

Spark Streaming Case Studies

Strata EU, Barcelona

2014-11-20

strataconf.com/strataeu2014/public/schedule/detail/37493

Paco Nathan
@pacoid



Spark, the elevator pitch

Spark, the elevator pitch

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

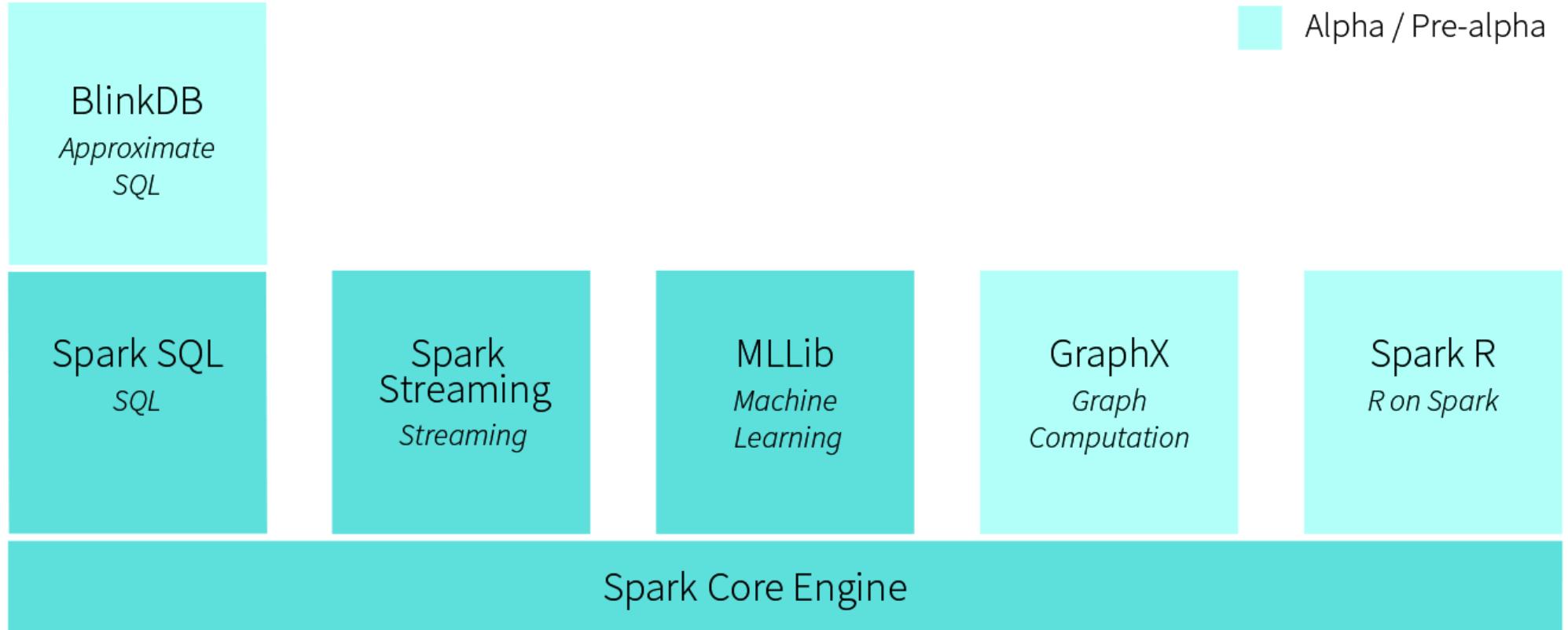
“Organizations that are looking at big data challenges – including collection, ETL, storage, exploration and analytics – should consider Spark for its in-memory performance and the breadth of its model.

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, the elevator pitch



Spark, the elevator pitch

circa 2010:
a unified engine for enterprise data workflows,
based on commodity hardware a decade later...



Spark: Cluster Computing with Working Sets

Matei Zaharia, Mosharaf Chowdhury,
Michael Franklin, Scott Shenker, Ion Stoica

people.csail.mit.edu/matei/papers/2010/hotcloud_spark.pdf

*Resilient Distributed Datasets: A Fault-Tolerant Abstraction for
In-Memory Cluster Computing*

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave,
Justin Ma, Murphy McCauley, Michael Franklin, Scott Shenker, Ion Stoica
usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf

Spark, the elevator pitch

Spark Core is the general execution engine for the Spark platform that other functionality is built atop:

- *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)



Spark, the elevator pitch

```
1 public class WordCount {
2     public static class TokenizerMapper
3         extends Mapper<Object, Text, Text, IntWritable> {
4
5     private final static IntWritable one = new IntWritable(1);
6     private Text word = new Text();
7
8     public void map(Object key, Text value, Context context
9                     ) throws IOException, InterruptedException {
10        StringTokenizer itr = new StringTokenizer(value.toString());
11        while (itr.hasMoreTokens()) {
12            word.set(itr.nextToken());
13            context.write(word, one);
14        }
15    }
16
17
18    public static class IntSumReducer
19        extends Reducer<Text,IntWritable,Text,IntWritable> {
20        private IntWritable result = new IntWritable();
21
22        public void reduce(Text key, Iterable<IntWritable> values,
23                           Context context
24                           ) throws IOException, InterruptedException {
25            int sum = 0;
26            for (IntWritable val : values) {
27                sum += val.get();
28            }
29            result.set(sum);
30            context.write(key, result);
31        }
32    }
33
34    public static void main(String[] args) throws Exception {
35        Configuration conf = new Configuration();
36        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
37        if (otherArgs.length < 2) {
38            System.err.println("Usage: wordcount <in> [<in>...] <out>");
39            System.exit(2);
40        }
41        Job job = new Job(conf, "word count");
42        job.setJarByClass(WordCount.class);
43        job.setMapperClass(TokenizerMapper.class);
44        job.setCombinerClass(IntSumReducer.class);
45        job.setReducerClass(IntSumReducer.class);
46        job.setOutputKeyClass(Text.class);
47        job.setOutputValueClass(IntWritable.class);
48        for (int i = 0; i < otherArgs.length - 1; ++i) {
49            FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
50        }
51        FileOutputFormat.setOutputPath(job,
52            new Path(otherArgs[otherArgs.length - 1]));
53        System.exit(job.waitForCompletion(true) ? 0 : 1);
54    }
55 }
```

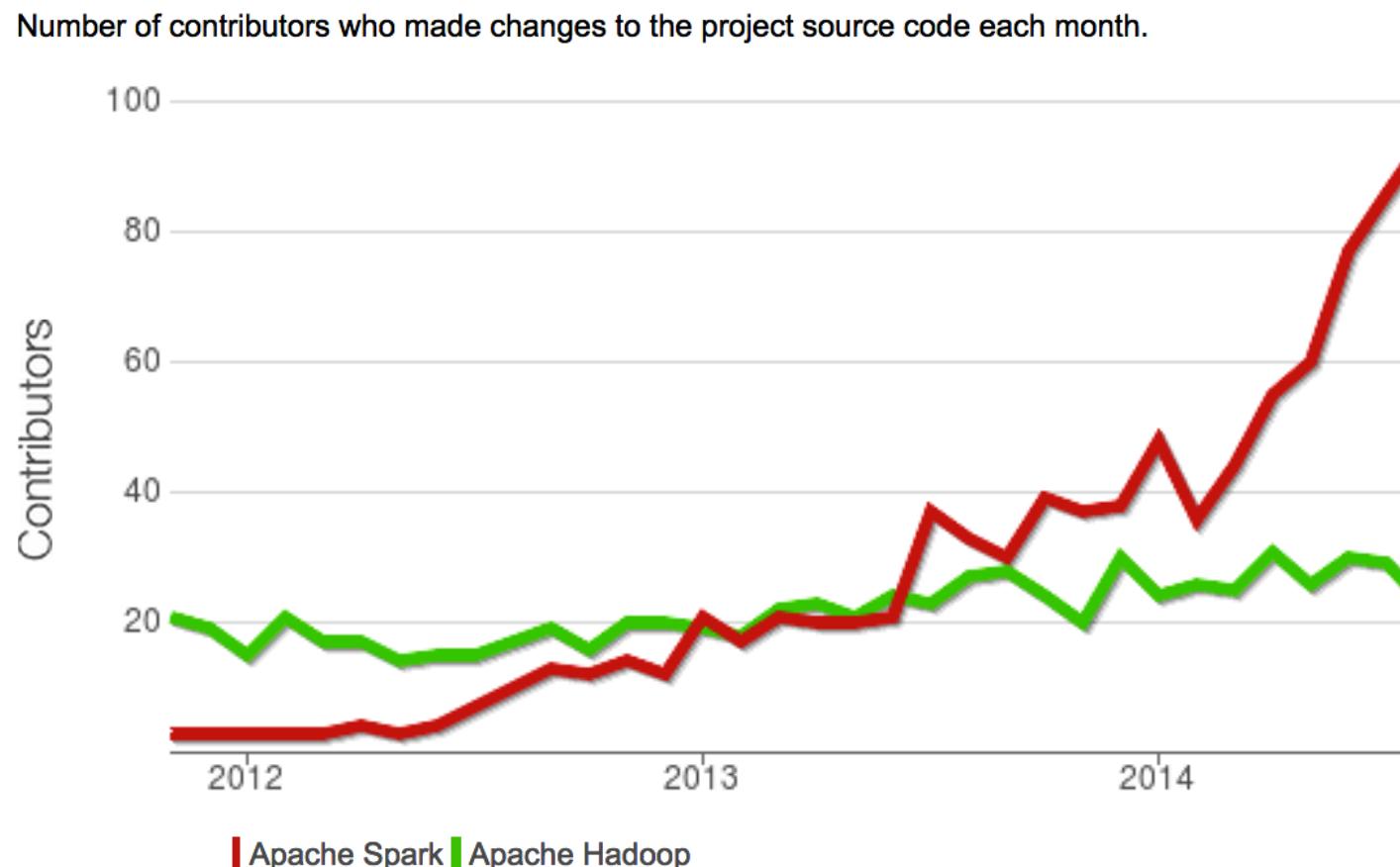
```
1 val f = sc.textFile(inputPath)
2 val w = f.flatMap(l => l.split(" ")).map(word => (word, 1)).cache()
3 w.reduceByKey(_ + _).saveAsText(outputPath)
```

WordCount in 3 lines of Spark

WordCount in 50+ lines of Java MR

Spark, the elevator pitch

Sustained exponential growth, as one of the most active Apache projects ohloh.net/orgs/apache



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



Why Streaming?

Why Streaming?

Because Machine Data!



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

Why Streaming?

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](http://research.google.com/pubs/pub41378.html)

Why Streaming?

Because IoT!

B4RM4N - be a cocktail hero

by Magnified Self

Home Updates 1 Backers 366 Comments 37 San Francisco, CA Product Design

366
Backers

\$38,055
pledged of \$100,000 goal

28
days to go

Back This Project
\$1 minimum pledge

This project will only be funded if at least \$100,000 is pledged by Thu, Dec 4 2014 10:59 PM PST.

kickstarter.com/projects/1614456084/b4rm4n-be-a-cocktail-hero

Why Streaming?

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/

Spark Streaming

Spark Streaming: Requirements

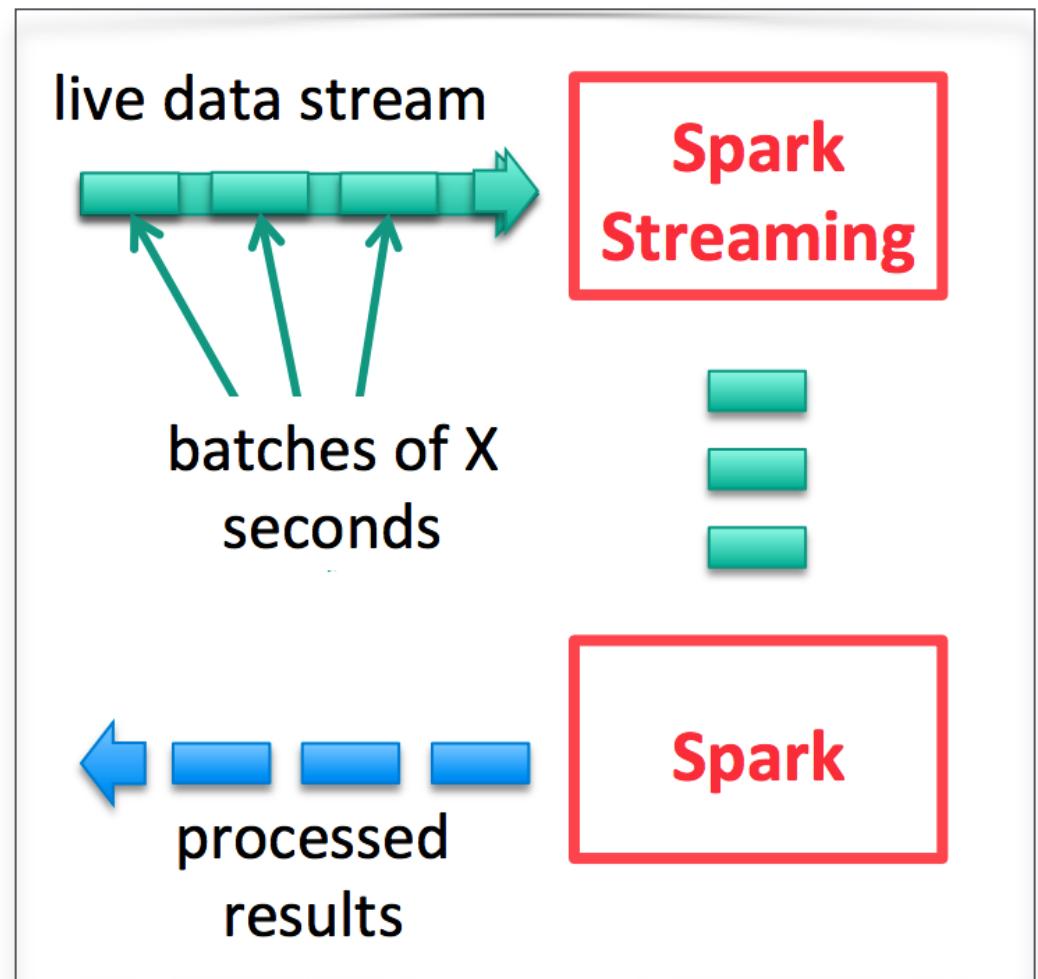
Let's consider the top-level requirements for a streaming framework:

- 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

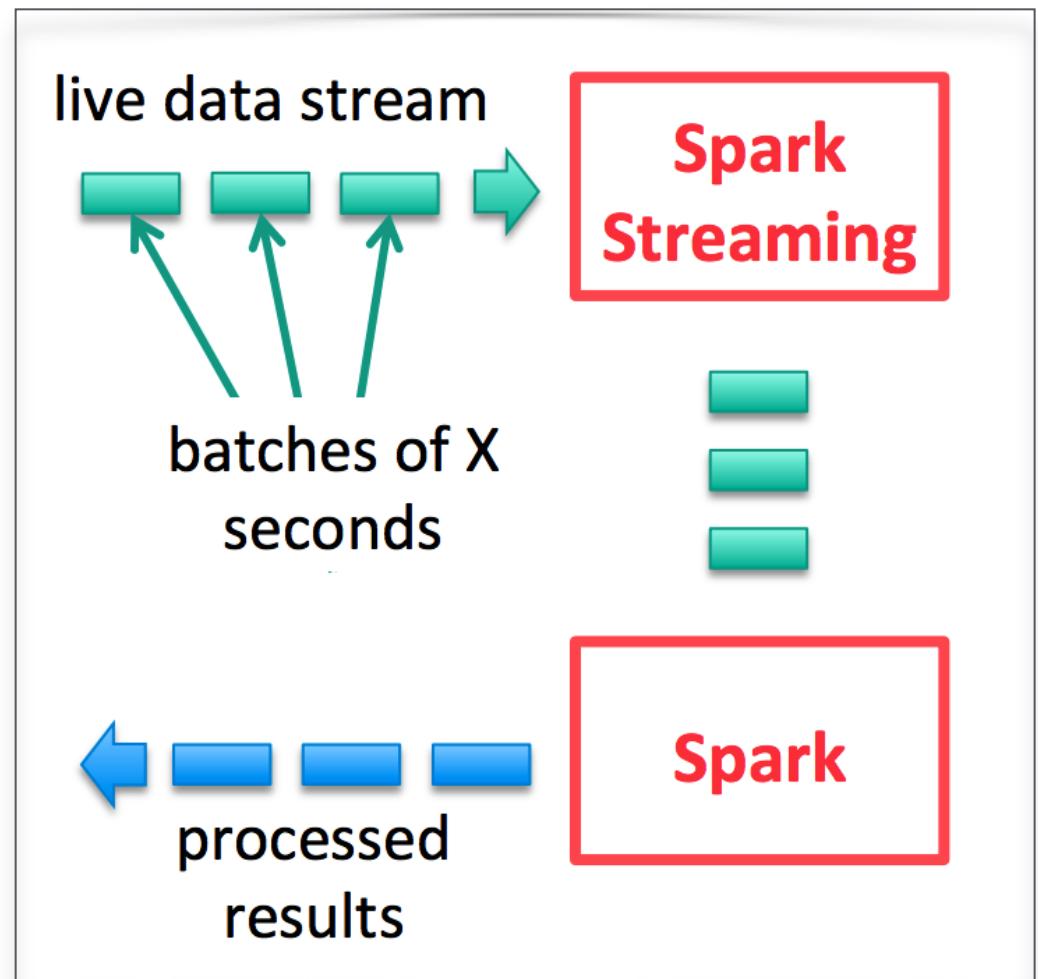
- *Chop up the live stream into batches of X seconds*
- *Spark treats each batch of data as RDDs and processes them using RDD operations*
- *Finally, the processed results of the RDD operations are returned in batches*



Spark Streaming: Requirements

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

- Batch sizes as low as $\frac{1}{2}$ sec, latency of about 1 sec
- Potential for combining batch processing and streaming processing in the same system



Spark Streaming: Integration

Data can be ingested from many sources:

Kafka, Flume, Twitter, ZeroMQ, TCP sockets, etc.

Results can be pushed out to filesystems,
databases, live dashboards, etc.

Spark's built-in machine learning algorithms and
graph processing algorithms can be applied to
data streams



Spark Streaming: Timeline

2012 project started

2013 alpha release (Spark 0.7)

2014 graduated (Spark 0.9)

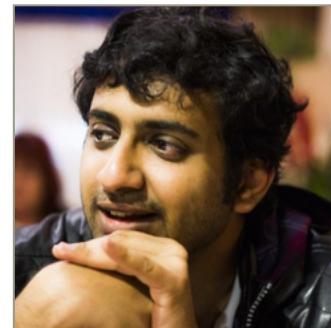
*Discretized Streams: A Fault-Tolerant Model
for Scalable Stream Processing*

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: Requirements

Typical kinds of applications:

- *datacenter operations*
- *web app funnel metrics*
- *ad optimization*
- *anti-fraud*
- *various telematics*

and much much more!

Spark Streaming: Some Excellent Resources

Programming Guide

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

TD @ Spark Summit 2014

youtu.be/o-NXwFrNAWQ?list=PLTPXxbhUt-YWGNTaDj6HSjnHMxiTDIHC

“Deep Dive into Spark Streaming”

[slideshare.net/spark-project/deep-divewithsparkstreaming-tathagatadassparkmeetup20130617](https://www.slideshare.net/spark-project/deep-divewithsparkstreaming-tathagatadassparkmeetup20130617)

Spark Reference Applications

databricks.gitbooks.io/databricks-spark-reference-applications/

Quiz: name the bits and pieces...

```
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()
```

Because
Use Cases

Because Use Cases: +40 known production use cases



sharethrough

NETFLIX



guavus



CISCO



PEARSON



kelkoo

viadeo



Asialinfo

stratio

OOYALA®



Because Use Cases: Analysis

Reasons for adopting/transitioning to Spark Streaming... the unified programming model is particularly relevant for real-time analytics that combine historical data:

- *Making data science accessible to non-scientists*
- *Higher productivity for data workers*
- *Exactly-once semantics*
- *No compromises on scalability and throughput*
- *Ease of operations*

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

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](http://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?

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

*Sharethrough Uses Spark Streaming to
Optimize Bidding in Real Time*

Russell Cardullo, Michael Ruggier

2014-03-25

[databricks.com/blog/2014/03/25/
sharethrough-and-spark-streaming.html](http://databricks.com/blog/2014/03/25/sharethrough-and-spark-streaming.html)



sharethrough

- the profile of a 24×7 streaming app is different than an hourly batch job...
- take time to validate output against the input...
- confirm that supporting objects are being serialized...
- the output of your Spark Streaming job is only as reliable as the queue that feeds Spark...
- monoids...

Because Use Cases: Viadeo

Spark Streaming As Near Realtime ETL

Djamel Zouaoui

2014-09-18

slideshare.net/DjamelZouaoui/spark-streaming



- Spark Streaming is topology-free
- workers and receivers are autonomous and independent
- paired with Kafka, RabbitMQ
- 8 machines / 120 cores
- use case for recommender system
- issues: how to handle lost data, serialization

Demos

Demos:

brand new Python support for Streaming in 1.2

github.com/apache/spark/tree/master/examples/src/main/python/streaming

For more Spark learning resources online:

databricks.com/spark-training-resources

Demo: PySpark Streaming Network Word Count

```
import sys
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

sc = SparkContext(appName="PyStreamNWC", master="local[*]")
ssc = StreamingContext(sc, Seconds(5))

lines = ssc.socketTextStream(sys.argv[1], int(sys.argv[2]))

counts = lines.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a+b)

counts.pprint()

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

Demo: PySpark Streaming Network Word Count - Stateful

```
import sys
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

def updateFunc (new_values, last_sum):
    return sum(new_values) + (last_sum or 0)

sc = SparkContext(appName="PyStreamNWC", master="local[*]")
ssc = StreamingContext(sc, Seconds(5))
ssc.checkpoint("checkpoint")

lines = ssc.socketTextStream(sys.argv[1], int(sys.argv[2]))

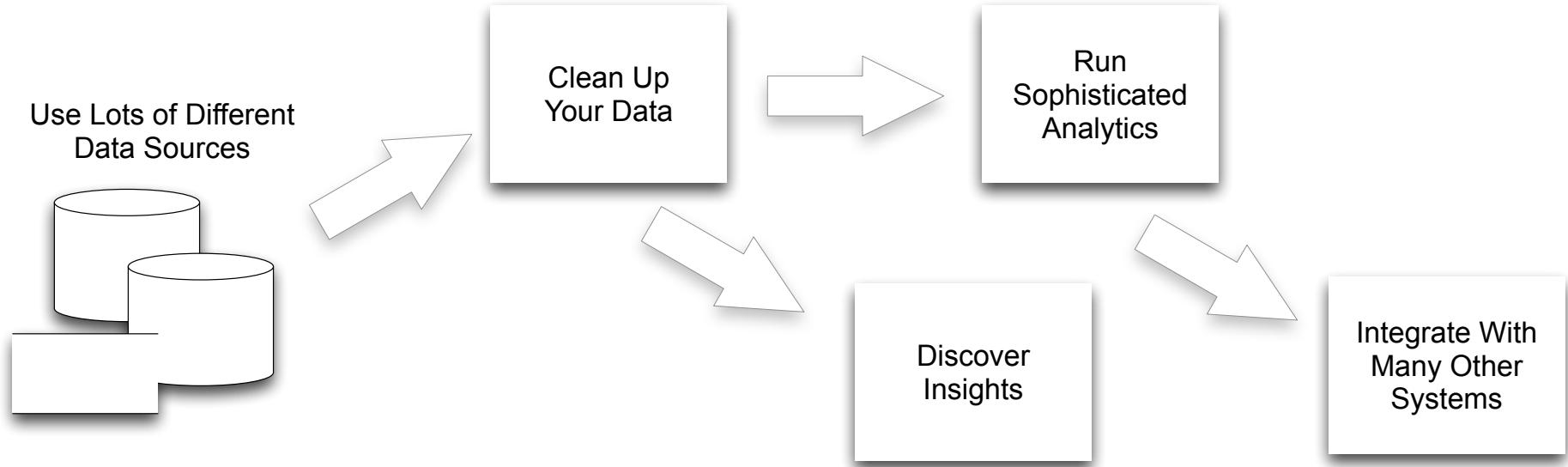
counts = lines.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .updateStateByKey(updateFunc) \
    .transform(lambda x: x.sortByKey())

counts.pprint()

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

Complementary Frameworks

Spark Integrations:



cloud-based notebooks... ETL... the Hadoop ecosystem...
widespread use of PyData... advanced analytics in streaming...
rich custom search... web apps for data APIs...
low-latency + multi-tenancy...

Spark Integrations: Advanced analytics for streaming use cases

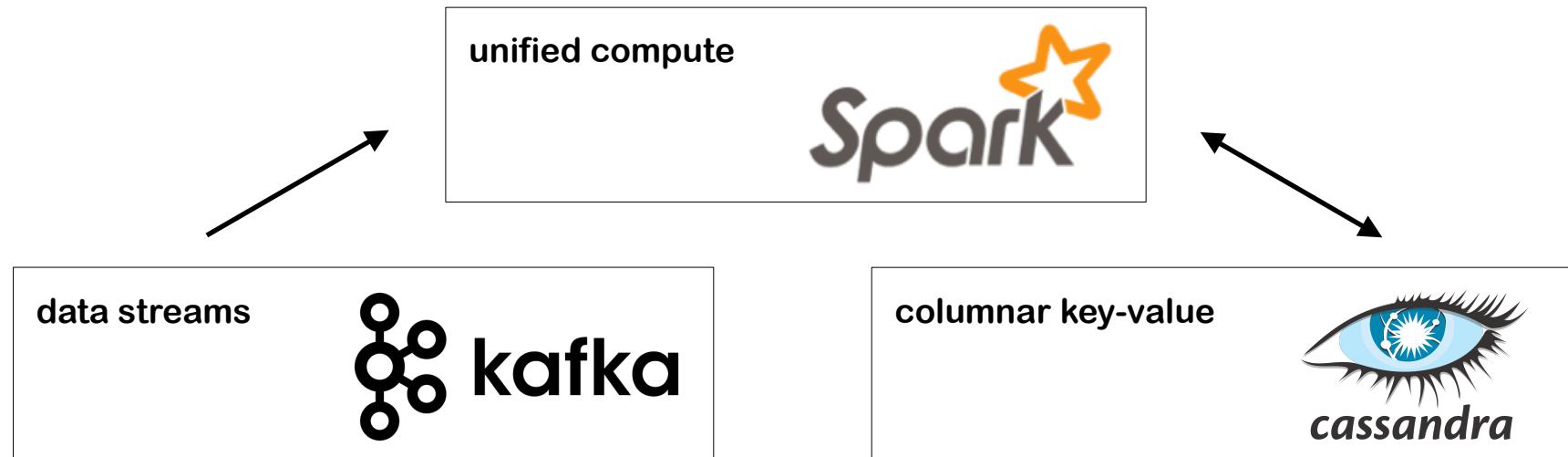
Kafka + Spark + Cassandra

[datastax.com/documentation/datastax_enterprise/4.5/
datastax_enterprise/spark/sparkIntro.html](http://datastax.com/documentation/datastax_enterprise/4.5/datastax_enterprise/spark/sparkIntro.html)

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

github.com/datastax/spark-cassandra-connector

github.com/dibbhatt/kafka-spark-consumer



Spark 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



Spark Integrations: General Guidelines

- use **Tachyon** as a best practice for sharing between two streaming apps
- or write to Cassandra or HBase / then read back
- design patterns for integration:
spark.apache.org/docs/latest/streaming-programming-guide.html#output-operations-on-dstreams

A Look Ahead...

A Look Ahead...

I. Greater Stability and Robustness

- *improved high availability via write-ahead logs*
- *enabled as an optional feature for Spark 1.2*
- *NB: Spark Standalone can already restart driver*
- *excellent discussion of fault-tolerance (2012):*
cs.duke.edu/~kmoses/cps516/dstream.html
- *stay tuned:*
meetup.com/spark-users/events/218108702/

A Look Ahead...

2. Support for more environments, i.e., beyond Hadoop

- *three use cases currently depend on HDFS*
- *those are being abstracted out*
- *could then use Cassandra, etc.*

A Look Ahead...

3. Improved support for Python

- e.g., *Kafka is not exposed through Python yet (next release goal)*

A Look Ahead...

4. Better flow control

- *a somewhat longer-term goal, plus it is a hard problem in general*
- *poses interesting challenges beyond what other streaming systems have faced*

A Big Picture

A Big Picture...



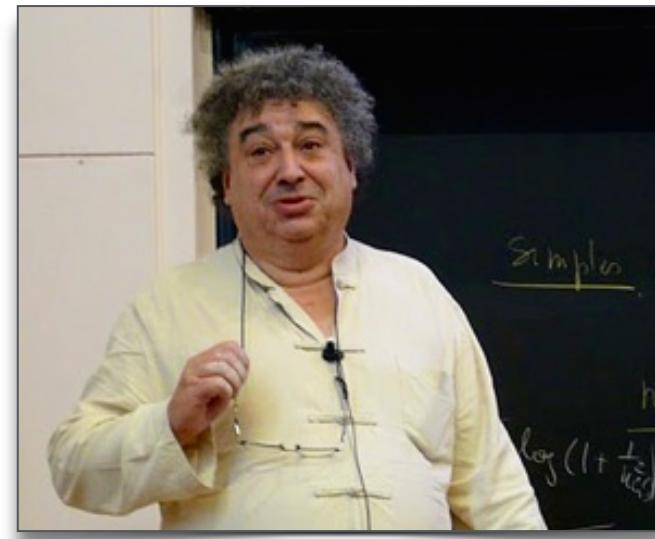
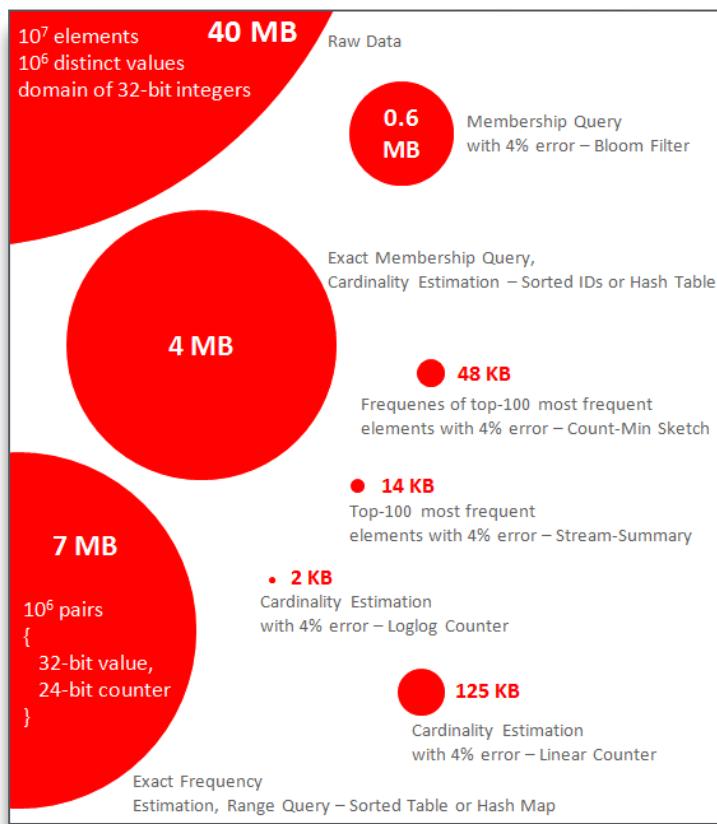
19-20c. statistics emphasized *defensibility* in lieu of *predictability*, based on analytic variance and goodness-of-fit tests

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

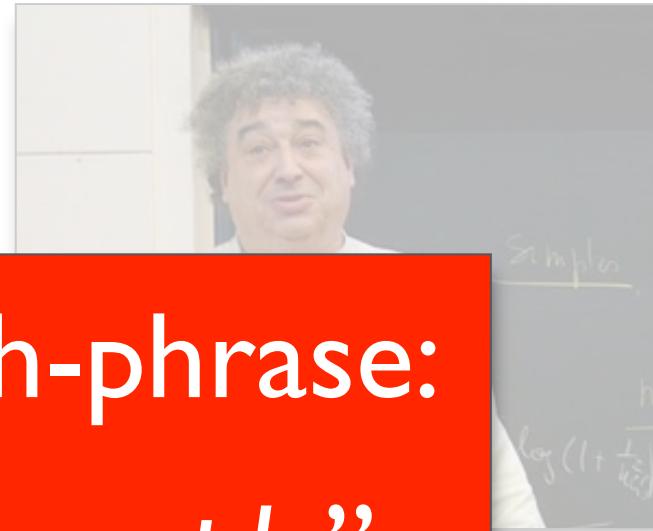
21c. shift towards modeling based on probabilistic approximations: trade bounded errors for greatly reduced resource costs



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

<i>algorithm</i>	<i>use case</i>	<i>example</i>
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	suggestion: consider these as your most quintessential collections data types at scale	
MinHash	streaming quantiles	
DSQ	ordered sequence search	
SkipList		

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](https://twitter.com/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

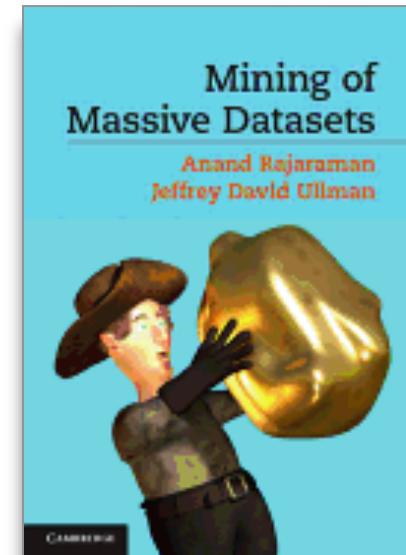
C. Titus Brown, O'Reilly (2010-11-10)

Algebird

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

Mining of Massive Datasets

Jure Leskovec, Anand Rajaraman, Jeff Ullman, Cambridge (2011)

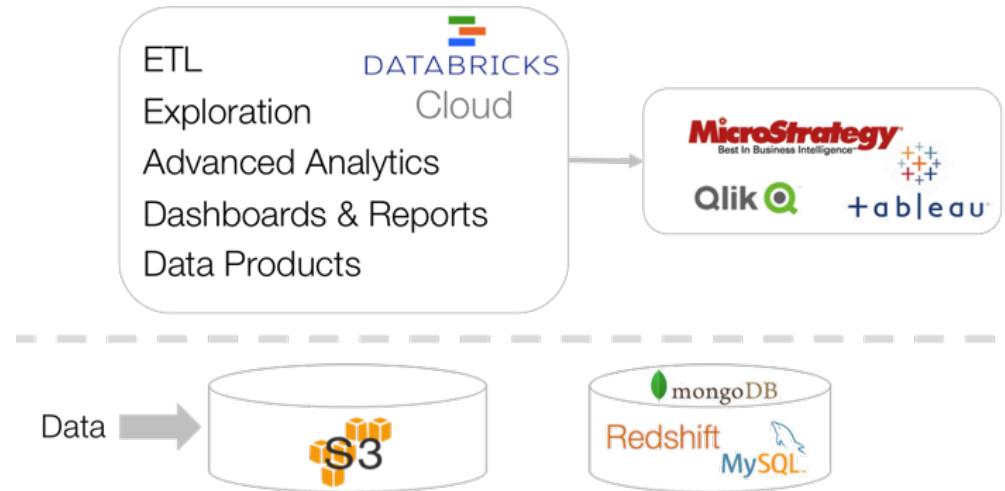


Resources

cloud-based notebooks:

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

[youtube.com/watch?v=dJQ5IV5Tldw#t=883](https://www.youtube.com/watch?v=dJQ5IV5Tldw#t=883)



certification:

Apache Spark developer certificate program

- <http://oreilly.com/go/sparkcert>
- defined by Spark experts @Databricks
- assessed by O'Reilly Media
- establishes the bar for Spark expertise



community:

spark.apache.org/community.html

video+slide archives: spark-summit.org

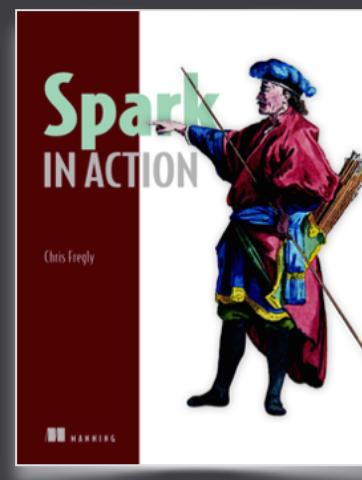
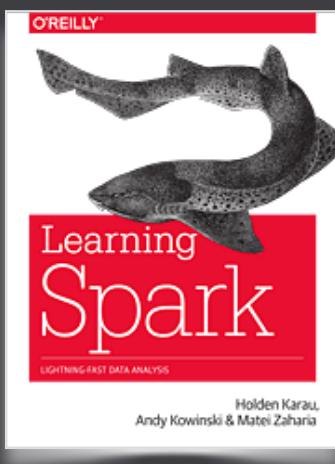
events worldwide: goo.gl/2YqJZK

resources: databricks.com/spark-training-resources

workshops: databricks.com/spark-training

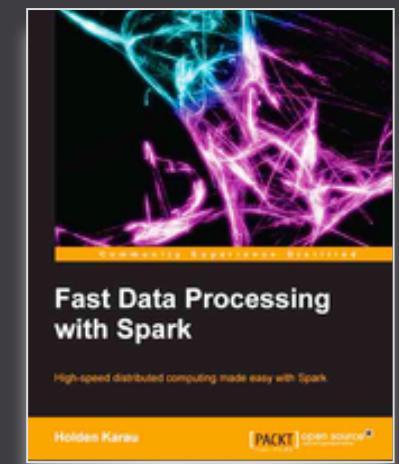
books:

Learning Spark
**Holden Karau,
Andy Konwinski,
Matei Zaharia**
O'Reilly (2015*)
[shop.oreilly.com/product/
0636920028512.do](http://shop.oreilly.com/product/0636920028512.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](http://shop.oreilly.com/product/9781782167068.do)



events:

Strata EU

Barcelona, Nov 19-21

strataconf.com/strataeu2014

Data Day Texas

Austin, Jan 10

datadaytexas.com

Strata CA

San Jose, Feb 18-20

strataconf.com/strata2015

Spark Summit East

NYC, Mar 18-19

spark-summit.org/east

Strata EU

London, May 5-7

strataconf.com/big-data-conference-uk-2015

Spark Summit 2015

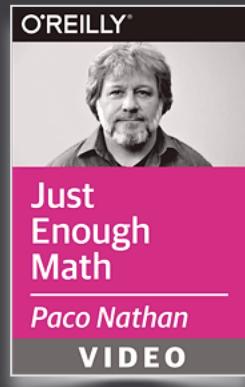
SF, Jun 15-17

spark-summit.org

presenter:

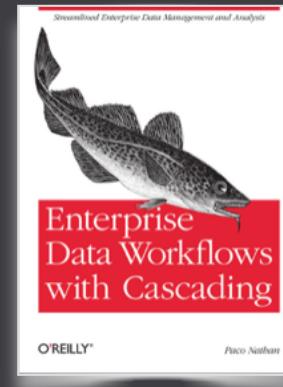
monthly newsletter for updates,
events, conf summaries, etc.:

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](http://shop.oreilly.com/product/0636920028536.do)