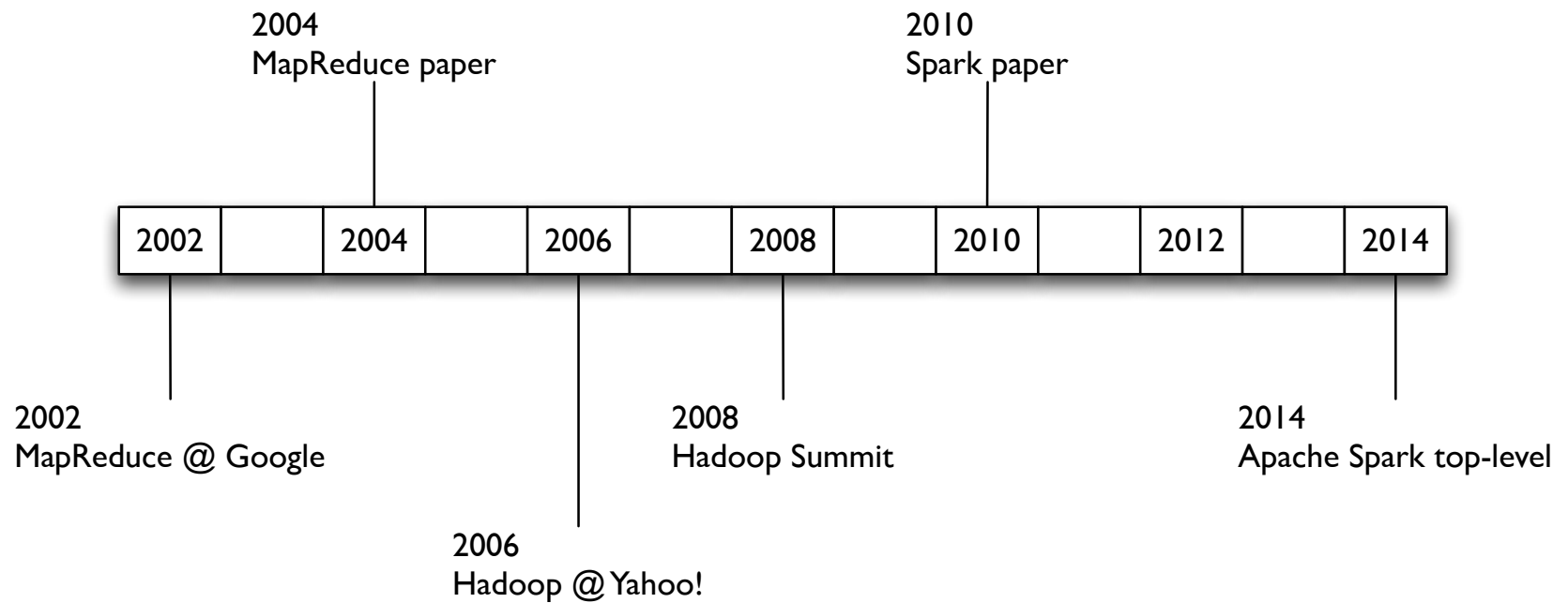


05: Getting Started

A Brief History

lecture: 35 min

A Brief History:



A Brief History: *MapReduce*

circa 1979 – Stanford, MIT, CMU, etc.

set/list operations in LISP, Prolog, etc., for parallel processing

www-formal.stanford.edu/jmc/history/lisp/lisp.htm

circa 2004 – Google

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

research.google.com/archive/mapreduce.html

circa 2006 – Apache

Hadoop, originating from the Nutch Project

Doug Cutting

research.yahoo.com/files/cutting.pdf

circa 2008 – Yahoo

web scale search indexing

Hadoop Summit, HUG, etc.

developer.yahoo.com/hadoop/

circa 2009 – Amazon AWS

Elastic MapReduce

Hadoop modified for EC2/S3, plus support for Hive, Pig, Cascading, etc.

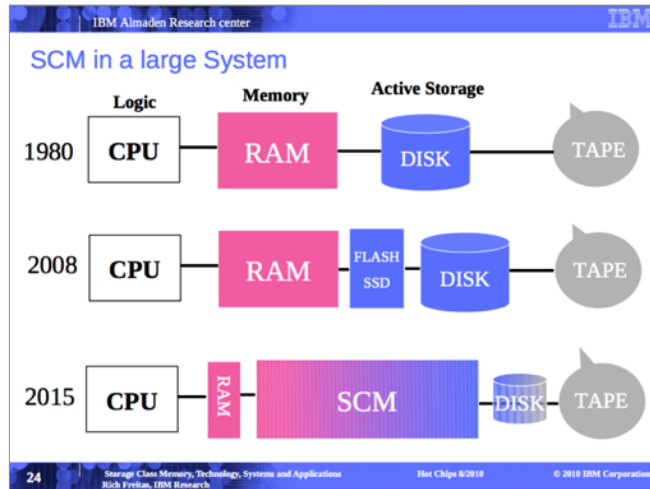
aws.amazon.com/elasticmapreduce/

A Brief History: *MapReduce*

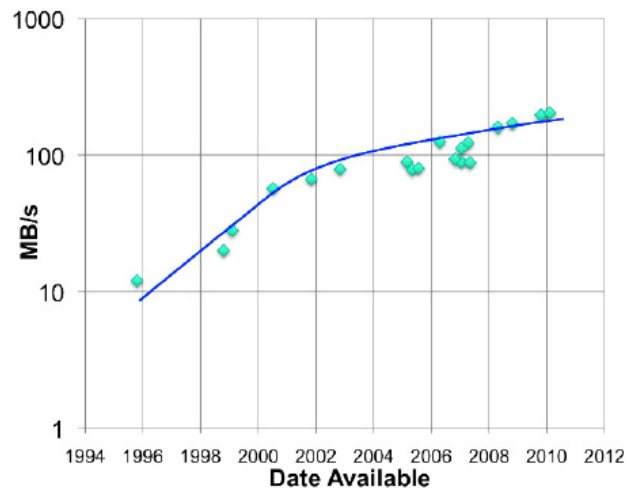
Open Discussion:

Enumerate several changes in data center technologies since 2002...

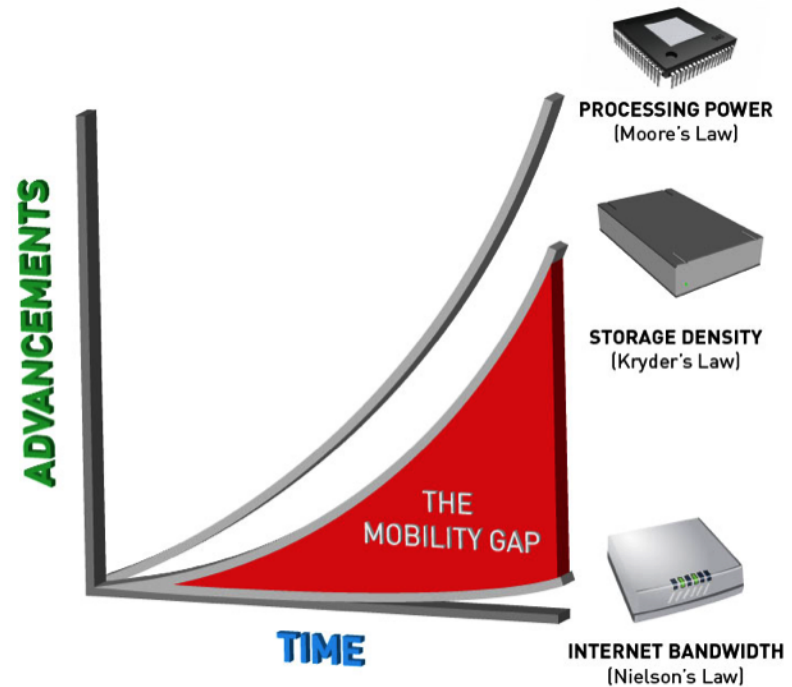
A Brief History: MapReduce



Rich Freitas, **IBM Research**



storagenewsletter.com/rubriques/hard-disk-drives/hdd-technology-trends-ibm/



pistoncloud.com/2013/04/storage-and-the-mobility-gap/

meanwhile, spinny disks haven't changed all that much...

A Brief History: *MapReduce*

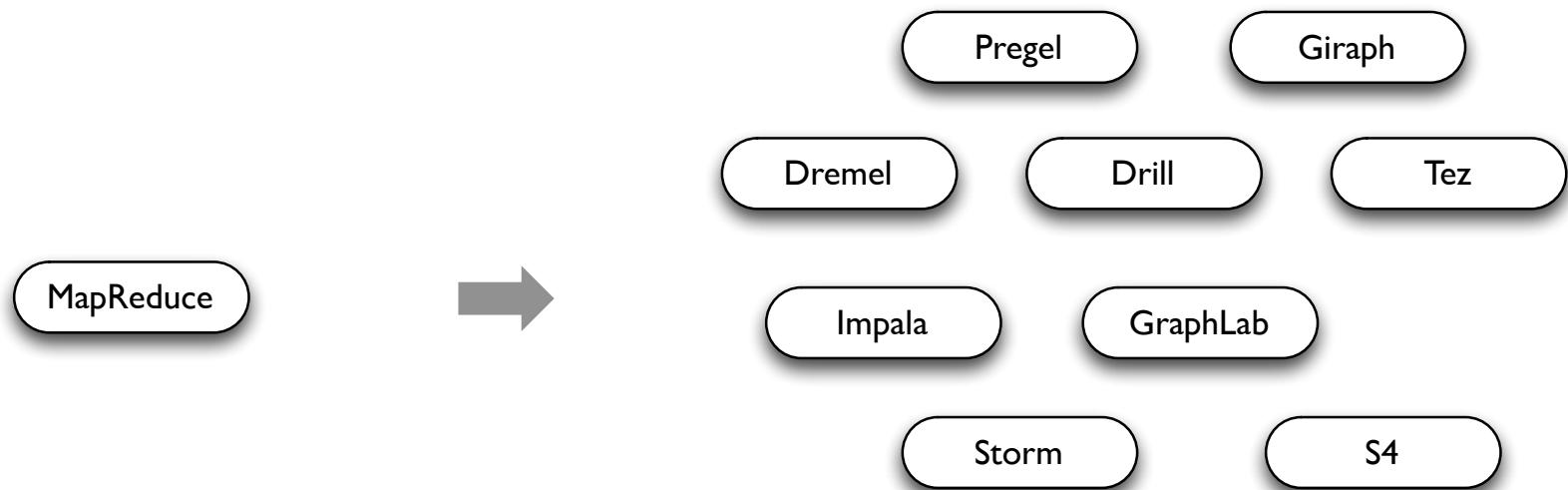
MapReduce use cases showed two major limitations:

1. difficulty of programming directly in MR
2. performance bottlenecks, or batch not fitting the use cases

In short, MR doesn't compose well for large applications

Therefore, people built *specialized systems* as workarounds...

A Brief History: *MapReduce*



General Batch Processing

Specialized Systems:

iterative, interactive, streaming, graph, etc.

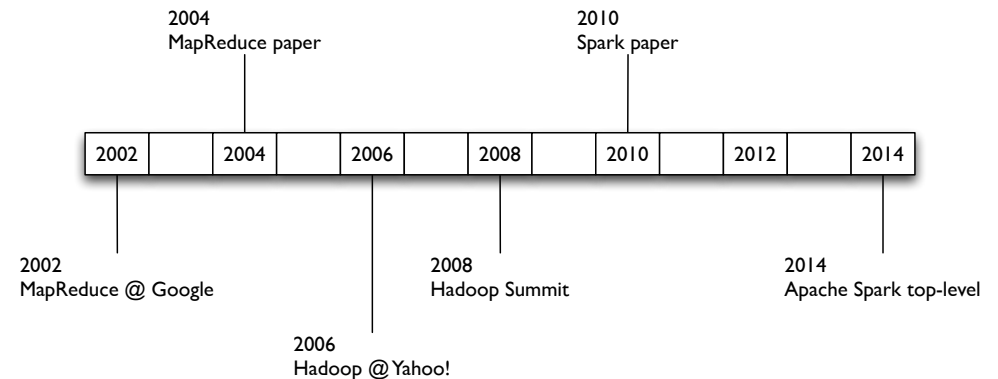
The State of Spark, and Where We're Going Next

Matei Zaharia

Spark Summit (2013)

youtu.be/nU6vO2EJAb4

A Brief History: *Spark*



Spark: Cluster Computing with Working Sets

Matei Zaharia, Mosharaf Chowdhury,
Michael J. Franklin, Scott Shenker, Ion Stoica
USENIX HotCloud (2010)

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 J. Franklin, Scott Shenker, Ion Stoica
NSDI (2012)

usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf

A Brief History: *Spark*

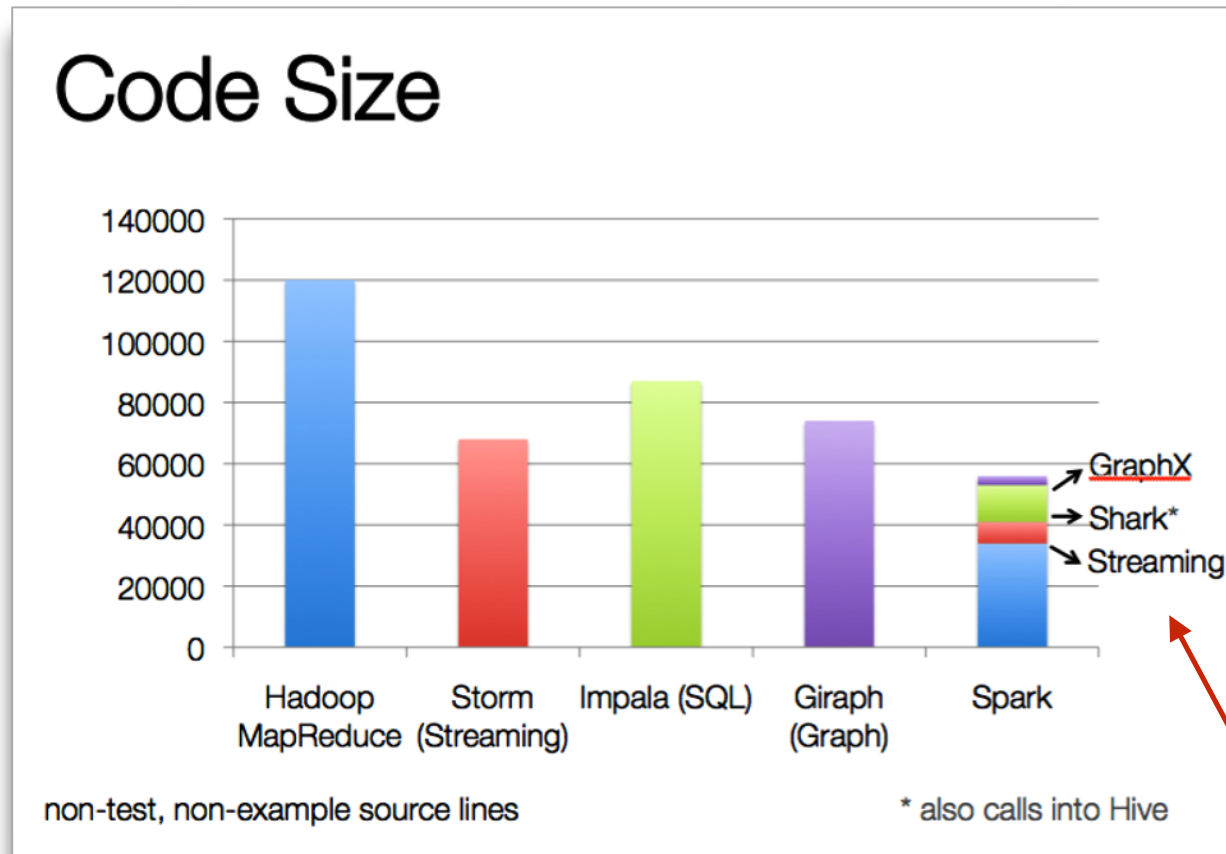
Unlike the various specialized systems, Spark's goal was to *generalize* MapReduce to support new apps within same engine

Two reasonably small additions are enough to express the previous models:

- *fast data sharing*
- *general DAGs*

This allows for an approach which is more efficient for the engine, and much simpler for the end users

A Brief History: *Spark*



The State of Spark, and Where We're Going Next

Matei Zaharia

Spark Summit (2013)

youtu.be/nU6vO2EJAb4

used as libs, instead of specialized systems

A Brief History: *Spark*

Some key points about Spark:

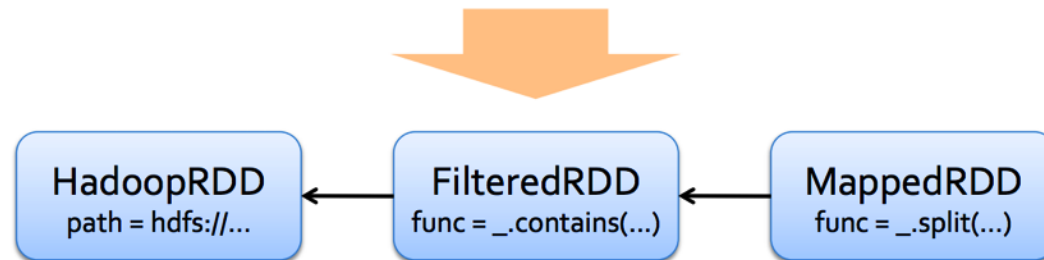
- handles batch, interactive, and real-time within a single framework
- native integration with Java, Python, Scala
- programming at a higher level of abstraction
- more general: map/reduce is just one set of supported constructs

A Brief History: *Spark*

RDD Fault Tolerance

RDDs track the series of transformations used to build them (their *lineage*) to recompute lost data

E.g: `messages = textFile(...).filter(_.contains("error")).map(_.split('\t')(2))`



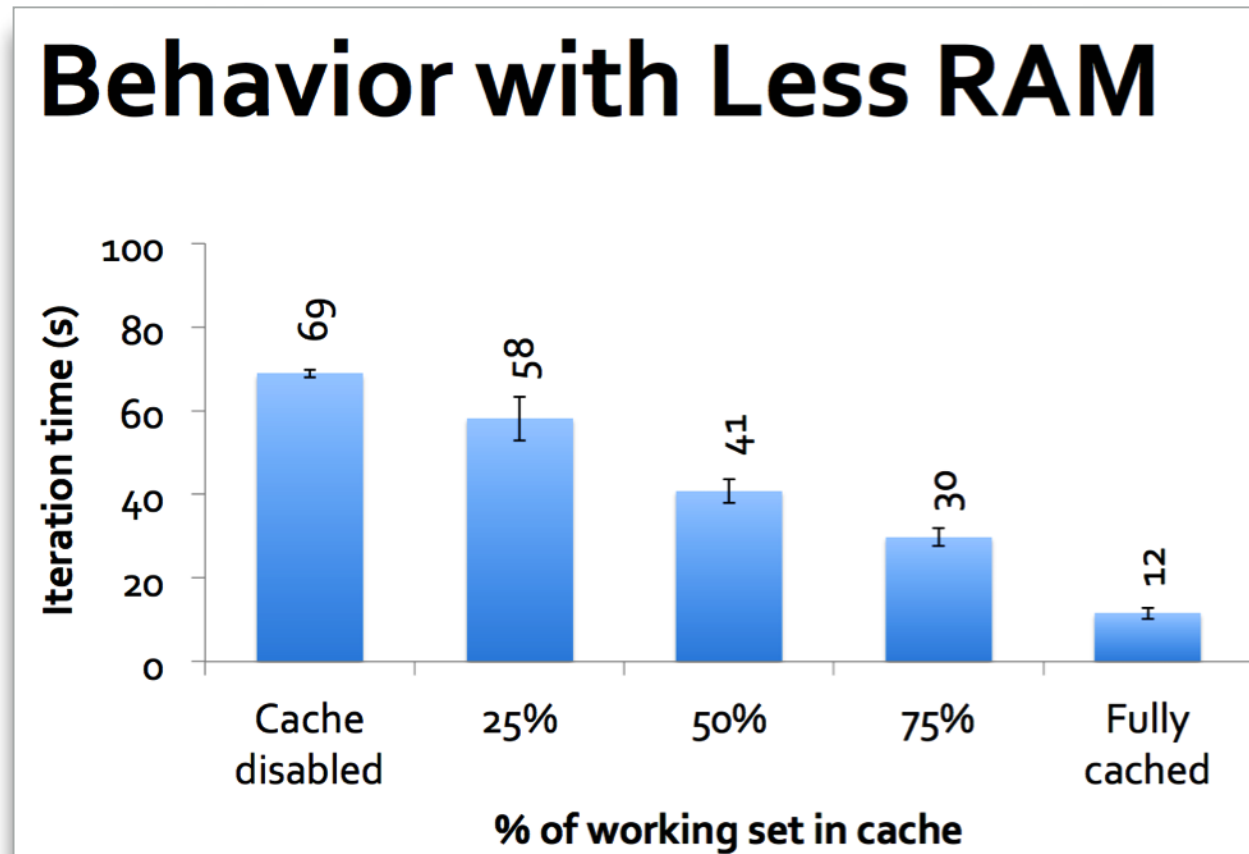
The State of Spark, and Where We're Going Next

Matei Zaharia

Spark Summit (2013)

youtu.be/nU6vO2EJAb4

A Brief History: *Spark*



The State of Spark, and Where We're Going Next

Matei Zaharia

Spark Summit (2013)

youtu.be/nU6vO2EJAb4