Distributed Computing with Apache Spark

Introduction to Big Data Concepts with Apache 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 technologies in wide deployment today. Developed in the AMPLab at UC Berkeley, Spark can help reduce data interaction complexity, increase processing speed and enhance data-intensive, real-time applications with deep intelligence.

Highly versatile in many environments, and with a strong foundation in functional programming, Spark is known for its ease of use in creating algorithms that harness insight from complex data. Spark was elevated to a top-level Apache Project in 2014 and continues to expand today.

Spark is one of the biggest reasons that Big Data is such an exciting area of work for technologists right now

Spark is hugely popular

It has the most active community of any open source big data project currently in development

This presentation is an overview of Spark

Disclaimer
I do not use Spark in a production setting

The source material for this presentation comes from my own Spark learning experience as well as my own knowledge of data engineering topics in general

But first ...

What is Big Data?

What do people mean when they say the words "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!

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)

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

Year 2015:

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

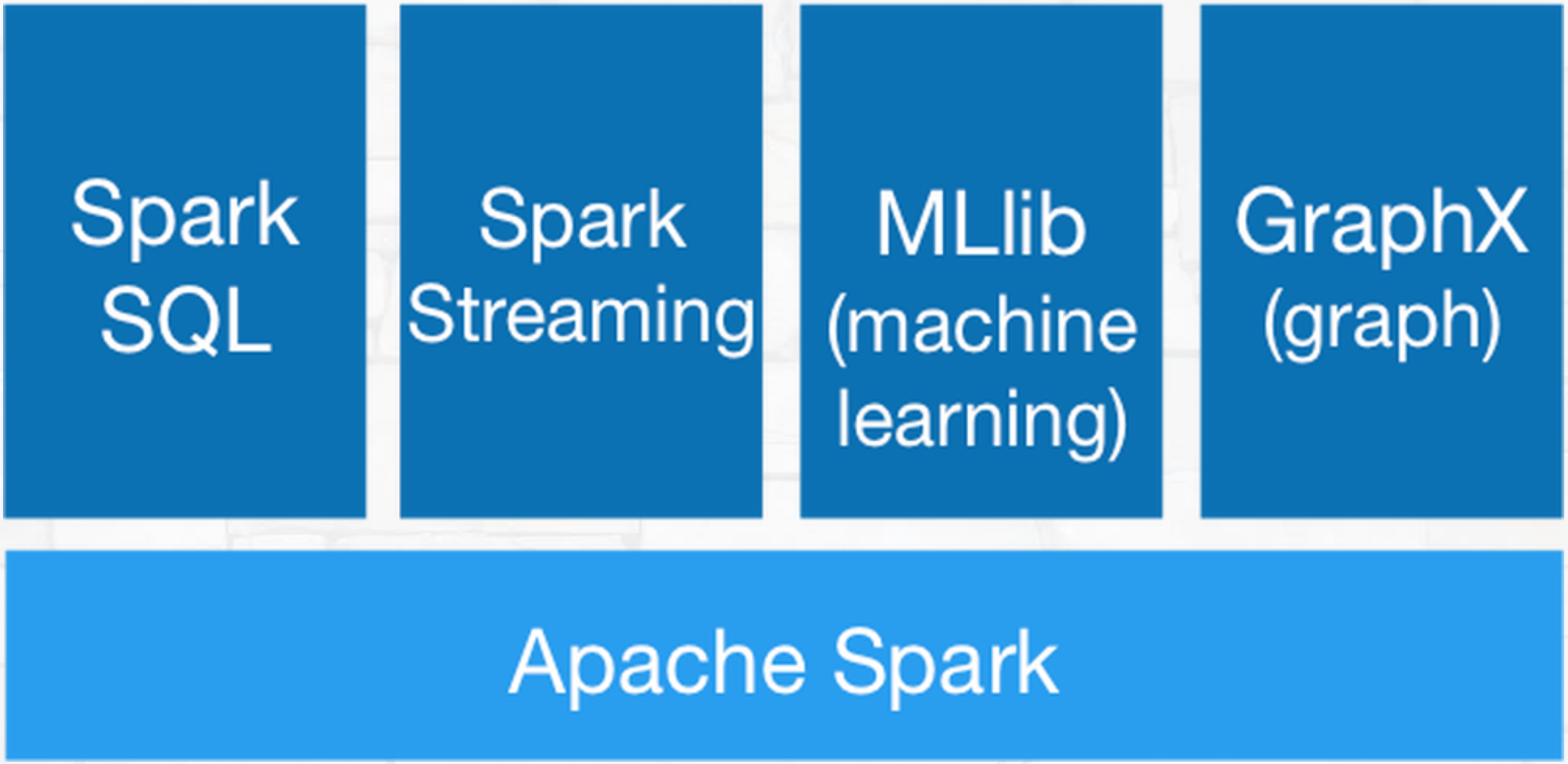
Spark
was designed
from the ground up
to support
the low-latency use cases
that Hadoop
traditionally struggled with

Spark
provides
a distributed execution engine,
a fun-to-work-with programming model
and a set of libraries
that can help you
do cool and powerful things with
Big Data

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

The 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