

Apache Spark for scalable data analytics

Raaz Sainudiin

May 03 2016, School of Maths & Stats, UC, Ilam

Where does big Data Come From?

- It's all happening online – could record every:

- » Click

- » Ad impression

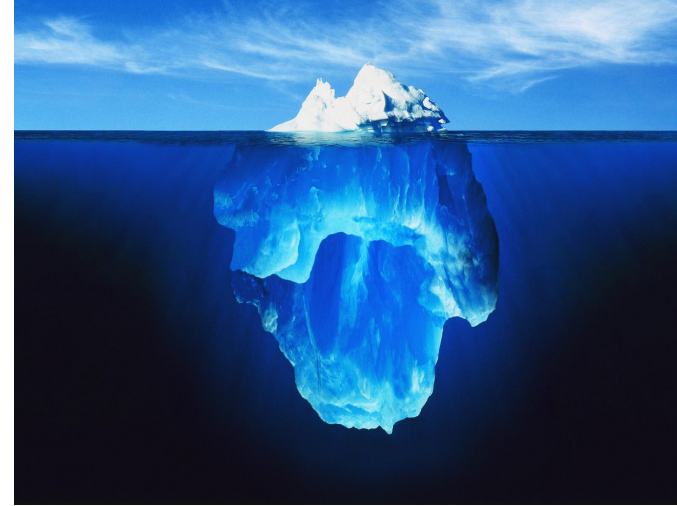
- » Billing event

- » Fast Forward, pause,...

- » Server request » GPS signal

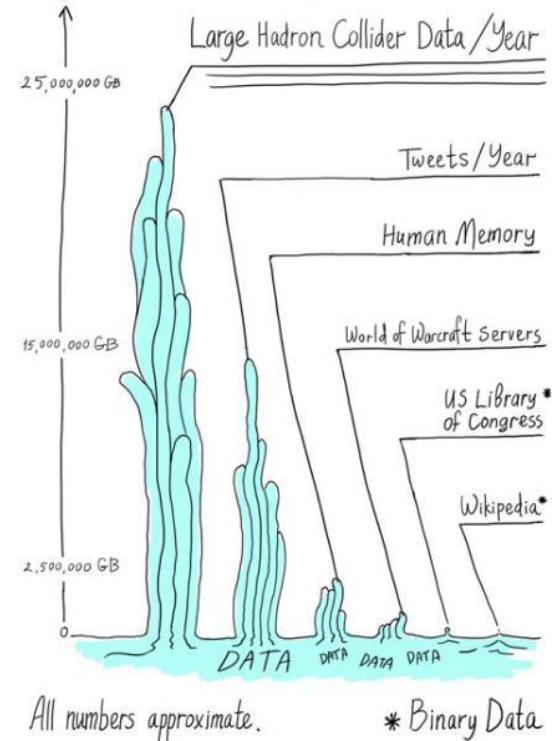
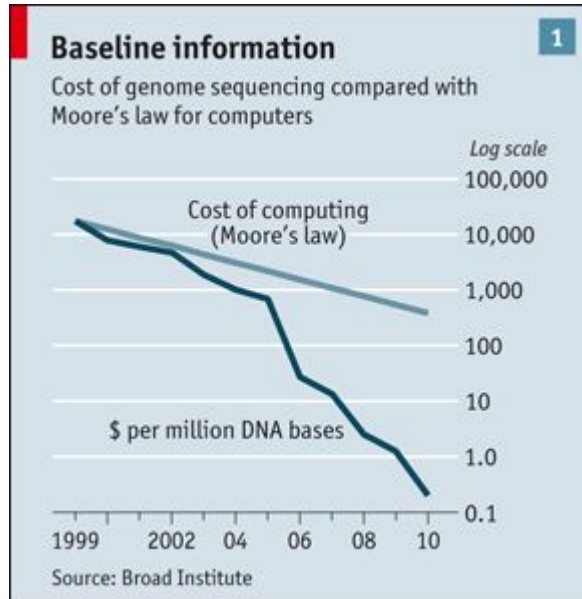
- » Transaction » Network message » Fault » ...

- » social media feed on twitter, yelp, facebook, instagram, youtube, ...



We can measure much faster than we can compute

Cost of DNA sequencing decays faster than that of computing, sensors are everywhere, ...



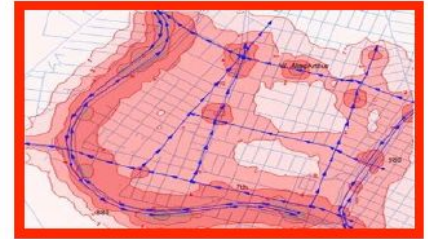
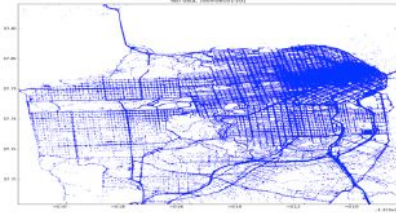
What can you do with Big Data?

Internet of Things: RFID tags

- California FasTrak Electronic Toll Collection transponder
 - Used to pay tolls
 - Collected data also used for traffic reporting
- » <http://www.511.org/>



What can you do with Big Data? App with real-time traffic maps, flows, ...

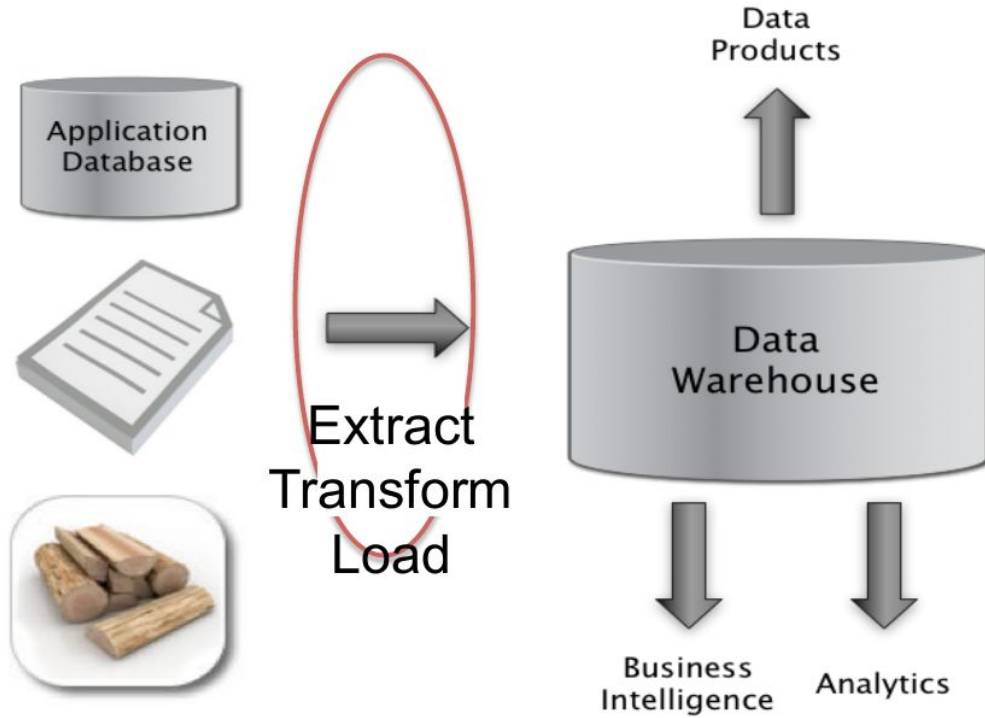


Crowdsourcing + Physical modeling + Sensing + Data Assimilation

=



The Big Picture



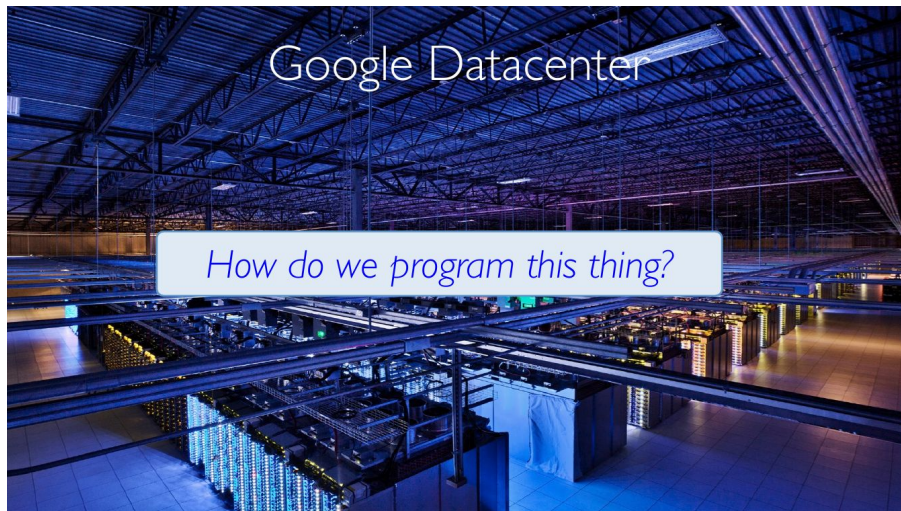
Traditional Analysis tools (sh, pandas, R) run on a single machine!!!

Big Data Examples

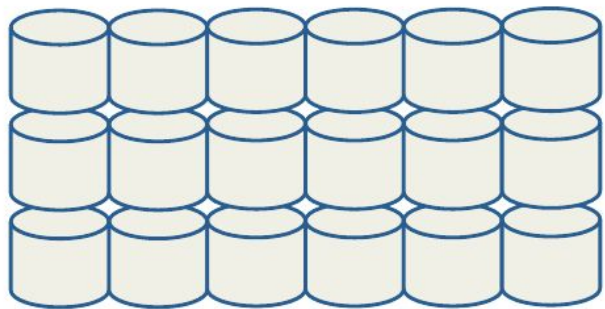
- Facebook's daily logs: **60 TB**
- 1,000 genomes project: **200 TB**
- Google web index: **10+ PB**
- Cost of 1 TB of disk: **~\$35**
- Time to read 1 TB from disk: **3 hours**
(100 MB/s)

The Big Data Problem

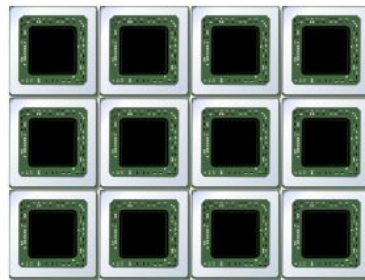
- A single machine can no longer process or even store all the data!
- Only solution is to **distribute** data over large clusters



Hardware for Big Data



Lots of hard drives



... and CPUs

Hardware for Big Data

Consumer-grade hardware
Not “gold plated”

Many desktop-like servers
Easy to add capacity
Cheaper per CPU/disk

Complexity in software



Image: Steve Jurvetson/Flickr

Problems with Cheap Hardware

Failures, Google's numbers:

1-5% hard drives/year

0.2% DIMMs/year

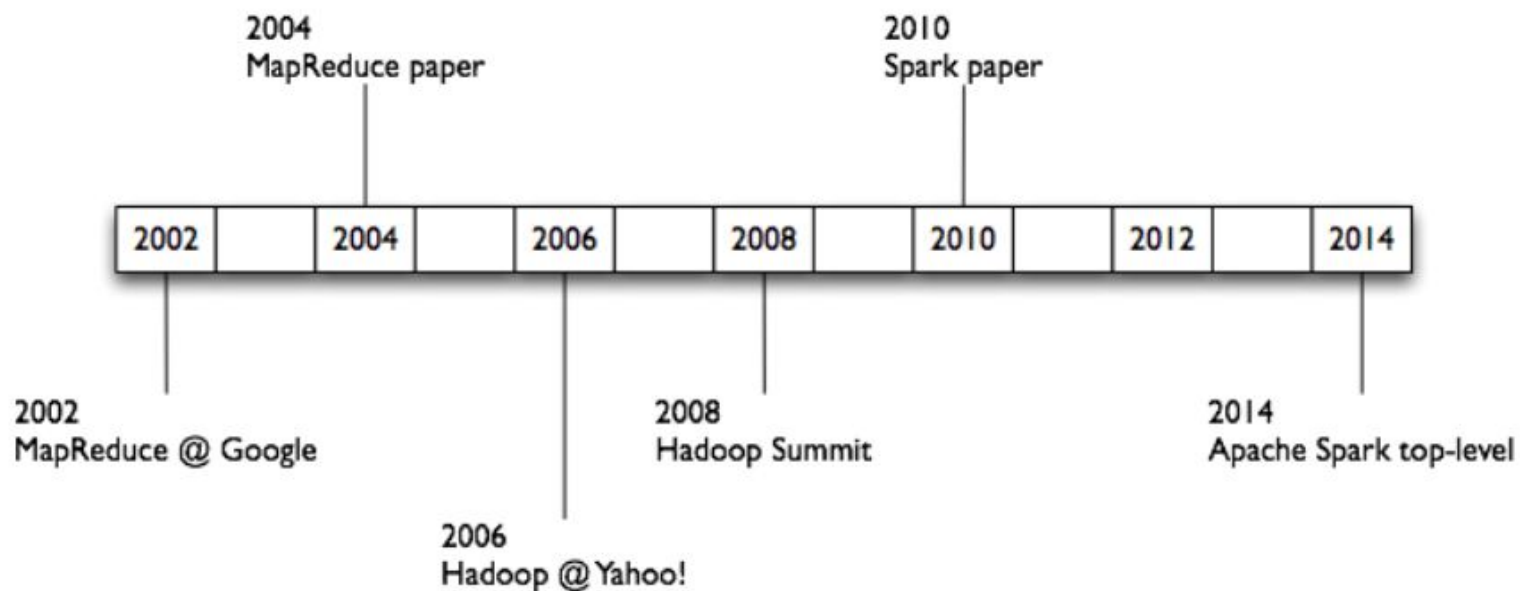
Network speeds versus shared memory

Much more latency

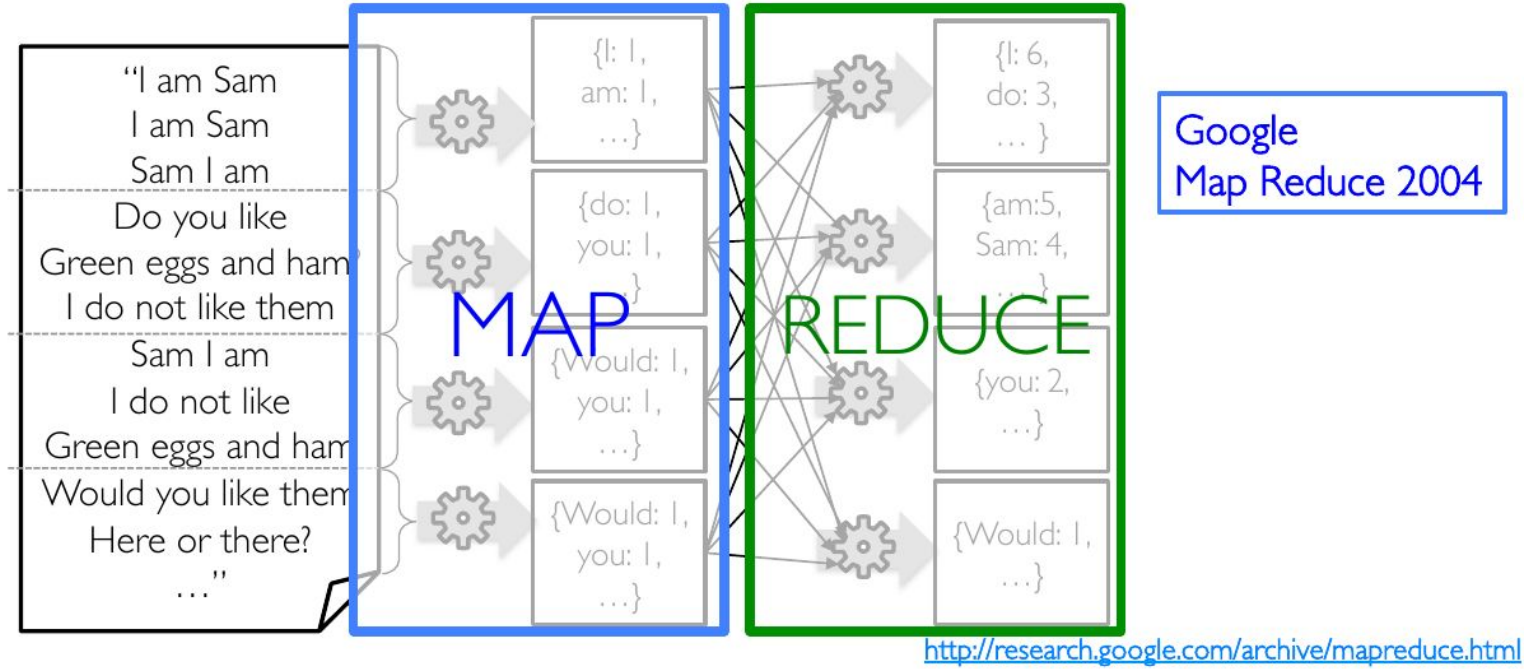
Network slower than storage

Uneven performance

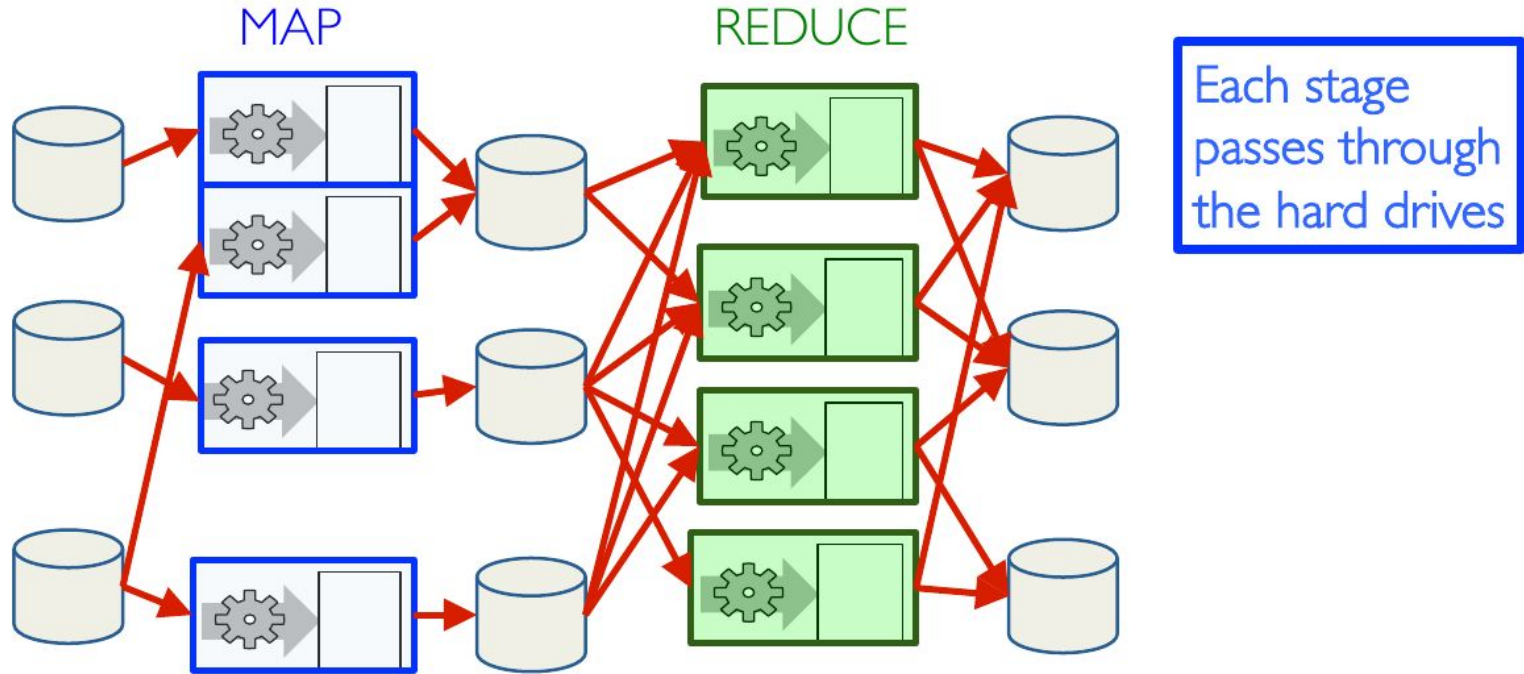
History Review



Word Count Example via Map & Reduce

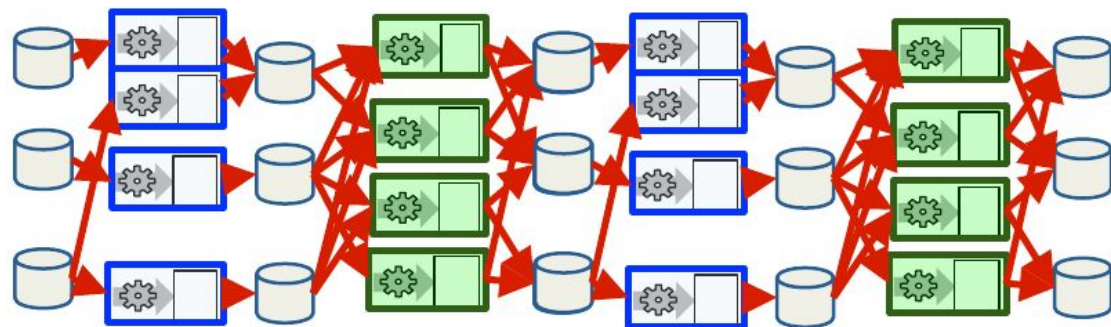


Map Reduce: Distributed Execution

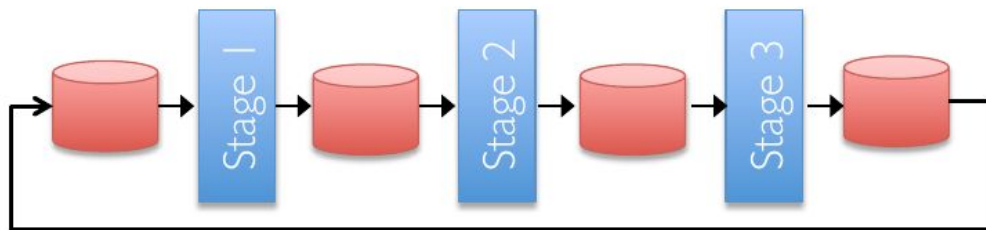


Map Reduce: Iterative Jobs

- Iterative jobs involve a lot of disk I/O for each repetition

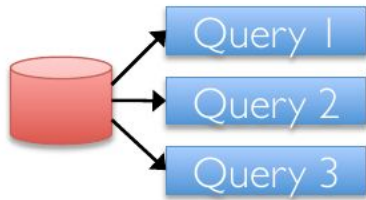


Disk I/O is
very slow!

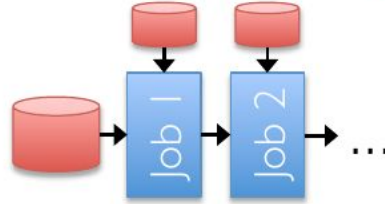


Apache Spark Motivation

- Using Map Reduce for complex jobs, interactive queries and online processing involves *lots of disk I/O*



Interactive mining

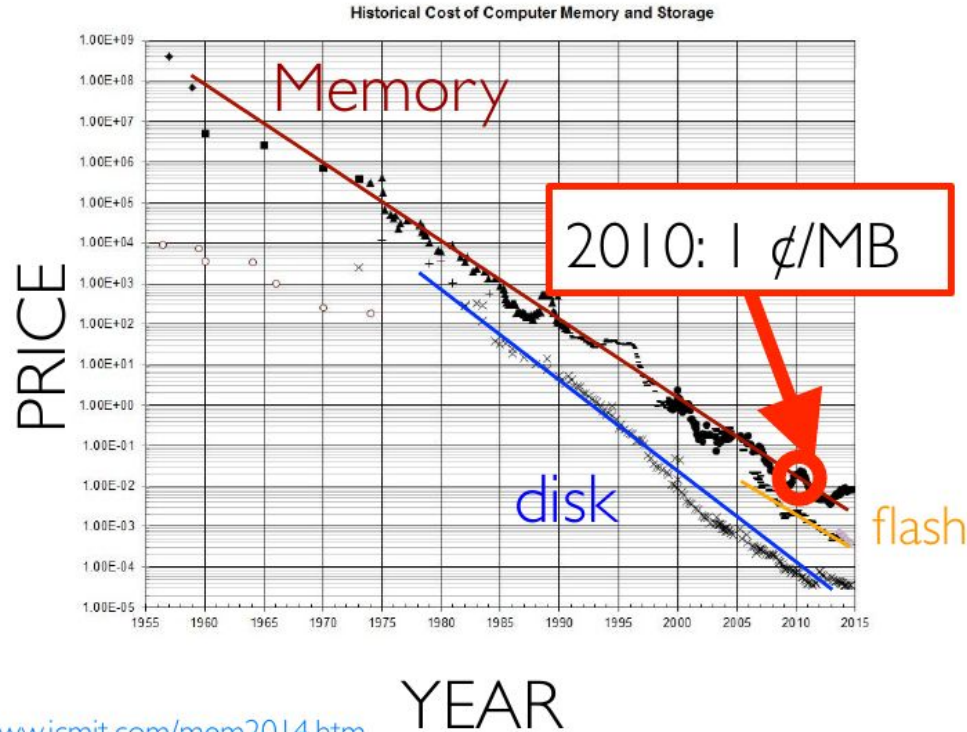


Stream processing

Also, iterative jobs

Disk I/O is very slow

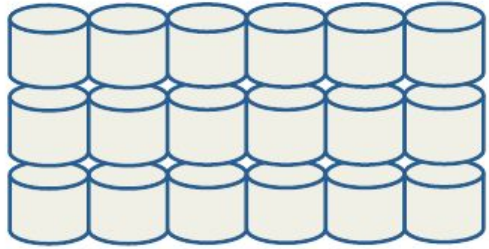
Tech Trend: Cost of Memory



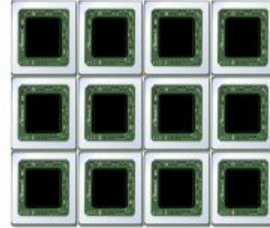
Lower cost means can
put more memory in
each server

<http://www.jcmit.com/mem2014.htm>

Hardware for Big Data



Lots of hard drives



... and CPUs



... and memory!

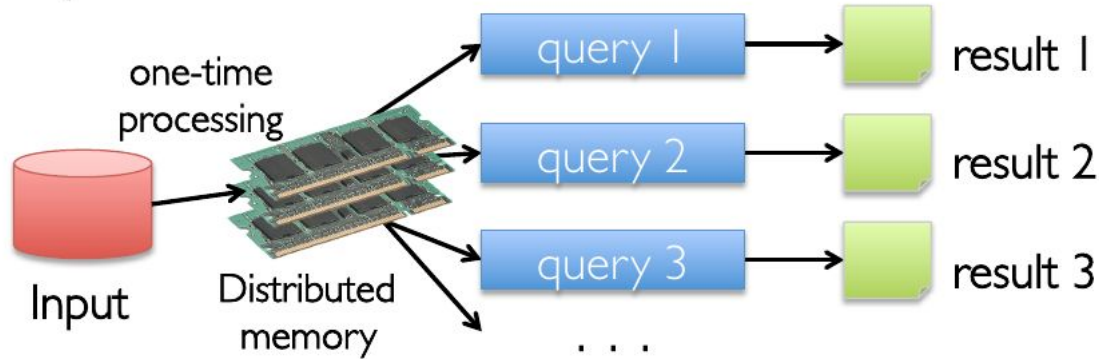
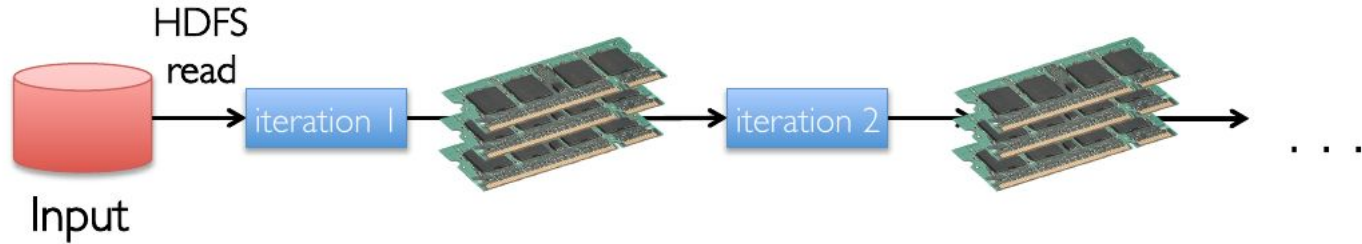
Opportunity

- Keep more data *in-memory*
- Create new distributed execution engine:



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

In-Memory Data Sharing

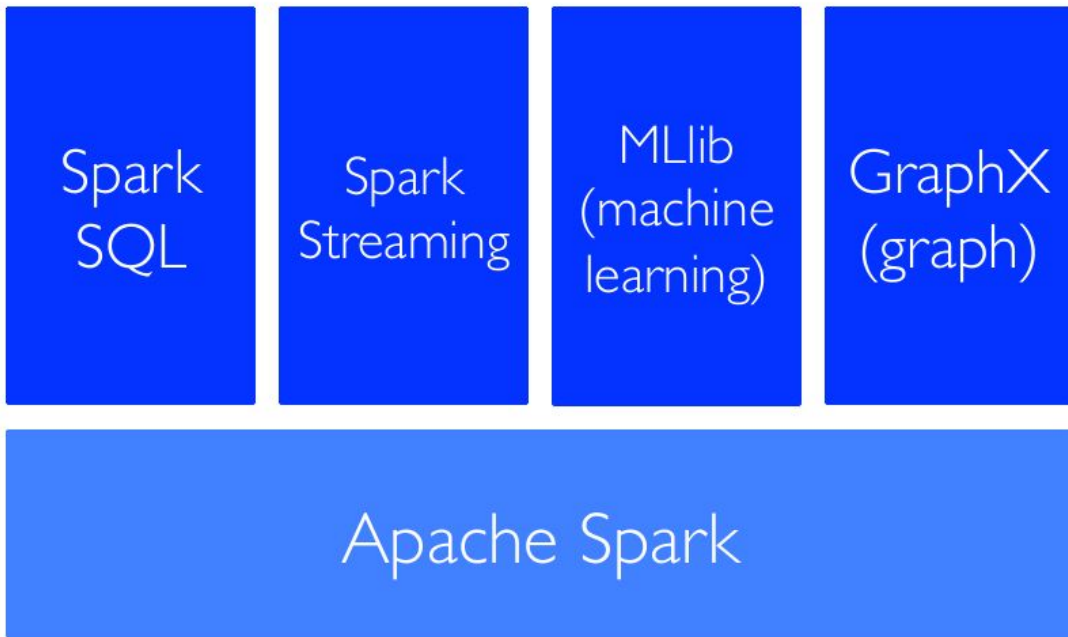


10-100x faster than network and disk

Resilient Distributed Datasets (RDDs)

- Write programs in terms of operations on distributed datasets
- Partitioned collections of objects spread across a cluster, stored in memory or on disk
- RDDs built and manipulated through a diverse set of parallel transformations (map, filter, join) and actions (count, collect, save)
- RDDs automatically rebuilt on machine failure

Spark Tools



One Spark to rule them all!

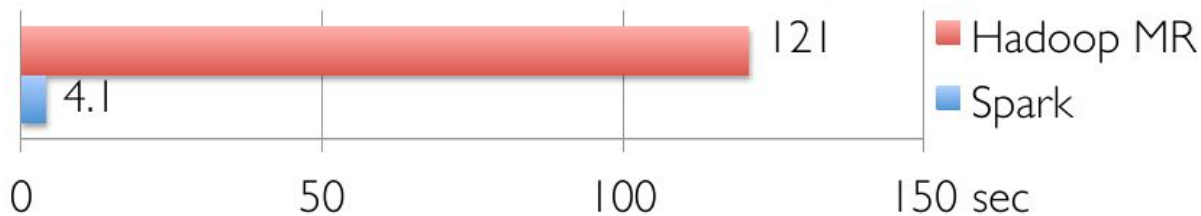
Spark and Map Reduce Differences

	Hadoop Map Reduce	Spark
Storage	Disk only	In-memory or on disk
Operations	Map and Reduce	Map, Reduce, Join, Sample, etc...
Execution model	Batch	Batch, interactive, streaming
Programming environments	Java	Scala, Java, R, and Python

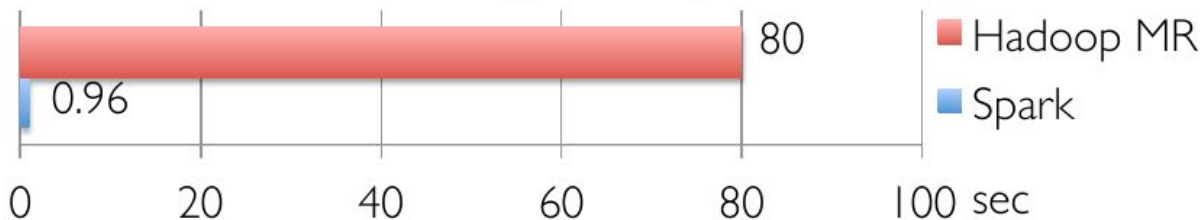
In-Memory Can Make a Big Difference

- Two iterative Machine Learning algorithms:

K-means Clustering



Logistic Regression



First Public Cloud Petabyte Sort

	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

[Daytona Gray 100TB](#)
sort benchmark record
(tied for 1st place)

<http://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html>

Join Christchurch-Apache-Spark-Meetup to learn Spark!

<https://itbrief.co.nz/article/university-canterbury-embraces-cloud-computing-and-big-data/>, <http://www.meetup.com/Christchurch-Apache-Spark-Meetup/>, <http://goo.gl/OIKRRL>



www.meetup.com/Christchurch-Apache-Spark-Meetup/

Find Start
a Meetup Group a Meetup Group

English Log in Sign up

Home Members Photos Pages Discussions More Join us!

Christchurch, New Zealand
Founded Feb 28, 2016

ChristchurchSparkers 76
Group reviews 4
Upcoming Meetups 5
Past Meetups 7
Our calendar

Organizer:
Raazesh Sainudiin

This is meetup in Christchurch, New Zealand for users of Spark (<http://spark.apache.org>), the high-speed cluster programming framework. We help one another learn and discuss various Spark-related projects, including Spark SQL, MLlib, GraphX and Spark Streaming, as well other aspects of the Berkeley Data Analytic Stack that are not part of the Spark sub-stack. The meetup includes introductions to the various Spark features, case studies from users, best practices for deployment and tuning, and updates on development.

Join us
Log in with Facebook to find out who you know here.

By creating a Meetup account, you agree to the Terms of Service

Christchurch Apache Spark Meetup: Welcome ChristchurchSparkers!

What's new

TAGS

Cloud Computing, Analytics, Big data, University of Canterbury

University of Canterbury embraces cloud computing and big data

MAY 02, 2016 4PM CATHERINE KNOWLES

The University of Canterbury has pulled off an Australasian first and is officially using cloud infrastructure and big data analytics to empower teachers and students.

Dr Raazesh Sainudiin, UC senior lecturer from the School of Mathematics and Statistics, secured grants from Databricks Academic Partners Program and Amazon Web Services Educate to give free and ongoing access to all UC faculty, staff and students and enable them to use their cloud-computing infrastructure for academic teaching and research.

Historical References

- circa 1979 – **Stanford, MIT, CMU, etc.:** set/list operations in LISP, Prolog, etc., for parallel processing
<http://www-formal.stanford.edu/jmc/history/lisp/lisp.htm>
- circa 2004 – **Google:** *MapReduce: Simplified Data Processing on Large Clusters*
Jeffrey Dean and Sanjay Ghemawat
<http://research.google.com/archive/mapreduce.html>
- circa 2006 – **Apache Hadoop**, originating from the Yahoo!'s Nutch Project
Doug Cutting
<http://research.yahoo.com/files/cutting.pdf>
- circa 2008 – **Yahoo!:** web scale search indexing
Hadoop Summit, HUG, etc.
<http://developer.yahoo.com/hadoop/>
- circa 2009 – **Amazon AWS:** Elastic MapReduce
Hadoop modified for EC2/S3, plus support for Hive, Pig, Cascading, etc.
<http://aws.amazon.com/elasticmapreduce/>

Spark Research Papers

- *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