UNIVERSITY OF WATERLOO

UROC 2017

September 22-23, 2017

Jimmy Lin and Ihab Ilyas
David R. Cheriton School of Computer Science
University of Waterloo

Agenda Today

Introductions

What do you want to get out of this session today?

Overview of big data and data science

Blah blah blah…

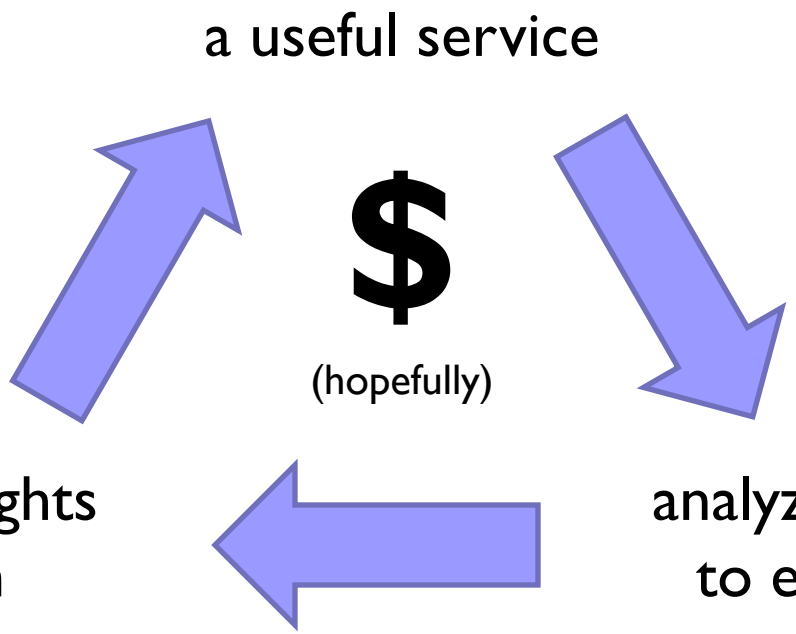
Crash course on Spark and Scala

Cluster walkthrough

Choose your own adventure!

What?

Data-Driven Consumer Companies



Google. Facebook. Twitter. Amazon. Uber.

data products **data science**

Google™ Processes 20 PB a day (2008)
Crawls 20B web pages a day (2012)
Search index is 100+ PB (5/2014)
Bigtable serves 2+ EB, 600M QPS (5/2014)

INTERNET ARCHIVE 400B pages, 10+ PB (2/2014)

YAHOO! 19 Hadoop clusters: 600 PB, 40k servers (9/2015)

JPMorganChase 150 PB on 50k+ servers running 15k apps (6/2011)

ebay Hadoop: 10K nodes, 150K cores, 150 PB (4/2014)

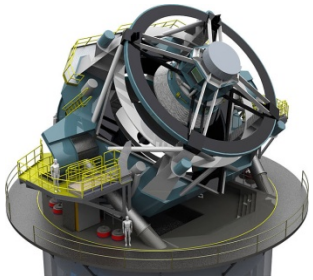
300 PB data in Hive + 600 TB/day (4/2014)

facebook

amazon web services™ S3: 2T objects, 1.1M request/second (4/2013)

LHC: ~15 PB a year

CERN



LSST: 6-10 PB a year (~2020)

SKA: 0.3 – 1.5 EB per year (~2020)

How much data?

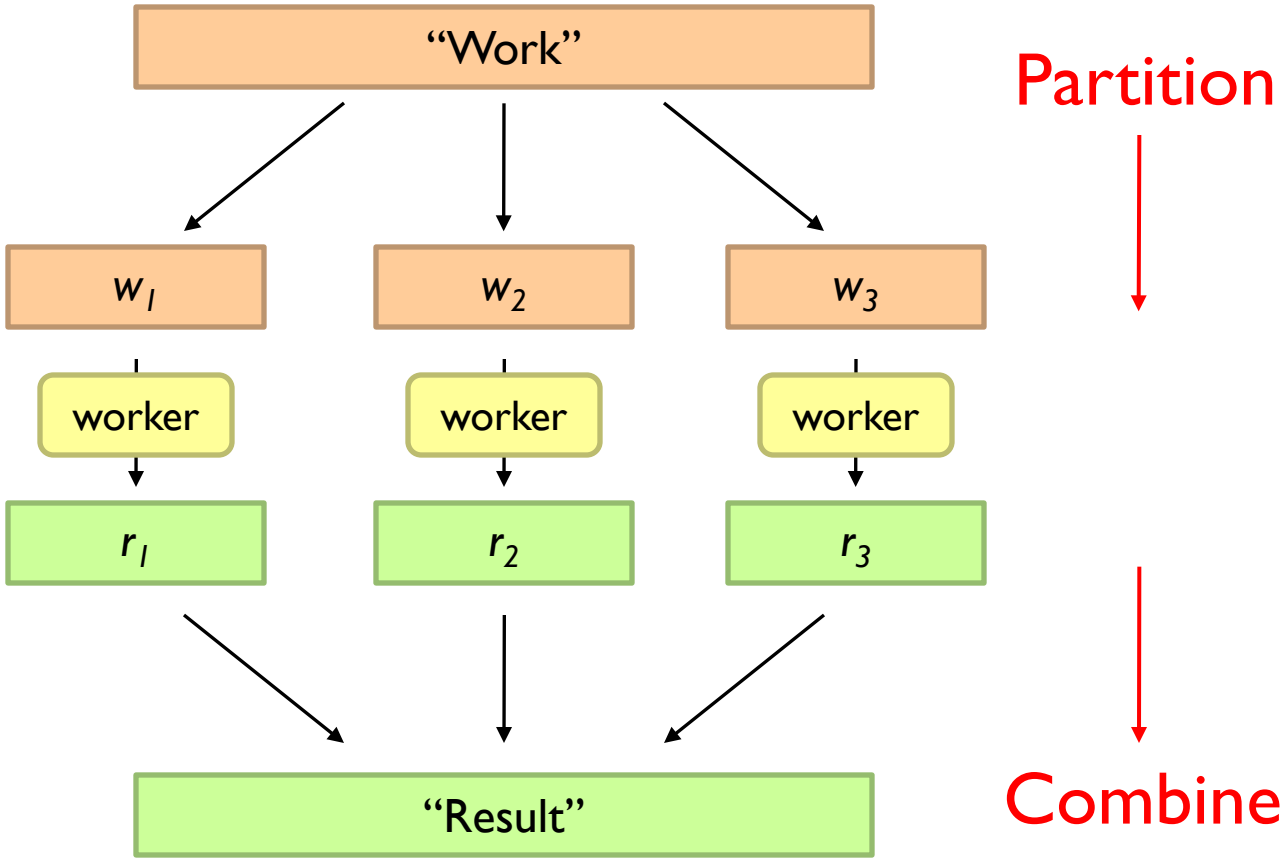
CADE METZ BUSINESS 01.27.16 01:00 PM

IN A HUGE BREAKTHROUGH, GOOGLE'S AI BEATS A TOP PLAYER AT THE GAME OF GO

GOOGLE

How?

Divide and Conquer



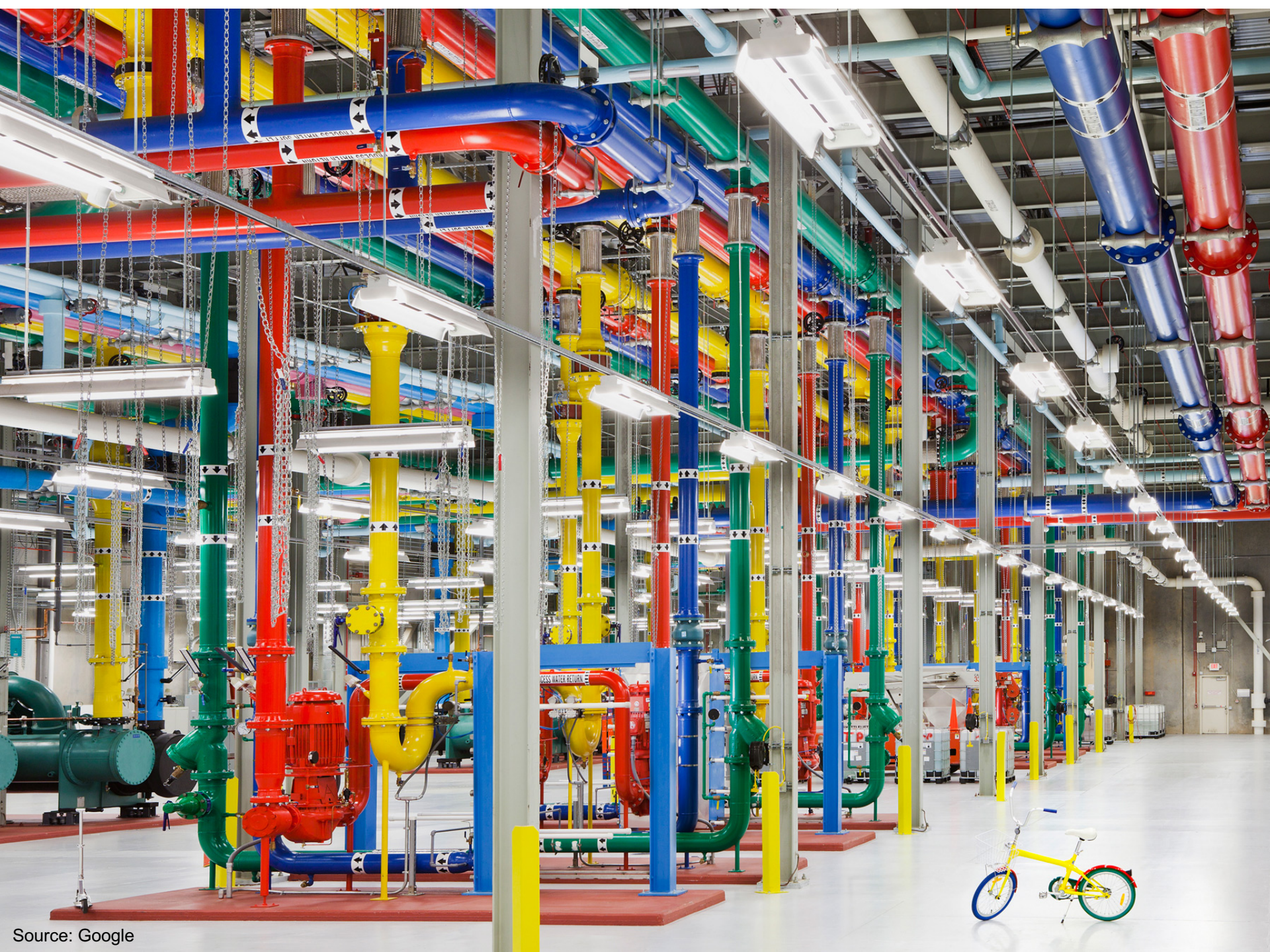

Source: Google

Source: Google

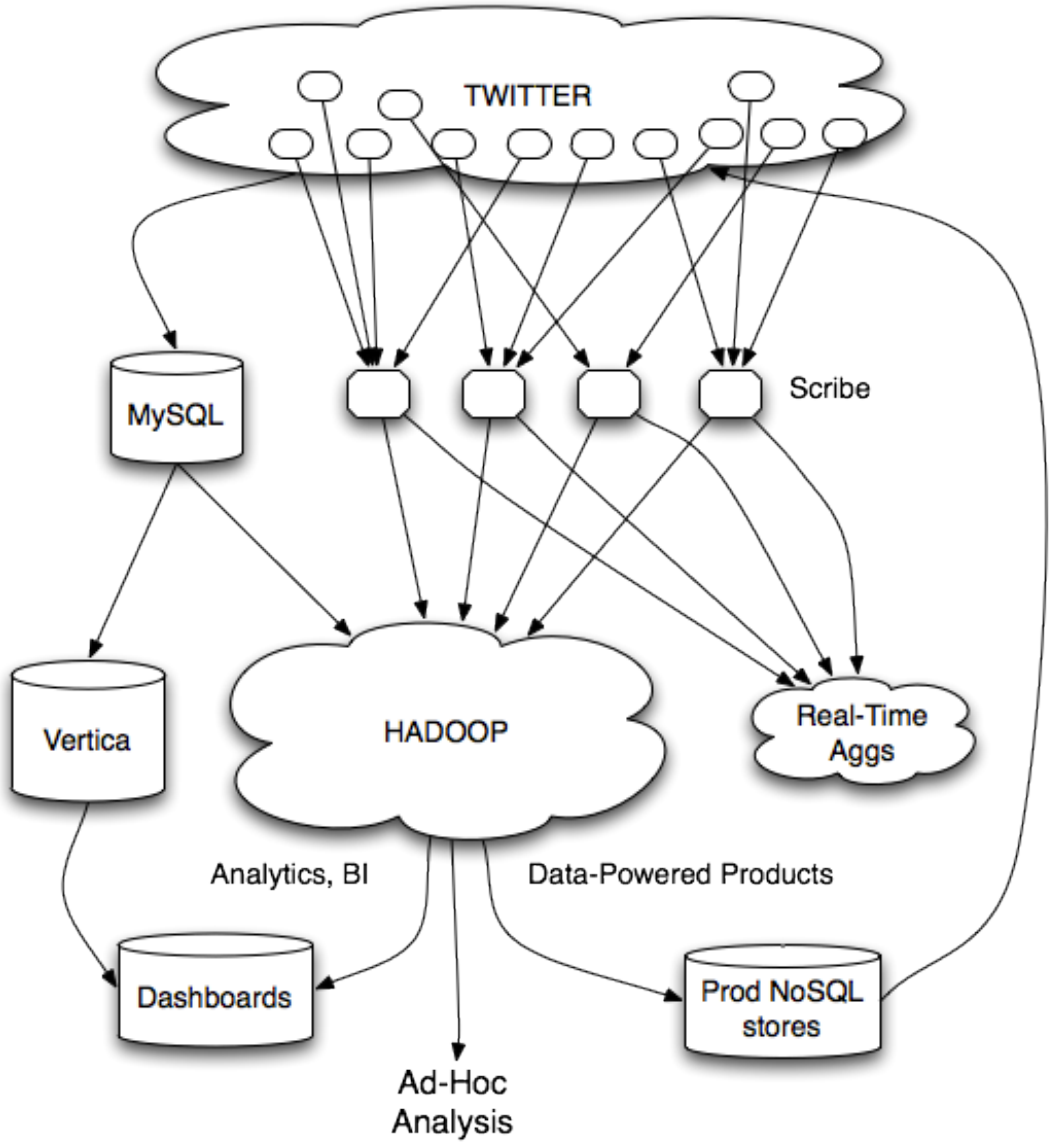
Source: Google

Source: Google

Source: Google



Source: Google



Twitter's data warehousing architecture (circa 2012)

How Exactly?

Typical Big Data Problem

Iterate over a large number of records

Extract something of interest from each

Shuffle and sort intermediate results

Aggregate intermediate results

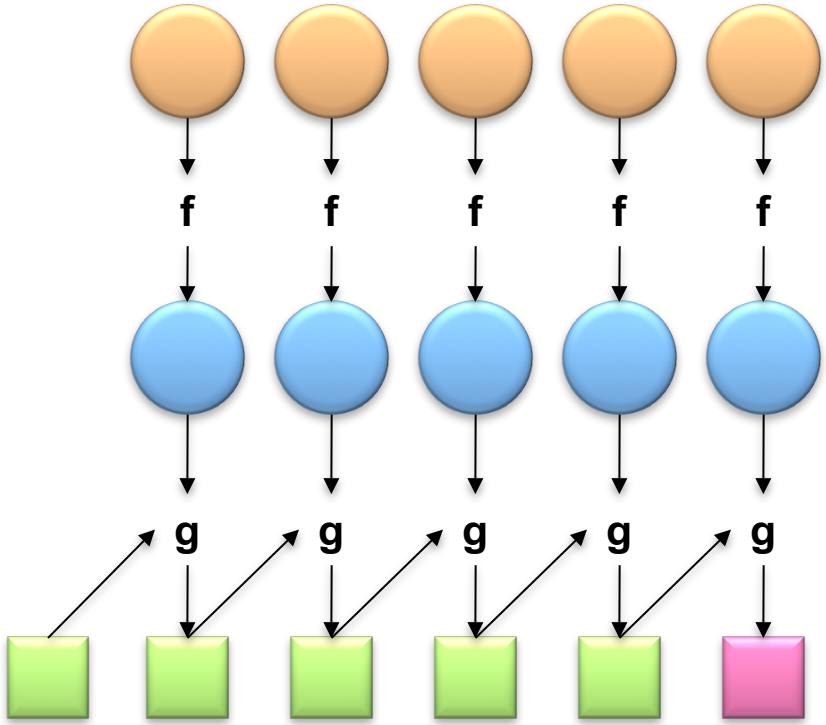
Generate final output

Key idea: data parallelism!

Roots in Functional Programming

Map

Fold



Functional Programming in Scala

```
scala> val t = Array(1, 2, 3, 4, 5)
t: Array[Int] = Array(1, 2, 3, 4, 5)

scala> t.map(n => n*n)
res0: Array[Int] = Array(1, 4, 9, 16, 25)

scala> t.map(n => n*n).foldLeft(0)((m, n) => m + n)
res1: Int = 55
```

Spark = functional programming + distributed computing!

What's Spark?

a large-scale data processing platform
implemented in Scala

What's Scala?

It's like Java, but for hipsters

BTW, you can use Spark with Scala, Python, and R.
(Scala is much faster, and one less dependency to manage.)

Source: https://www.flickr.com/photos/cmichel67/9428305104/

Crash Course on Scala

Type ‘scala’ to enter Scala shell

```
scala> val x = 1
x: Int = 1

scala> var y = x + 1
y: Int = 2

scala> var y = x + 5
y: Int = 6

scala> x = 2
<console>:27: error: reassignment to val
       x = 2
```

vals and vars, oh my!

Crash Course on Scala

Type ‘scala’ to enter Scala shell

```
scala> val f = (r: Int) => r * r
f: Int => Int = <function1>

scala> f(2)
res1: Int = 4
```

Remember =>
means a function!

Crash Course on Scala

Type 'scala' to enter Scala shell

```
scala> val r = (1, 2, 3, "a", "b", (4, 6))
r: (Int, Int, Int, String, String, (Int, Int))
= (1,2,3,a,b,(4,6))

scala> r._1
res2: Int = 1

scala> r._6
res3: (Int, Int) = (4,6)

scala> r._6._2
res4: Int = 6
```

What's a tuple?
Collection of individual fields

Don't be scared of these funny underscores…

Crash Course on Scala

Type 'scala' to enter Scala shell

```
scala> val f = (r: Int) => (r, r+1, r+2)
f: Int => (Int, Int, Int) = <function1>

scala> f(1)
res5: (Int, Int, Int) = (1,2,3)

scala> f(2)
res6: (Int, Int, Int) = (2,3,4)
```

Guess what, it's common for functions to create tuples!

Crash Course on Spark

What's an RDD?

Resilient Distributed Dataset (RDD)

A (potentially) large collection of "stuff"

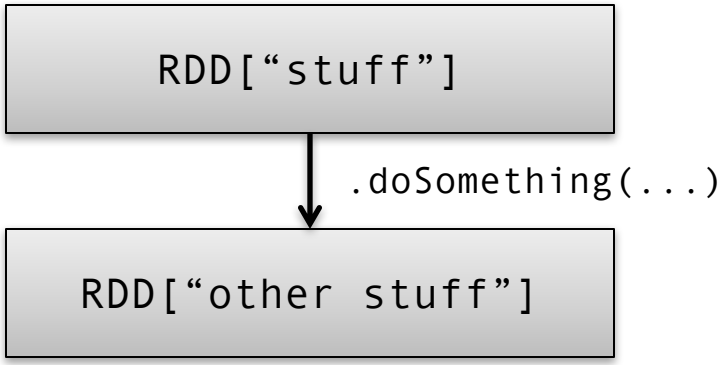
Denoted RDD["stuff"]

Where do they come from?

Initially, from disk

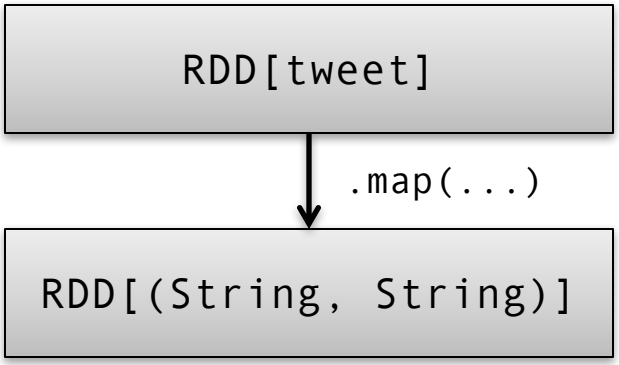
Transformations on RDDs

```
rdd.doSomething(...)
```



```
val t = rdd.map(r => (r.id, r.text))
```

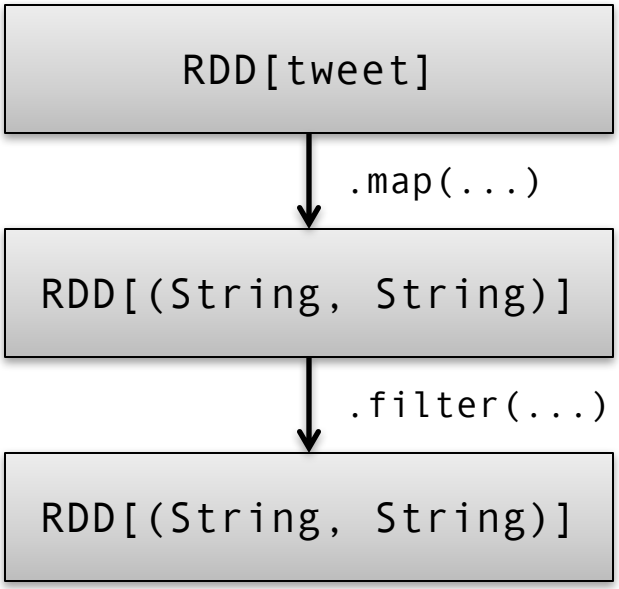
Remember => means a function,
(...) means a tuple



```
val t = rdd.map(r => (r.id, r.text))
          .filter(r => r._2.contains(“#maga"))
```

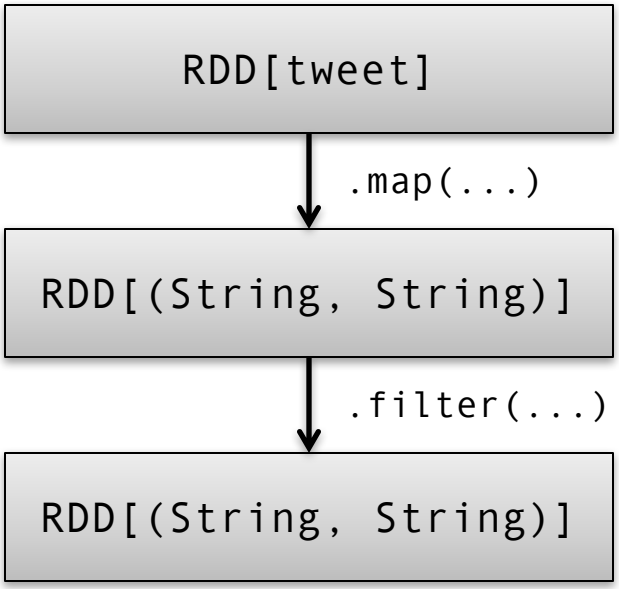
Don’t be scared of these funny underscores…

You can “chain” transformations



```
val t = rdd.map(r => (r.id, r.text))
          .filter(r => r._2.contains(“#maga"))

val results = t.collect()
```

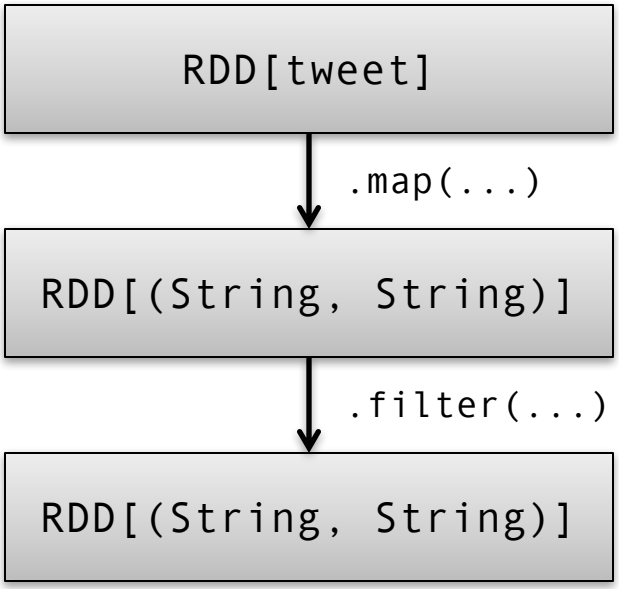
Actions actually produce results!



Action!

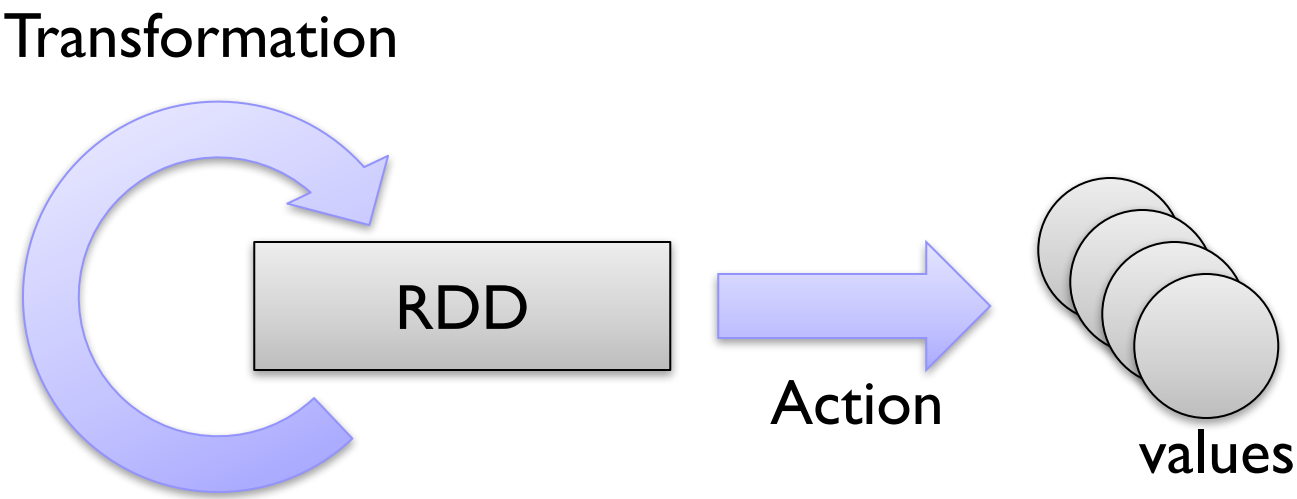
Boilerplate for loading data

```
val rdd = RecordLoader.loadTweets ("/path/to/tweets", sc)


val t = rdd.map(r => (r.id, r.text))
           .filter(r => r._2.contains(“#maga"))

val results = t.collect()
```



Action!

RDD Lifecycle



Transformations are lazy:
Framework keeps track of lineage

Actions trigger actual execution

The beauty of "scale out"

few GBs in a few minutes on your laptop
10s-100s GBs in 10s of minutes on a server
TBs in hours on a Spark cluster

All with the same script!
(Just throw more hardware at it…)

Let’s do some hacking!