# Distributed Computing with Apache Spark

# What is Spark?

Spark is
a better implementation
of the
MapReduce paradigm
for Big Data

# Why Spark?

# Oh wait...

# What is Big Data?
# What is MapReduce?

# What do people mean when they say the words "Big Data" ???

(Bad joke coming up on the next slide.)

BIG

DATA

**Big Data**
**is data that exceeds**
**the processing capacity**
**of conventional**
**database systems**

# Big Data
# is defined by the three Vs

- volume
- velocity
- variety

# Big Data
# There is overwhelmingly too much of it

## To the point where the solution you developed for small datasets performs very poorly

# (volume)

# Big Data
# Comes to you in the form of an unrelenting stream at firehose speeds (velocity)

Big Data
Is data for which
the well-understood
structure
of a traditional
database architecture
is not a good fit
(variety)

**Therefore,**
**to gain value from Big Data,**
**you must choose**
**an alternative way**
**to process it**

# Year 2003:
# MapReduce to the rescue!

# Data analytics at scale was enabled by two big ideas:

a framework for
distributed storage (HDFS)

a framework for
distributed computing (MapReduce)

# Knowledge vs. Insight

# Knowledge comes by taking things apart (map)

# Knowledge vs. Insight

Insight comes
by inventing ways
to put those pieces
back together
(reduce)

MapReduce
should have been called
ExplodeRecombine
(because that's what it is)

Until recently,
Hadoop was
the standard tool
for distributed computing
across really large data sets

# Hadoop

is an operating system for Big Data

is a rich ecosystem of
tools and techniques

allows engineers to build clusters
from commodity hardware
and do computing
at supercomputer scale

Big Data processing
went from
prohibitively expensive
in the pioneering days,
to feasible for even
the smallest garage startups,
who can cheaply rent server
time in the cloud

Hadoop
became successful
when spinning media storage
became cheap enough

# Powerful force #1: research has shifted focus toward generalizations of distributed computation, expanding on the ideas first imagined in MapReduce

# Powerful force #2:

# RAM
# has also become
# cheap enough

Hadoop's
most annoying bottleneck
nowadays
comes from being designed
for disk-to-disk-to-disk
data processing pipelines

# Obvious idea:

# How about a RAM-to-RAM-to-RAM pipeline?

# Meet Spark

# Apache Spark

is an
open-source
cluster computing framework
with in-memory processing
to make analytics applications
up to 100 times faster
compared to

# Hadoop

Spark
was designed
from the ground up
to support
the low-latency use cases
that Hadoop
traditionally struggled with

# Spark

combines
an engine for distributing programs
across clusters of machines
with
an elegant programming model
for writing programs on top of it
and a set of libraries
that can help you
do cool and powerful things with

# Big Data

# Spark

is finally an open source framework
that allows a data scientist
to be productive
with large data sets
by making distributed programming
truly accessible

# Spark's promise

is to make writing distributed programs feel like writing regular programs

# Spark's execution engine achieves near linear scalability

# Linear scalability

**As the data size increases,**
**we can throw more computers at it**
**and see jobs complete**
**in the same amount of time**

# Spark

**is resilient to the fact**
**that failures which**
**occur rarely on a single machine**
**occur all the time**
**on clusters of thousands of machines**

# Spark

breaks up work
into small tasks
and can gracefully accommodate
task failures
without compromising
the job to which they belong

# Spark

is well suited
for highly interactive
data applications
that quickly respond
to user queries
by scanning
large in-memory data sets

# Spark's fundamental abstraction

the Resilient Distributed Dataset (RDD)
allows developers
to materialize
any point in a processing pipeline
into memory
across the cluster

In this way,
future steps
that want to deal
with the same data set
do not need
to recompute it
or reload it from disk

# An RDD

is essentially
a read-only collection of objects
that are partitioned
across machines

# An RDD

can be operated on
in a parallel manner
using Spark's built-in set of
over 80 high-level operators,
enabling users
to express computation steps
in a natural fashion,
using a functional programming mindset

# An RDD

can be created at the shell prompt

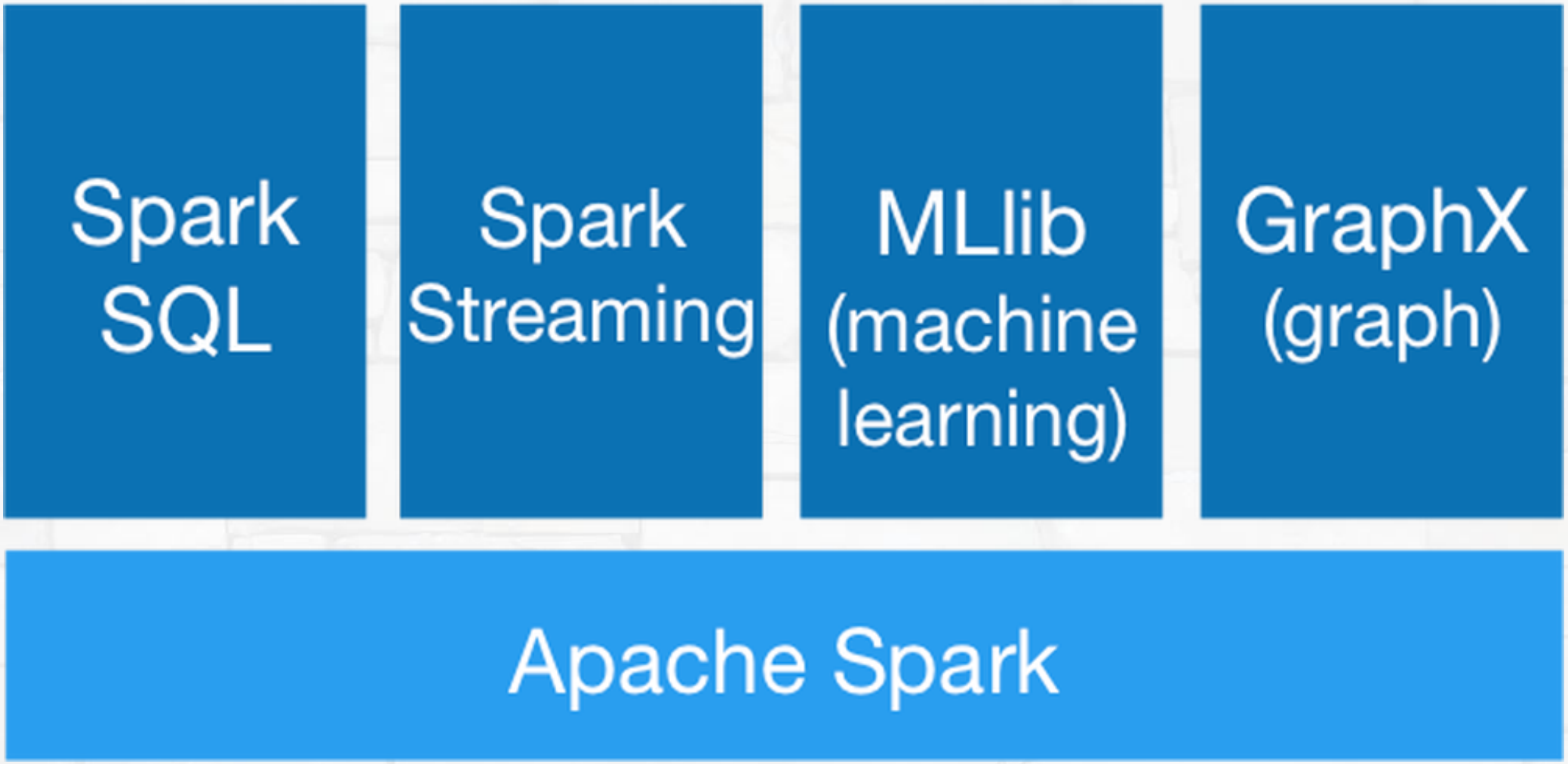
# An RDD

**can be interactively queried,**
**sliced and diced at the shell prompt**
**using more than 80 operators**

# Did I mention it's fun to program in Spark?

# The RDD API is highly focused on the needs of its users (the data engineers)

Spark's vision
is to make the transition
from working
on a single machine
to working on a cluster,
a seamless experience


Spark SQL
Spark Streaming
MLlib (machine learning)
GraphX (graph)
Apache Spark

# Demo time!

# Follow me on GitHub

# github.com/dserban