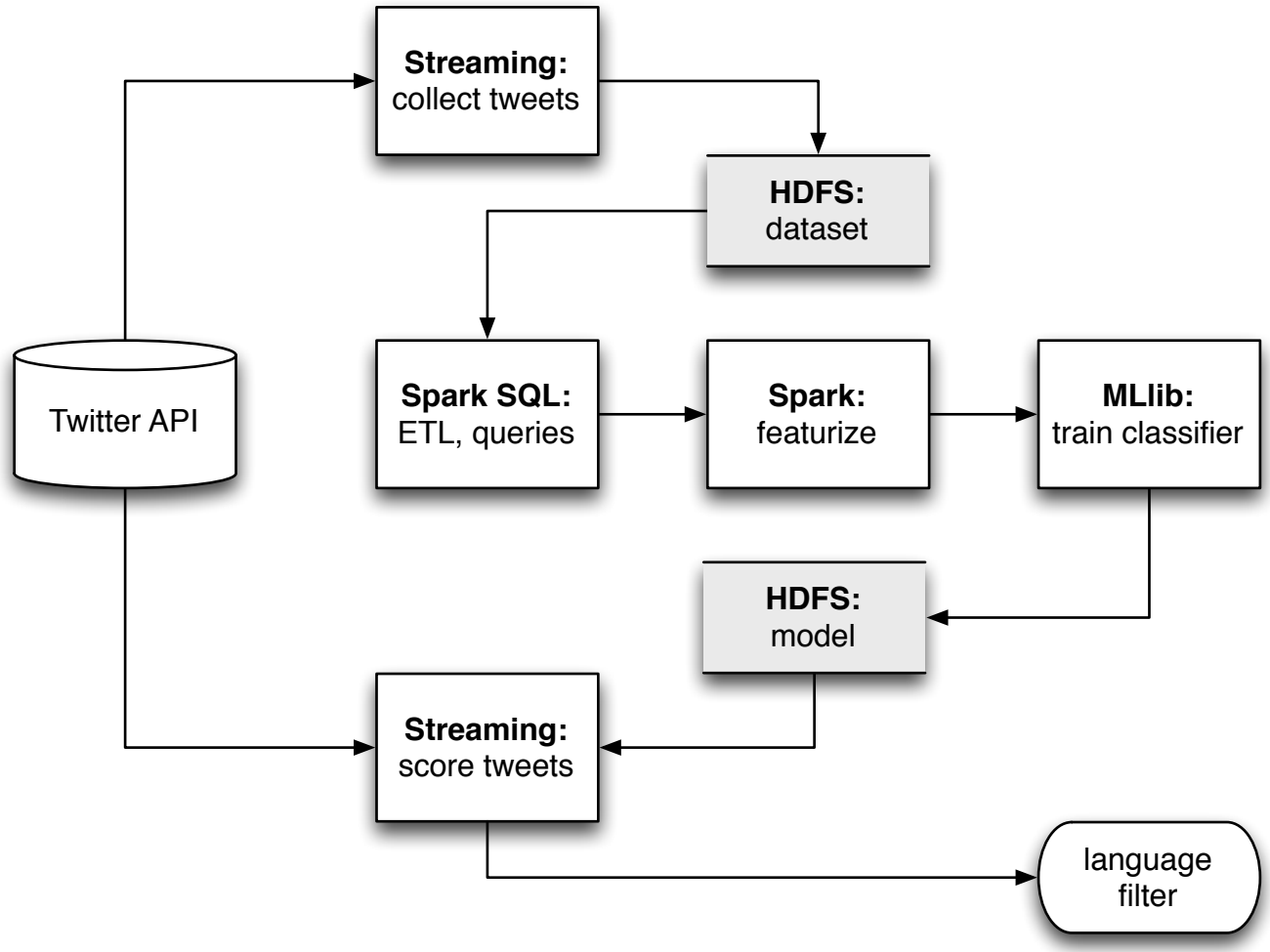
Demo

Demo: *Twitter Streaming Language Classifier*

databricks.gitbooks.io/databricks-spark-reference-applications/content/twitter_classifier/README.html



Demo: *Twitter Streaming Language Classifier*



Demo: *Twitter Streaming Language Classifier*

From tweets to ML features, approximated as sparse vectors:

1. extract text from the tweet	`https://twitter.com/andy_bf/status/16222269370011648`	`"Ceci n'est pas un tweet"`
2. sequence text as bigrams	`tweet.sliding(2).toSeq`	`("Ce", "ec", "ci", …, )`
3. convert bigrams into numbers	`seq.map(_.hashCode())`	`(2178, 3230, 3174, …, )`
4. index into sparse tf vector	`seq.map(_.hashCode() % 1000)`	`(178, 230, 174, …, )`
5. increment feature count	`Vector.sparse(1000, …)`	`(1000, [102, 104, …], [0.0455, 0.0455, …])`

Demo: *Twitter Streaming Language Classifier*

Sample Code + Output:

gist.github.com/ceteri/835565935da932cb59a2

```
val sc = new SparkContext(new SparkConf())
val ssc = new StreamingContext(conf, Seconds(5))

val tweets = TwitterUtils.createStream(ssc, Utils.getAuth)
val statuses = tweets.map(_.getText)

val model = new KMeansModel(ssc.sparkContext.objectFile[Vector]
(modelFile.toString).collect())

val filteredTweets = statuses
  .filter(t =>
    model.predict(Utils.featurize(t)) == clust)
filteredTweets.print()

ssc.start()
ssc.awaitTermination()
```

```
CLUSTER 1:
TLあんまり見ないけど
@くれたっら
いつでもくっるよ٩(δωδ)۶
そういえばディスガイアも今日か

CLUSTER 4:
قالوا العروبه روحت بعد صدام
واقول مع سلمان تحيى العروبه
RT @vip588: √ فولو مي \ زيادة متابعين \ للمتواجدين الان
vip588 √ \ فولو للي عمل رتويت \ رتويت للتغريدة \ فولو باك
اللي ما يلتزم ما بيستفيد ...
ن سورة
```

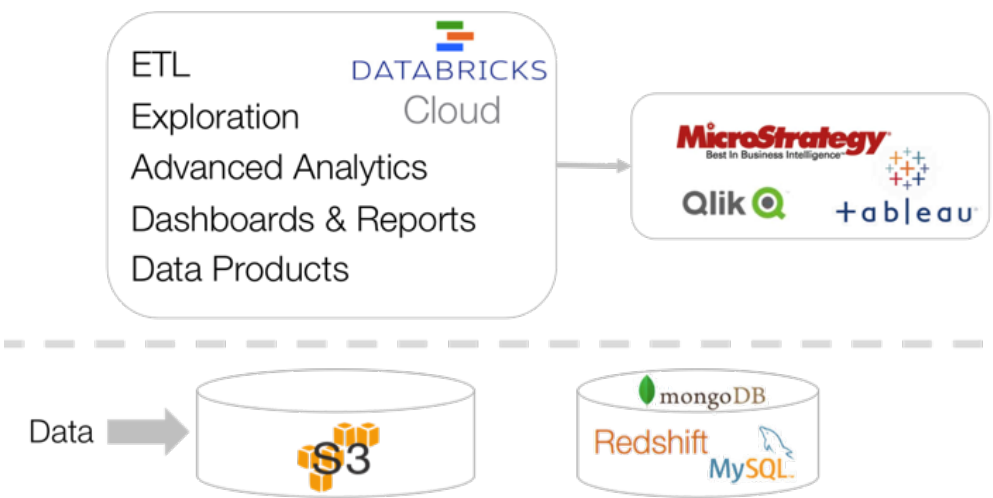
Further Resources + Q&A

databricks

cloud-based notebooks:

databricks.com/blog/2014/07/14/databricks-cloud-making-big-data-easy.html

our monthly newsletter:
go.databricks.com/newsletter-sign-up



Spark Developer Certification

- go.databricks.com/spark-certified-developer
- defined by Spark experts @Databricks
- assessed by O'Reilly Media
- establishes the bar for Spark expertise

community:

spark.apache.org/community.html

events worldwide: goo.gl/2YqJZK

YouTube channel: goo.gl/N5Hx3h

video+preso archives: spark-summit.org

resources: databricks.com/spark-training-resources

workshops: databricks.com/spark-training

MOOCs:

Anthony Joseph
UC Berkeley
begins Apr 2015
edx.org/course/uc-berkeleyx/uc-berkeleyx-cs100-1x-introduction-big-6181

Introduction to Big Data with Apache Spark

Learn how to apply data science techniques using parallel programming in Apache Spark to explore big (and small) data.

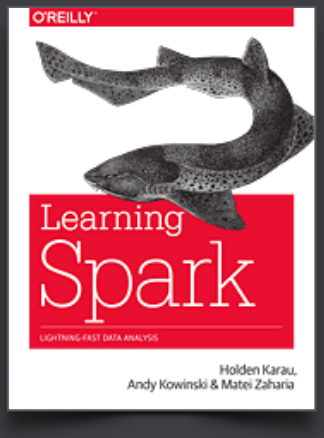
Scalable Machine Learning

Learn the underlying principles required to develop scalable machine learning pipelines and gain hands-on experience using Apache Spark.

Ameet Talwalkar
UCLA
begins Q2 2015
edx.org/course/uc-berkeleyx/uc-berkeleyx-cs190-1x-scalable-machine-6066

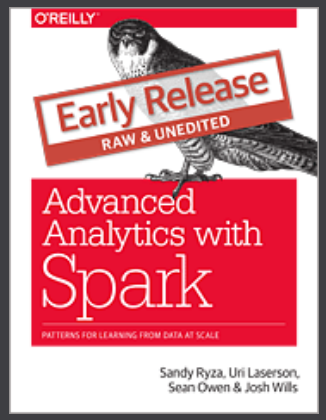
books+videos:

Learning Spark
Holden Karau, Andy Konwinski, Parick Wendell, Matei Zaharia
O'Reilly (2015)
shop.oreilly.com/product/06369200285 12.do



Intro to Apache Spark
Paco Nathan
O'Reilly (2015)
shop.oreilly.com/product/0636920036807.do

Advanced Analytics with Spark
Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills
O'Reilly (2014)
shop.oreilly.com/product/0636920035091.do



Spark in Action
Chris Fregly
Manning (2015)
sparkinaction.com/

Fast Data Processing with Spark
Holden Karau
Packt (2013)
shop.oreilly.com/product/9781782167068.do

databricks

confs:

CodeNeuro
NYC, Apr 10-11
codeneuro.org/2015/NYC/

Big Data Tech Con
Boston, Apr 26-28
bigdatatechcon.com

Next.ML
Boston, Apr 27
www.next.ml/

Strata EU
London, May 5-7
strataconf.com/big-data-conference-uk-2015

GOTO Chicago
Chicago, May 11-14
gotocon.com/chicago-2015

Scala Days
Amsterdam, Jun 8-10
event.scaladays.org/scaladays-amsterdam-2015

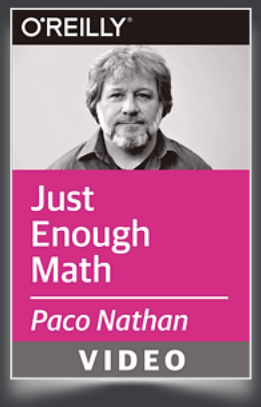
Spark Summit 2015
SF, Jun 15-17
spark-summit.org


databricks

presenter:

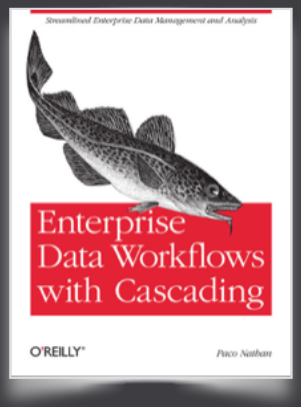
monthly newsletter for updates, events, conf summaries, etc.:

liber118.com/pxn/



Just Enough Math
O'Reilly, 2014

justenoughmath.com
preview: youtu.be/TQ58cWgdCpA



Enterprise Data Workflows with Cascading
O'Reilly, 2013

shop.oreilly.com/product/0636920028536.do

Obrigado!
Perguntas?

Tweet with #QCONBIGDATA to ask questions for the Big Data panel:

qconsp.com/presentation/Painel-Big-Data-e-Data-Science-tudo-que-voce-sempre-quis-saber
2015-03-26 18:00