Big Data

Mohammad Javad Dousti

A Famous Tweet

Big Data Borat
@BigDataBorat

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

6:17 AM · Feb 27, 2013 · Twitter Web Client

544 Retweets **24** Quote Tweets **394** Likes

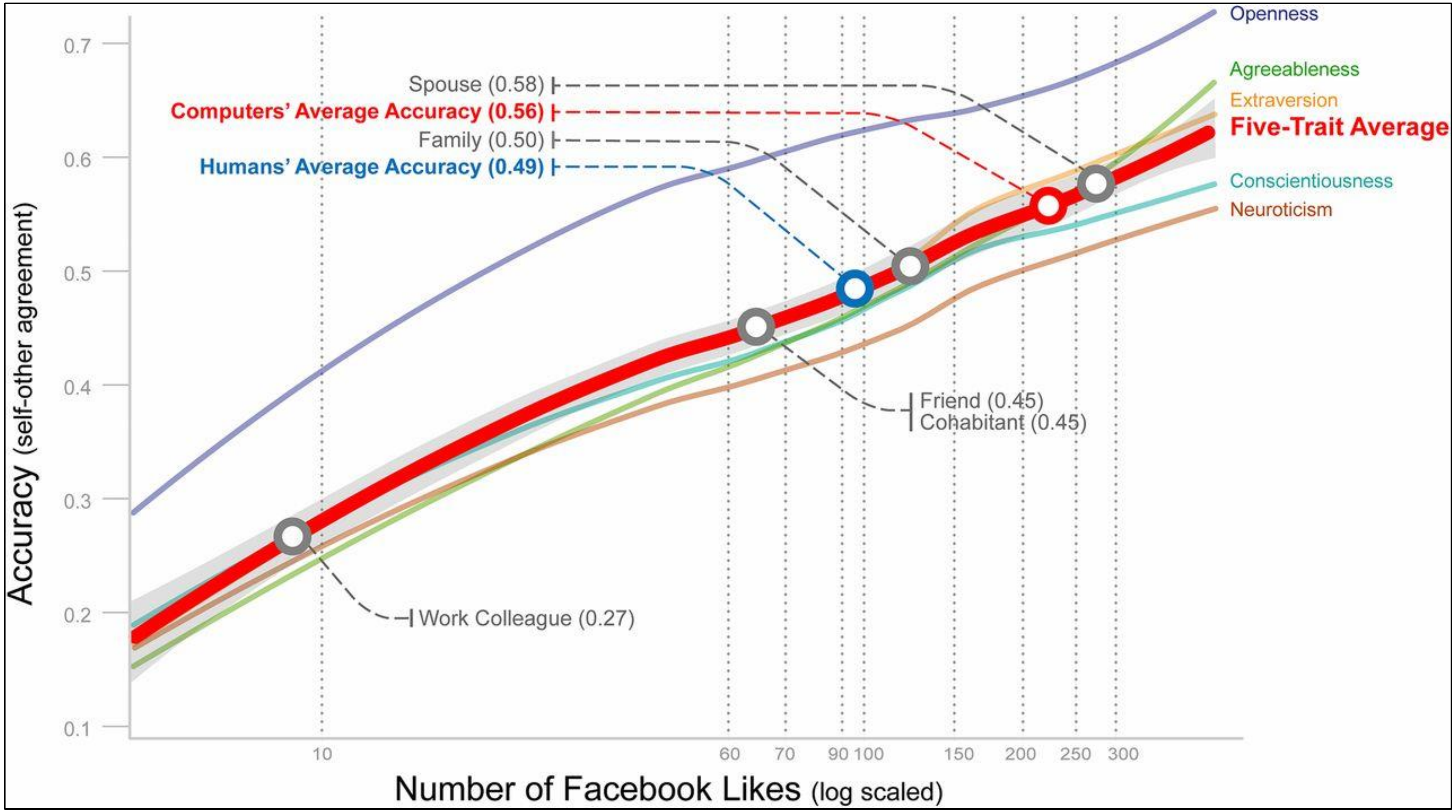
Experienced vs. Novice Machine Learning Engineer/Scientist

- I’ve conducted many machine learning system design interviews during my tenure at Facebook.
- Generally, an easy to spot difference between an experienced vs. a novice (yet knowledgeable) machine learning engineer/scientist boils down to understanding the followings:
  1. Data preparation (guideline creation, working w/ data annotators, data cleanup, etc.)
  2. Model deployment to production

What is big data?

- How much data is actually considered big?
  - 1 GB, 10GB, 100GB, 1TB, etc.?
- Big data means your memory is small!
- How to handle big data?
  - Sampling
  - Distributing
  - Streaming

Why is data important?

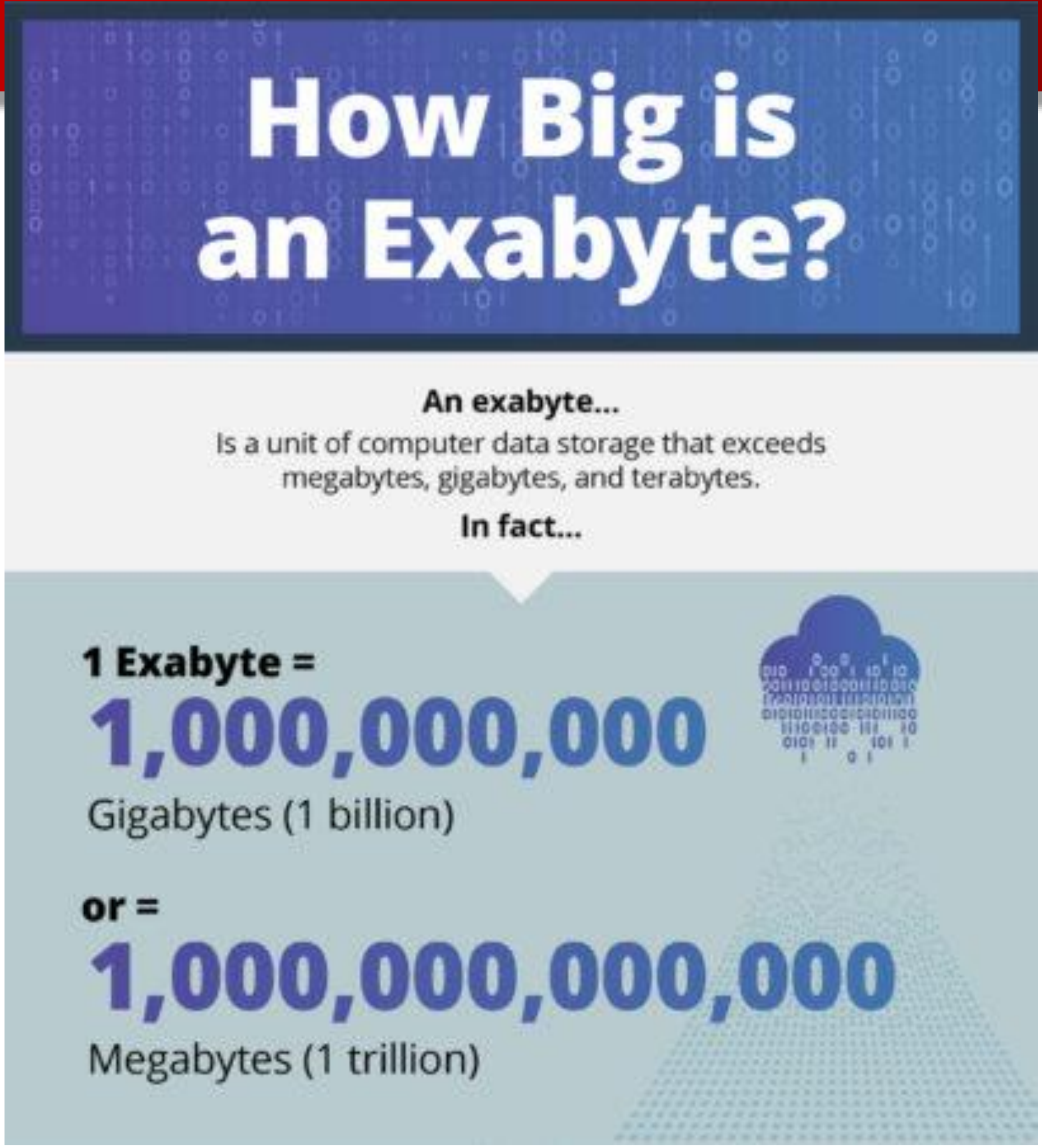


W. Youyou, M. Kosinski, and D. Stillwell. "Computer-based personality judgments are more accurate than those made by humans." *Proceedings of the National Academy of Sciences* 112.4 (2015): 1036-1040.

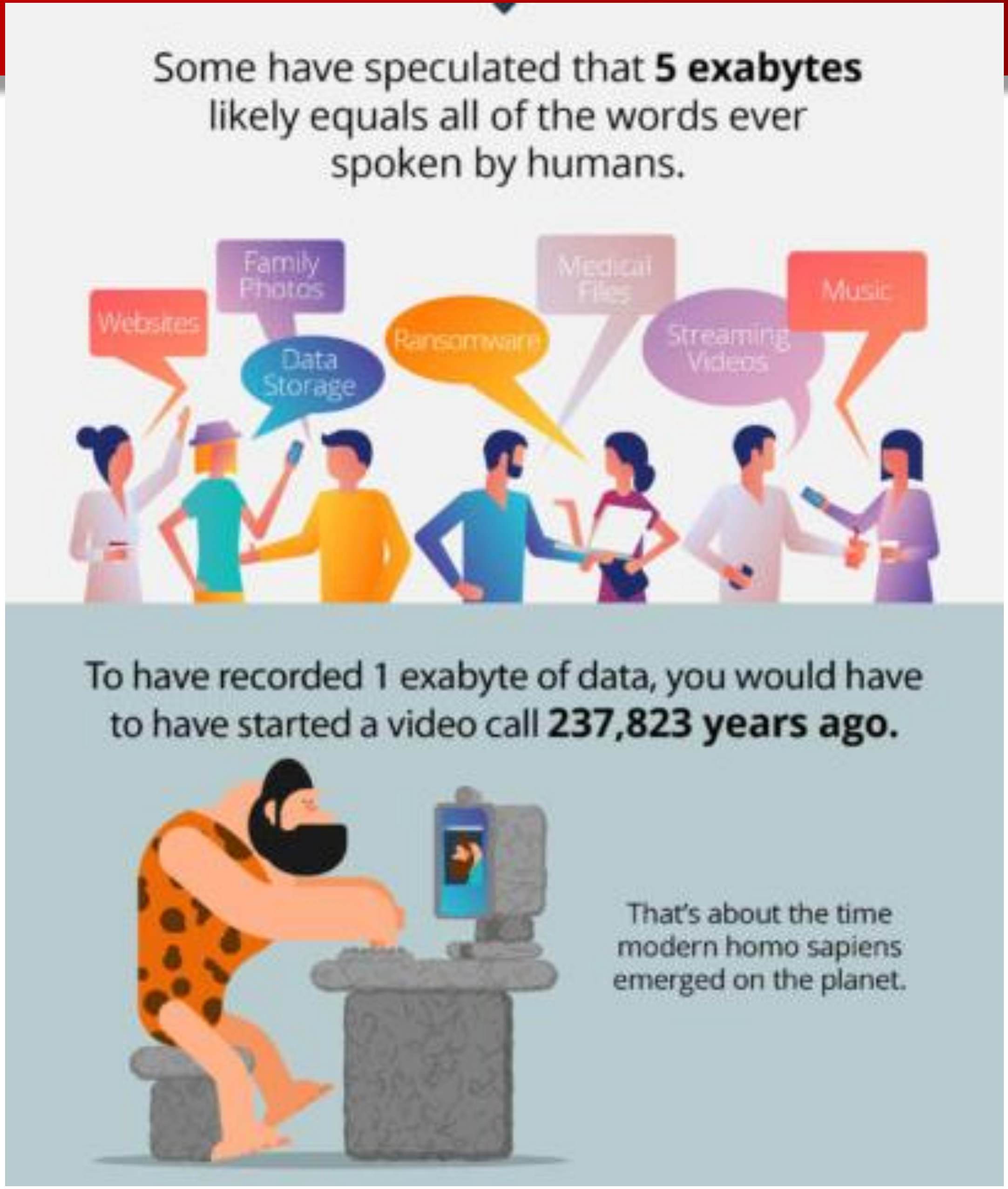
How much data does Facebook have?

- It contains extremely heterogeneous set of data:
  - Binary blobs (e.g., photos & videos)
  - Textual data (e.g., post contents)
  - Meta data (e.g., impressions & metadata)
- Facebook stores several exabytes of data* and the size grows exponentially.

* https://www.datanami.com/2020/02/19/storage-in-the-exabyte-era/
* https://www.datacenterknowledge.com/archives/2013/01/18/facebook-builds-new-data-centers-for-cold-storage


How Big is an Exabyte?
An exabyte...
Is a unit of computer data storage that exceeds megabytes, gigabytes, and terabytes.
In fact...
1 Exabyte =
1,000,000,000
Gigabytes (1 billion)
or =
1,000,000,000,000
Megabytes (1 trillion)


To put that into perspective...
Most smartphones come with about 64GB of built-in storage.
64G
That is .000000064 of an exabyte.
If one gigabyte is the size of Earth,
then an exabyte is the size of the sun.


Some have speculated that **5 exabytes** likely equals all of the words ever spoken by humans.
Family Photos
Medical Files
Websites
Music
Data Storage
Ransomware
Streaming Videos
To have recorded 1 exabyte of data, you would have to have started a video call **237,823 years ago.**
That's about the time modern homo sapiens emerged on the planet.

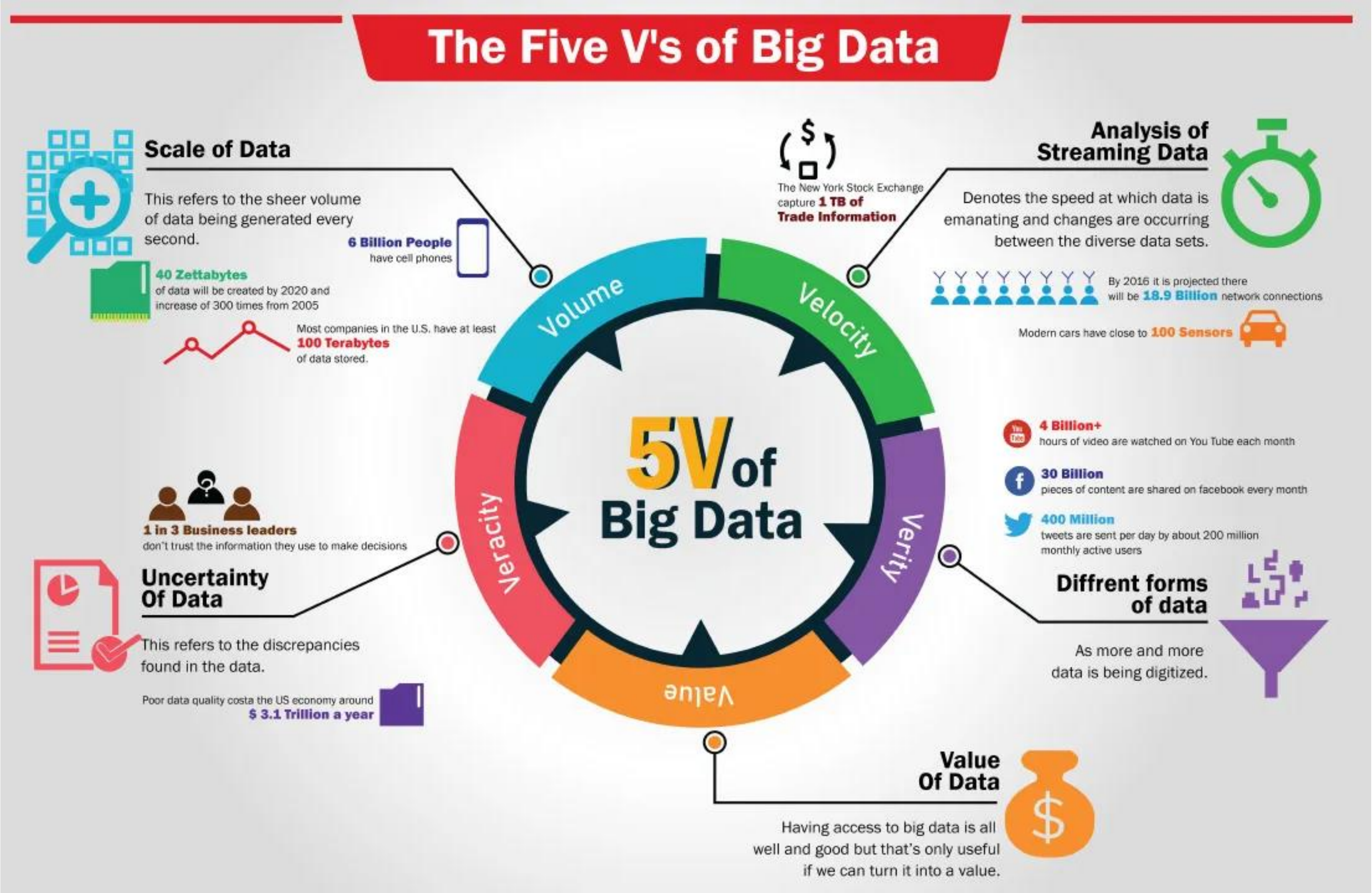

And since normal web browsing uses about **20 megabytes every hour,**
An office of 100 people would have to **search the web for 57,077 years** to reach an exabyte of data.

It’s all about a poster…

Where is our “Big” Data?

Big Data Characteristics

- 3 V's (Laney 2001)
  - Volume
  - Variety
  - Velocity
- Plus one
  - Value
- Another one
  - Veracity
- Plus many more
  - Validity
  - Variability
  - Viscosity & Volatility
  - Viability,
  - Venue,
  - Vocabulary

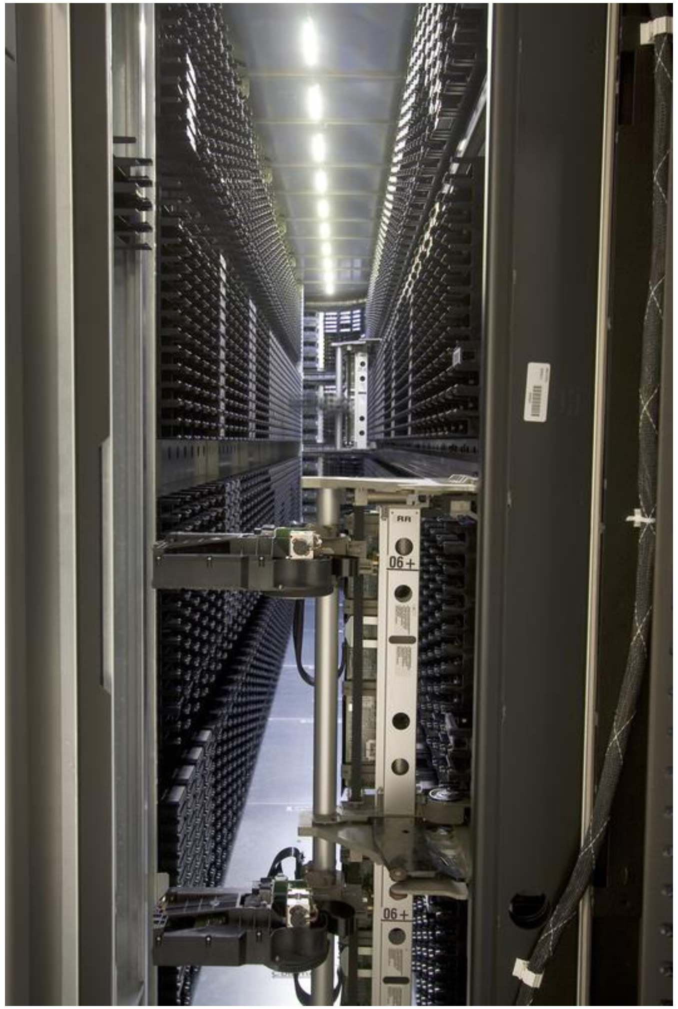


Source: https://morioh.com/p/ca19c6b8c0fe

Volume

- How much storage space the data takes up
  - Driven by exponential growth in storage capacity
  - Mediated by technology
    - Parallel processing
    - Better hardware
  - Zetabyte Era:
    - Cisco Inc. report:
      - The global IP traffic achieved an estimated 1.2 ZB (or an average of 96 exabytes (EB) per month) in 2016.
      - Global IP traffic: All digital data that passes over an IP network which includes, but is not limited to, the public Internet.
      - The largest contributing factor to the growth of IP traffic comes from video traffic (including online streaming services like **Netflix** and **YouTube**.)

Value		Metric
1000	kB	kilobyte
1000^2	MB	megabyte
1000^3	GB	gigabyte
1000^4	TB	terabyte
1000^5	PB	petabyte
1000^6	EB	exabyte
1000^7	ZB	zettabyte
1000^8	YB	yottabyte



Volume

- European Union industry chief Thierry Breton called on streaming platforms to help reduce their load on the continent's infrastructure at the beginning of COVID-19 lockdown.

THE VERGE TECH REVIEWS SCIENCE CREATORS ENTERTAINMENT VIDEO MORE

US & WORLD TECH CORONAVIRUS

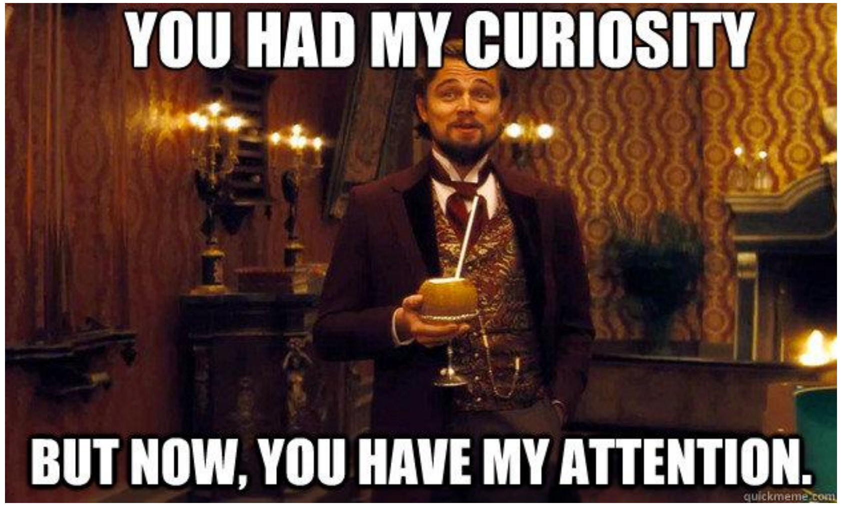
33

YouTube joins Netflix in reducing video quality in Europe

Streams now set to standard definition by default

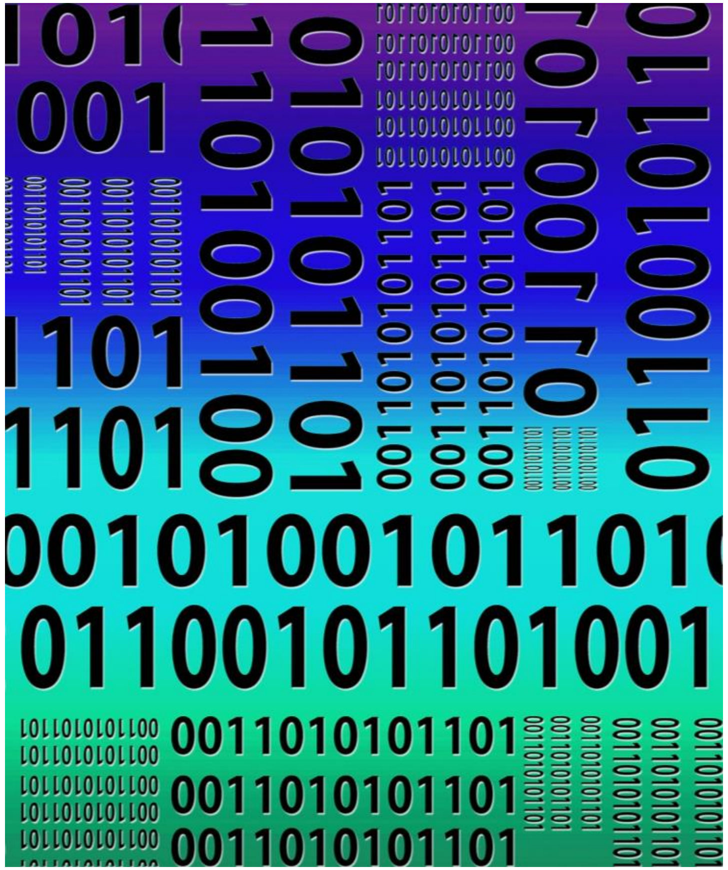
By Jon Porter | @JonPorty | Mar 20, 2020, 6:17am EDT

- ***Billion*** is the keyword we're looking for…



Variety

- How heterogeneous the data is.
  - Many features per item
  - Irregular structure (as opposed to structured data for RDBMSes)
  - Need to store and retrieve different data types quickly, efficiently, cheaply
  - Need to align & integrate different representations
  - Dealt with using standards, specs, etc.
- Big data draws from text, images, audio, and video
  - It completes missing pieces through data fusion.



Dimensions of Variety

- Content:
  - Image, spectrum, timeseries
- Form:
  - Text, numeric, relational, graphical, geospatial, sensory
- Format:
  - Plain-text file, .csv, fixed-width, Excel spreadsheet, HTML table
- Structure:
  - Unstructured text, semi-structured email, semantically-marked-up document
- Source:
  - Human-generated, automated sensor logging, scientific instruments, simulations
- Meaning:
  - “**This dish is hot.**”
- Representation:
  - Jan. 14, 2016 vs. 2016/01/14 vs. 2016/14/01
- etc.

Velocity

- How quickly data must be generated and processed
- Speed of storage / retrieval / analysis
- Aspects:
  - Real-time (acted on immediately)
  - Timeliness (rate of capture/usage)
  - Lifespan (how long it’s valuable)
  - Response time
- Strategies:
  - Simple ingest & access
  - Parallelization
  - Better hardware

Value

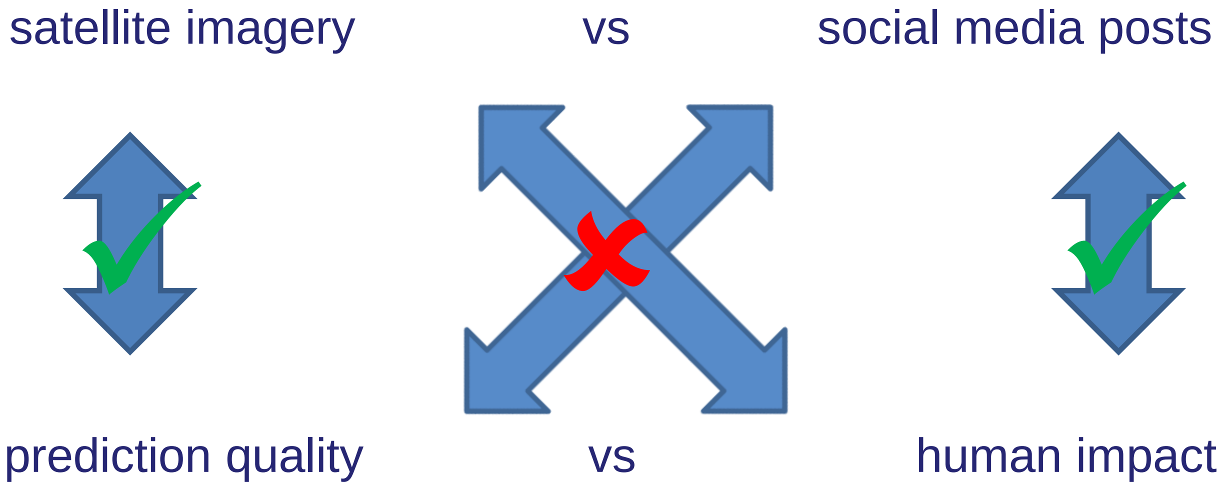
- “*Business value*” or ROI
- Data value can be achieved by the processing and analysis of large datasets.
- Value also can be measured by an assessment of the *other qualities of big data*.
- Value may also represent the profitability of information that is retrieved from the analysis of big data.

Veracity

- Is the data trustworthy?
  - Provenance, reliability, accuracy, completeness, ambiguity.
  - Importance of Veracity depends on what the *Value* of the data is.
- Strategies:
  - Transparent QC
  - Provenance tracking
  - Data management best practice
  - Good governance practices
- Note: Provenance and other veracity metadata can itself become Big Data.

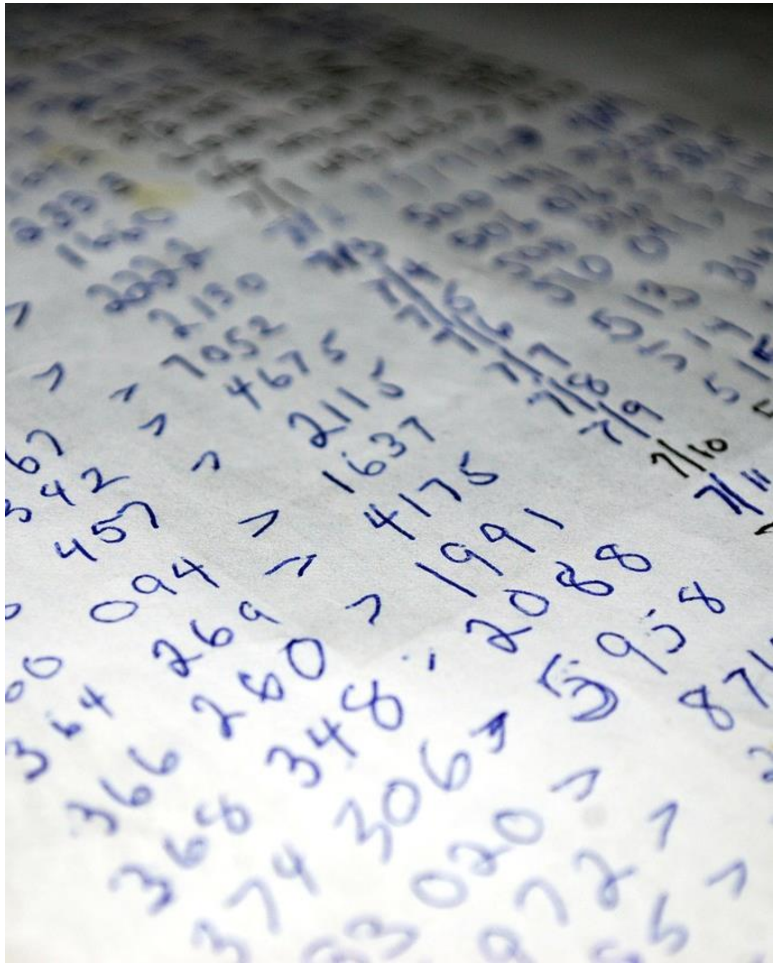
Validity

- Accuracy and correctness of the data relative to a particular use
  - Example: Gauging storm intensity



Variability

- How the *meaning of the data changes over time*
  - Language evolution
  - Data availability
  - Sampling processes
  - Changes in characteristics of the data source



Viscosity & Volatility

- Both related to velocity
- Viscosity: *data velocity relative to timescale of event being studied*
- Volatility: *rate of data loss and stable lifetime of data*
  - Scientific data often has practically unlimited lifespan, but social / business data may evaporate in finite time.

More V’s

- Viabililty
  - Which data has meaningful relations to questions of interest?
  - Another take on value.
- Venue
  - Where does the data live and how do you get it?
- Vocabulary
  - Metadata describing structure, content, & provenance
  - Schemas, semantics, ontologies, taxonomies, vocabularies

Critiques of Big V's Model

- Big V's model concerns mostly about scalability than understandability.
- An alternative is *cognitive big data* which concerns around:
  - **Data completeness:** Understanding of the non-obvious from data.
  - **Data correlation, causation, and predictability:** Causality as not essential requirement to achieve predictability.
  - **Explainability and interpretability:** Humans desire to understand and accept what they understand, where algorithms do not cope with this.
  - **Level of automated decision making:** Algorithms that support automated decision making and algorithmic self-learning.

Source: A. Lugmayr, et al. *A comprehensive survey on big-data research and its implications-What is really 'new' in big data? It's cognitive big data!. Pacific Asia Conference on Information Systems,* 2016.

Meaningfulness of Analytic Answers

- **A risk with "Data mining" is that an analyst can "discover" patterns that are meaningless**
- Statisticians call it ***Bonferroni's principle***:
  - Roughly, if you look in more places for interesting patterns than your amount of data will support, you are bound to find crap.

Apache Spark

Slides are taken from various sources. See references slide.

Mandatory reading: **Matei Zaharia et al. "Spark: Cluster computing with working sets," *HotCloud,* 2010.**

Motivation

- Moore's law:
  - The number of transistors in a dense integrated circuit (IC) doubles about every two years.
- Kryder's Law:
  - *Inside of a decade and a half, hard disks had increased their capacity 1,000-fold, a rate that Intel founder Gordon Moore himself has called flabbergasting.*
  - This is much faster than the two-year doubling time of semiconductor chip density suggested by Moore's law!
- Unfortunately, disk speeds don't increase at the rate of capacity.

Gordon Moore, ex-Intel CEO

Mark Kryder, Seagate SVP

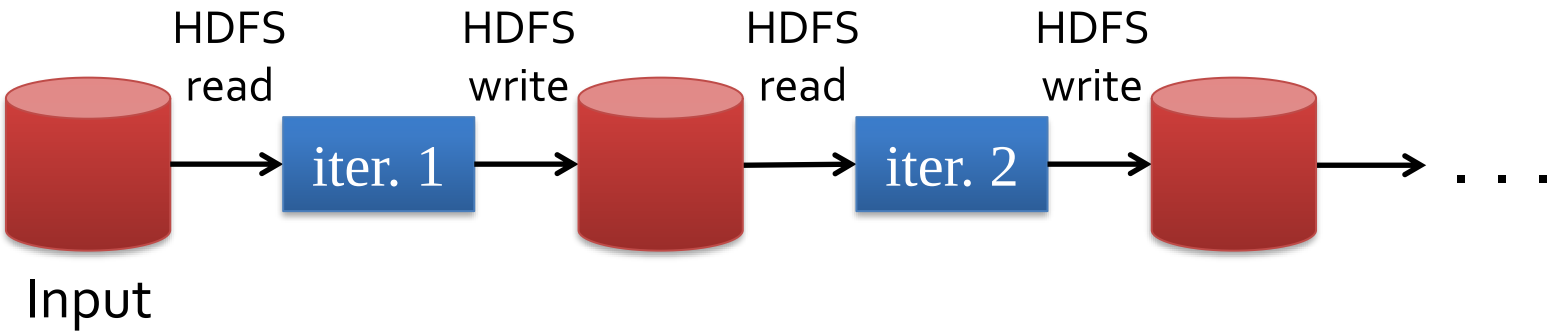
Why use multiple disks?

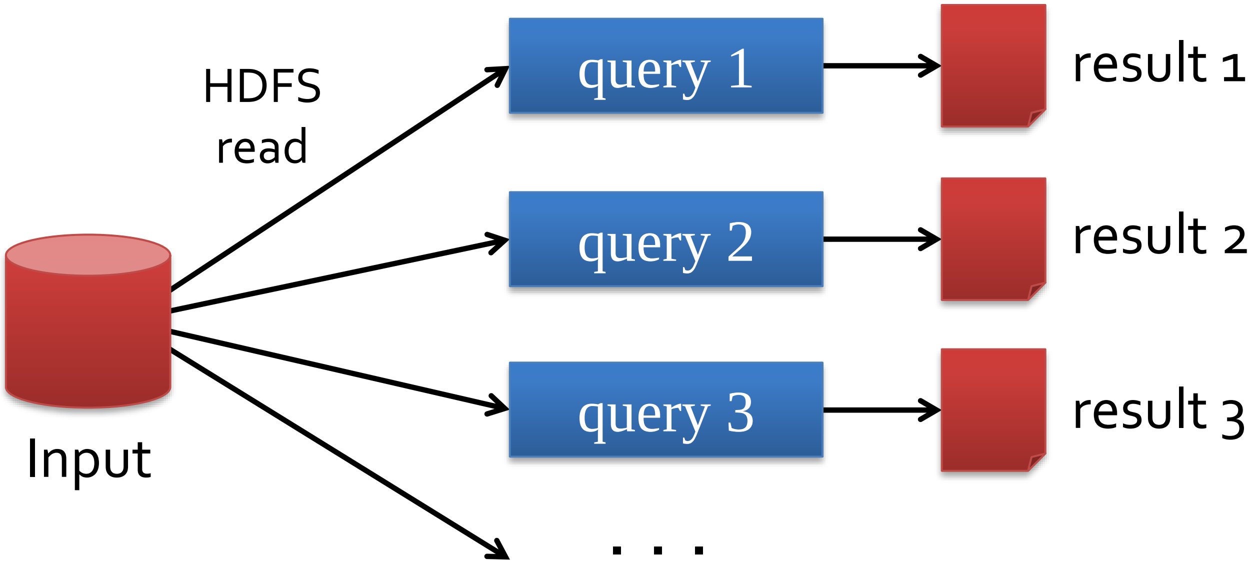
- Capacity
  - More disks allows us to store more data.
- Performance
  - Access multiple disks in parallel.
  - Each disk can be working on independent read or write.
  - Overlap seek and rotational positioning time for all.
- Reliability
  - Recover from disk (or single sector) failures.
  - Will need to store multiple copies of data to recover.
- So, what is the simplest arrangement?

Limitations of MapReduce

- MapReduce is great at one-pass computation, but inefficient for multi-pass algorithms.
- No efficient primitives for data sharing.
  - State between steps goes to distributed file system.
  - Slow due to replication & disk storage.

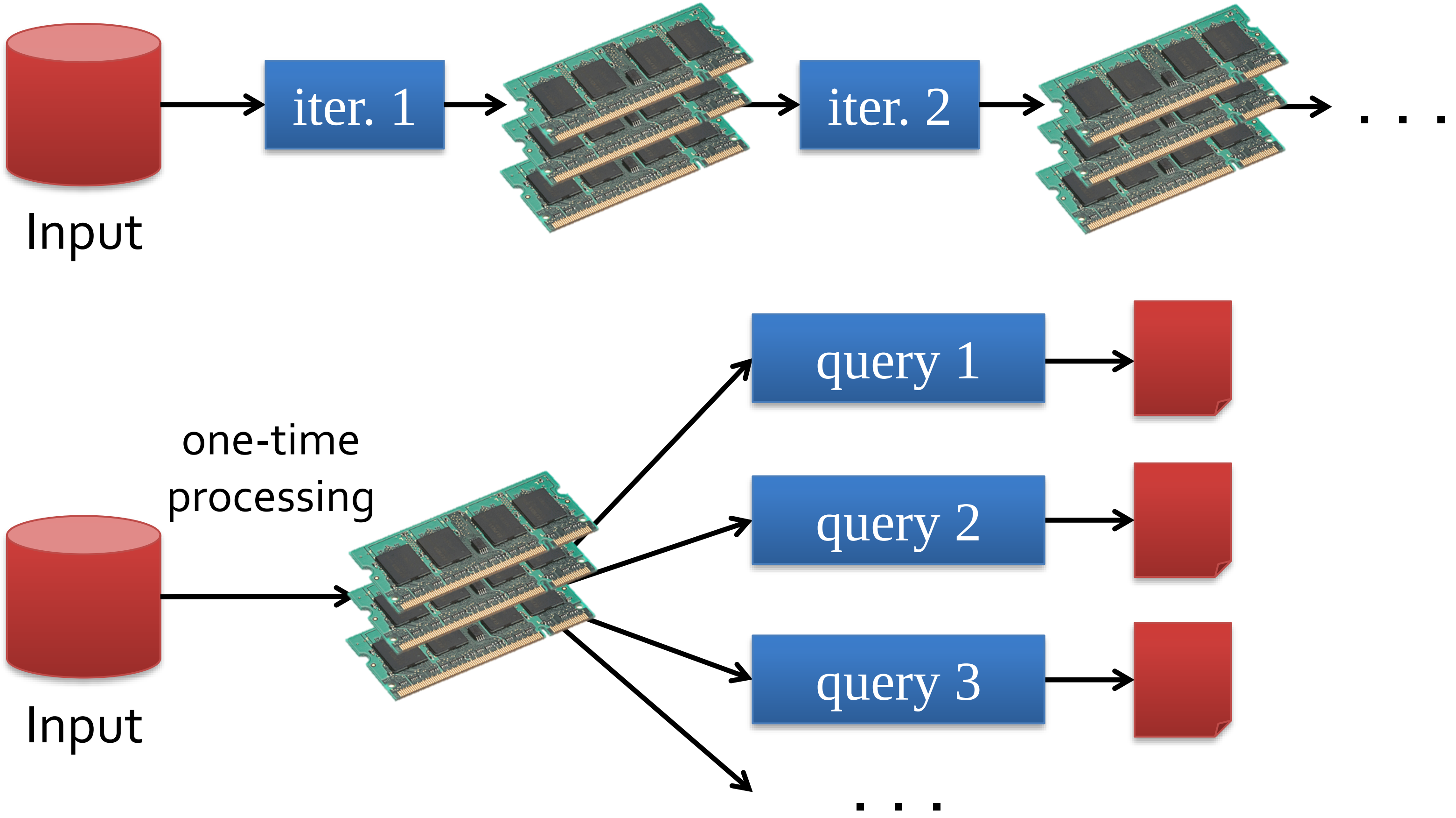
Example: Iterative Apps





Slow due to replication and disk I/O,
but necessary for fault tolerance

Example: In-Memory Data Sharing



design a distributed memory abstraction that is both **fault-tolerant** and **efficient**

Spark Brief History

Matei Zaharia

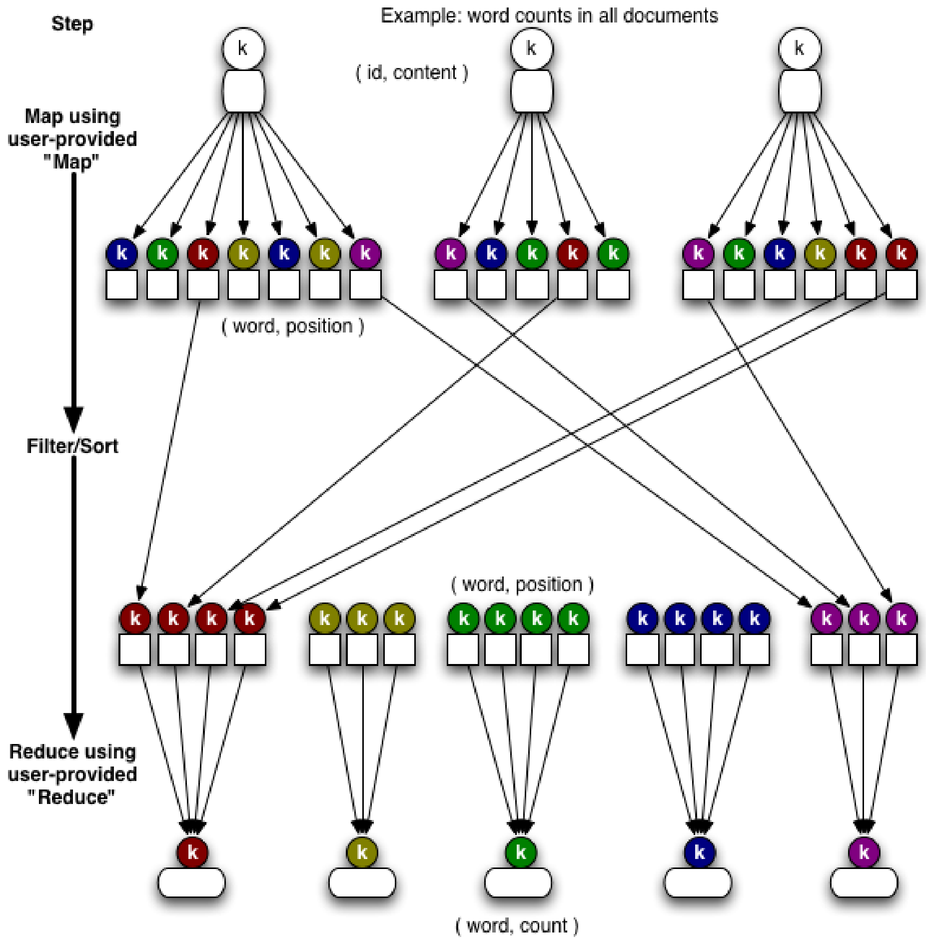
- AMPLab UC Berkley
  - Project Lead: Matei Zaharia (professor at MIT and then Stanford)
- First paper published on RDD’s was in 2012 (Spark paper was published in 2010)
- Open sourced from day one, growing number of contributors
- Supports Java, Scala and Python ☺
- Released its 1.0 version in May 2014. Currently in 3.5.3.
- Databricks company established to support Spark and all its related technologies.
  - Matei currently sits as its CTO
- Current users: Amazon, Alibaba, Baidu, eBay, Facebook, Groupon, Ooyala, OpenTable, Box, Shopify, TechBase, Yahoo!, and so on.

What is Spark?

- Data-flow engine to support data analysis in clusters

Computation model that views data moving from computation unit to computation unit.

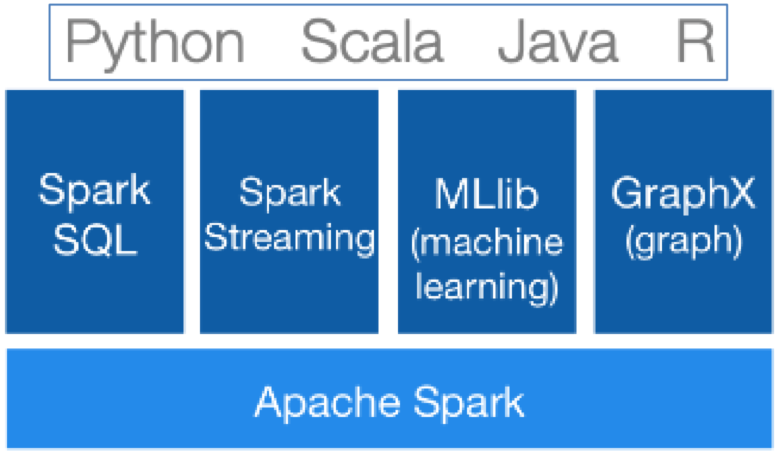
E.g., MapReduce



What is Spark?

- Data-flow engine to support data analysis in clusters
- Numerous libraries
  - Machine learning (MLlib)
  - Graph processing
  - Time-series
  - SQL
  - …
- Many parallel primitives
  - Map, filter, reduce, group by, join, …

Not a database!
Underlying data is considered mostly static.



What is Spark?

- Data-flow engine to support data analysis in clusters
- Generally built on top of Hadoop File System (HDFS)
  - Write-once read-many
  - Large file – distributed
  - Fault-tolerant

Primarily for large-scale computing.

Hides implementation details.

Key Data Abstraction: Resilient Distributed Datasets *(RDDs)*

- Immutable collection spread across cluster
- Statically typed: RDD[T] has objects of type T
- *Transformations* build RDDs from other RDDs – map, filter, …
  - Lazily built in parallel
  - Automatically rebuilt on failure
- *Actions* do things with RDDs – aggregate, save, …
- Controllable persistence – e.g., caching in RAM

```
val sc = new SparkContext()
val lines = sc.textFile("log.txt")   // RDD[String]

// Transform using standard collection operations
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split('\t')(2))

messages.saveAsTextFile("errors.txt")
```

→ lazily evaluated

→ kicks off a computation

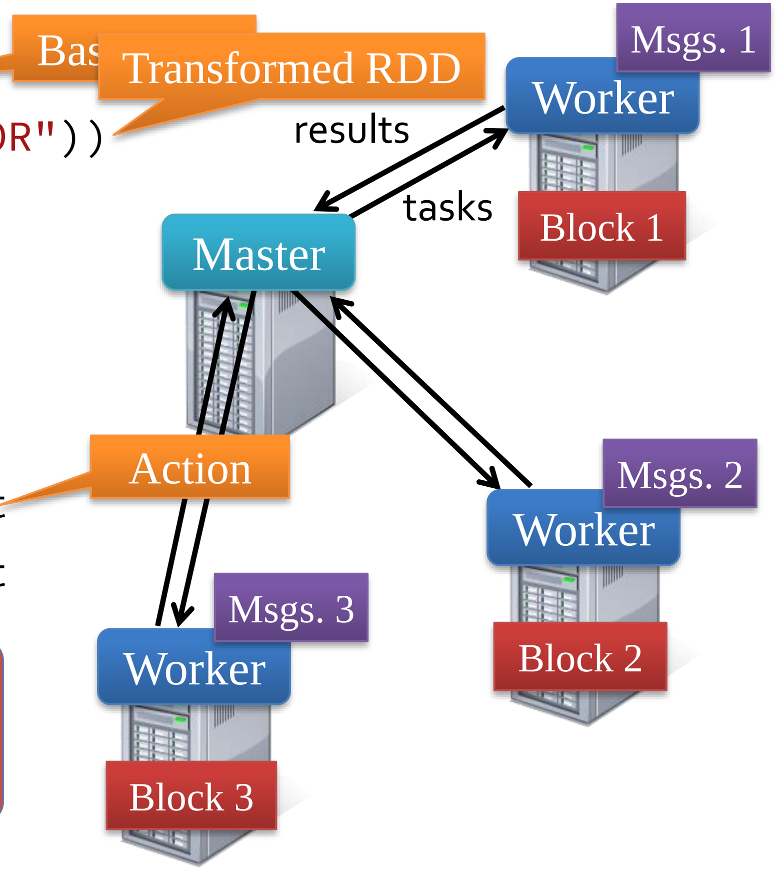
Example: Log Mining

- Load error messages from a log into memory, then interactively search for various patterns

```
lines = spark.textFile("hdfs://...")
errors = lines.filter(_.startsWith("ERROR"))
messages = errors.map(_.split('\t')(2))
messages.persist()
```

hdfs → lines → errors → message

```
messages.filter(_.contains("foo")).count
messages.filter(_.contains("bar")).count
```

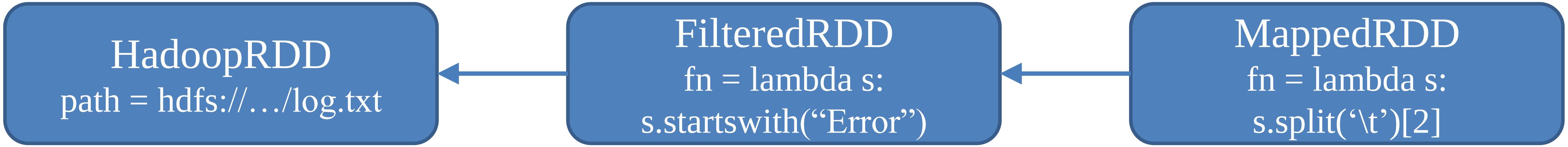


Result: scaled to 1 TB data in 5-7 sec
(vs 170 sec for on-disk data)

RDD fault-tolerance

- *Lineage:* Each RDD knows transformation used to (re)compute it.
  - By default, only store the lineage, not the data.

```
messages = sc.textFile(“hdfs://…/log.txt”)
                       .filter(lambda entry: entry.startswith(“Error”))
                       .map(lambda entry: entry.split(‘\t’)[2])
```

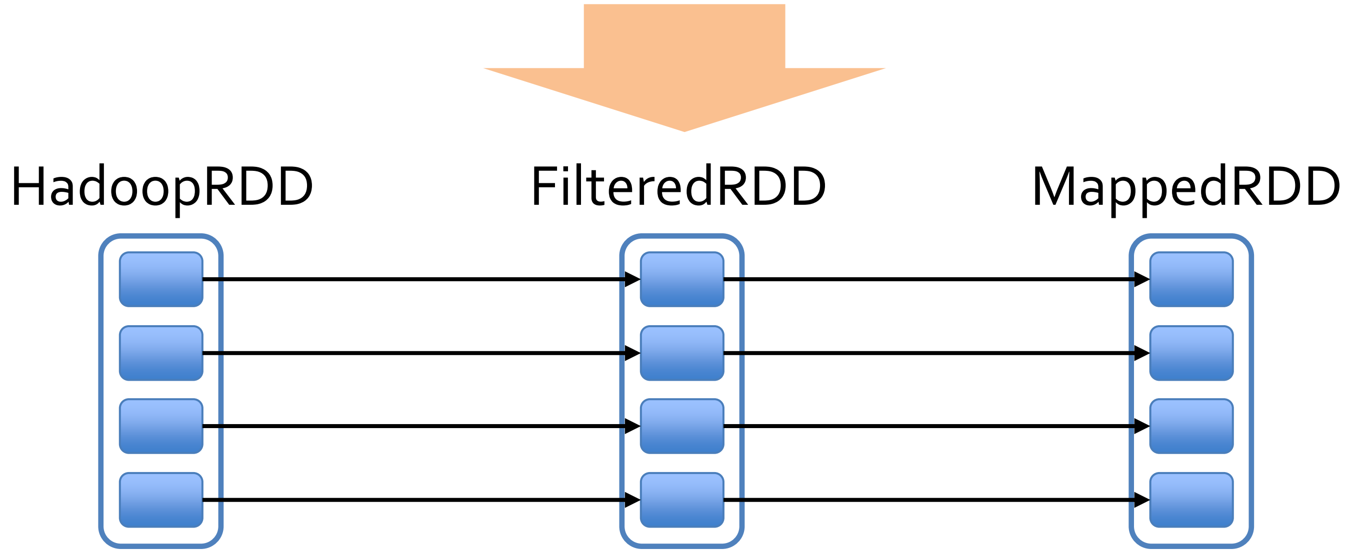


Fault Recovery

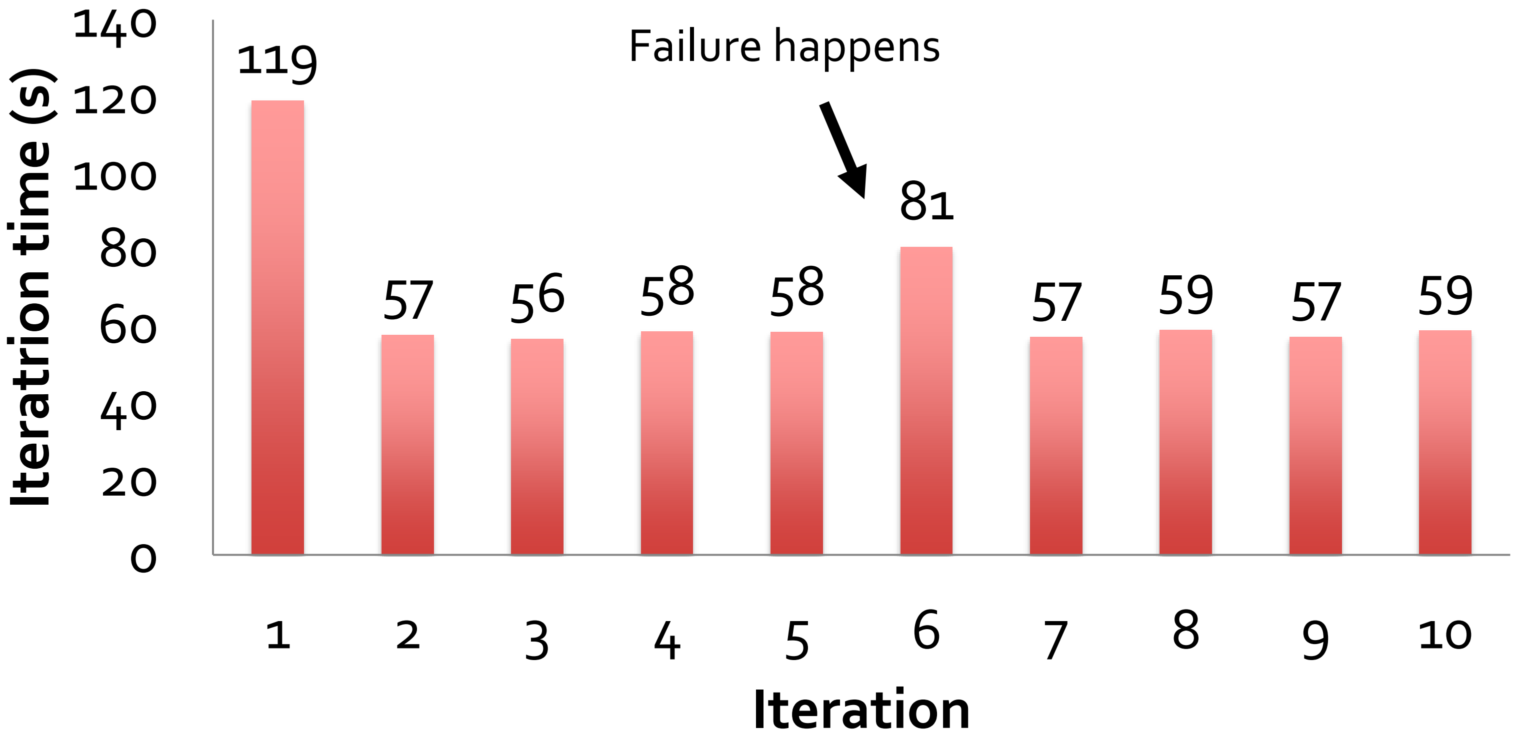
- RDDs track the graph of transformations that built them (their *lineage*) to rebuild lost data

- Example:

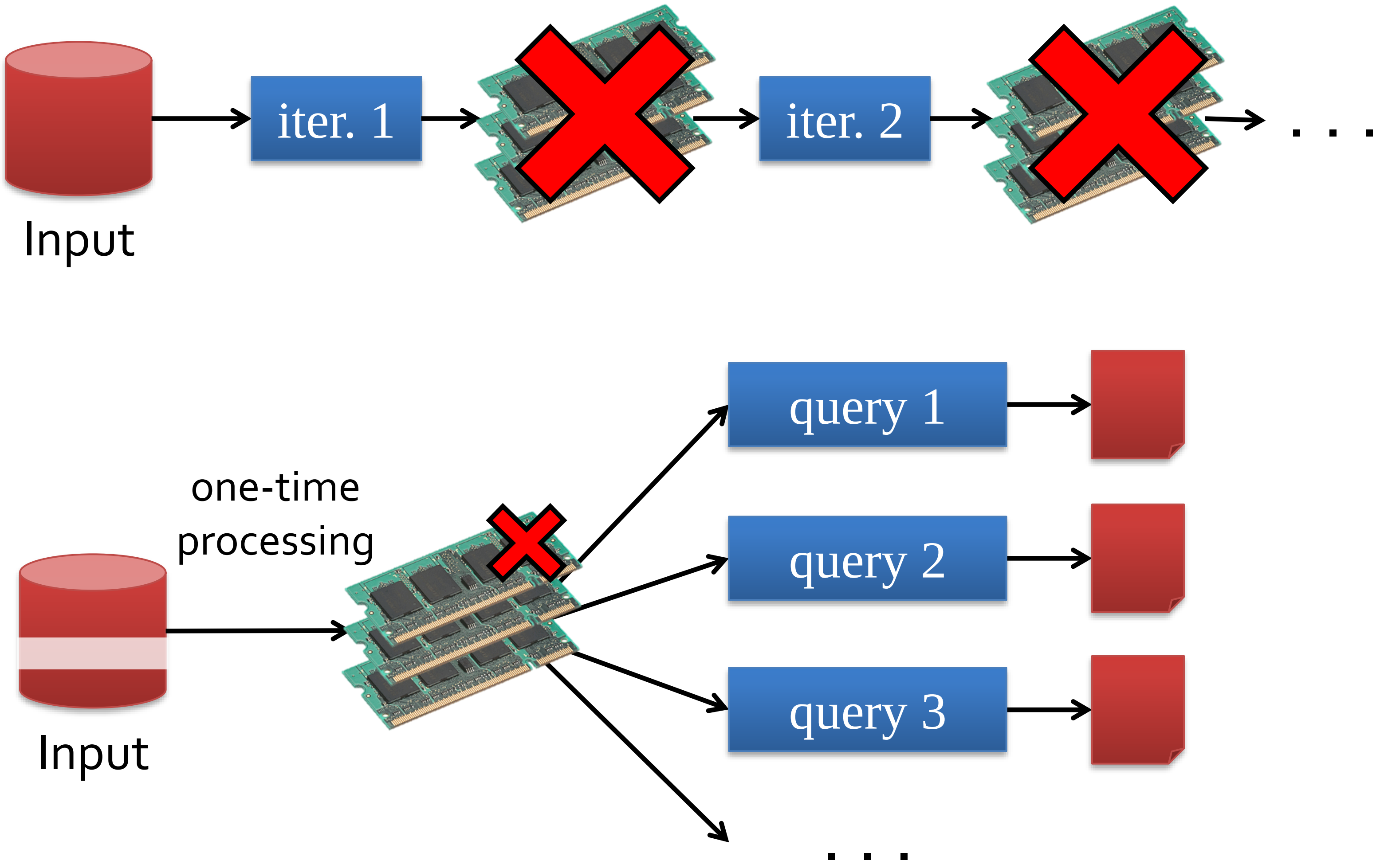
```
messages = textFile(...).filter(_.contains("error"))
                        .map(_.split('\t')(2))
```



Fault Recovery Results



RDD Recovery



Spark Examples

MapReduce

```
result = data.flatMap(map_fn)
             .groupByKey()
             .map(lambda (k,vs): reduce_fn(k,vs))
```

```
result = data.flatMap(map_fn)
             .reduceByKey(combiner_fn)
             .map(lambda (k,vs): reduce_fn(k,vs))
```

Word count

```
counts = sc.textfile("hdfs://...")
             .flatMap(lambda line: line.split('\s'))
             .map(lambda word: (word, 1))
             .reduceByKey(operator.add)
counts.save("hdfs://...")
```

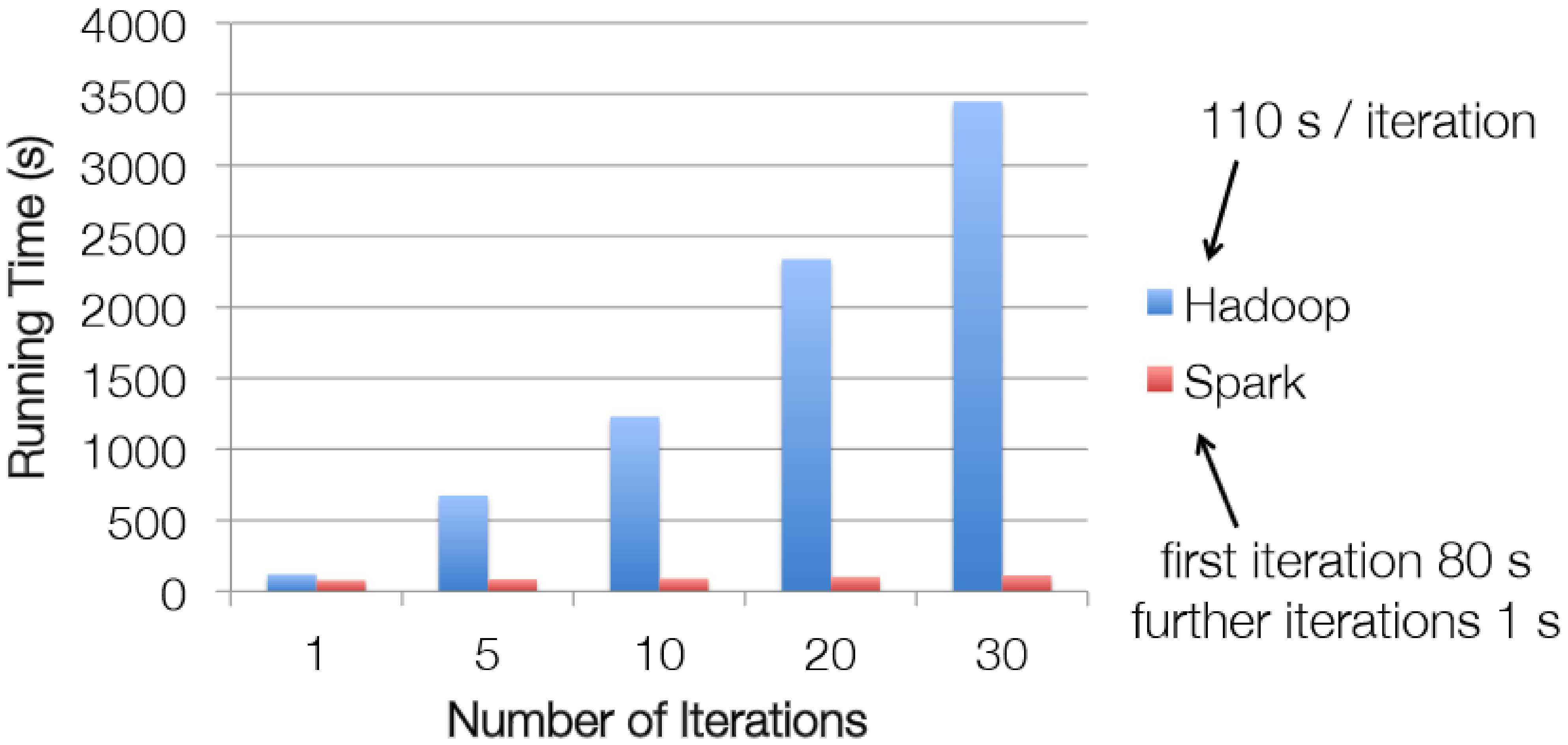
Machine Learning Library (MLlib)

```
points = context.sql("select latitude, longitude from tweets")
model = KMeans.train(points, 10)
```

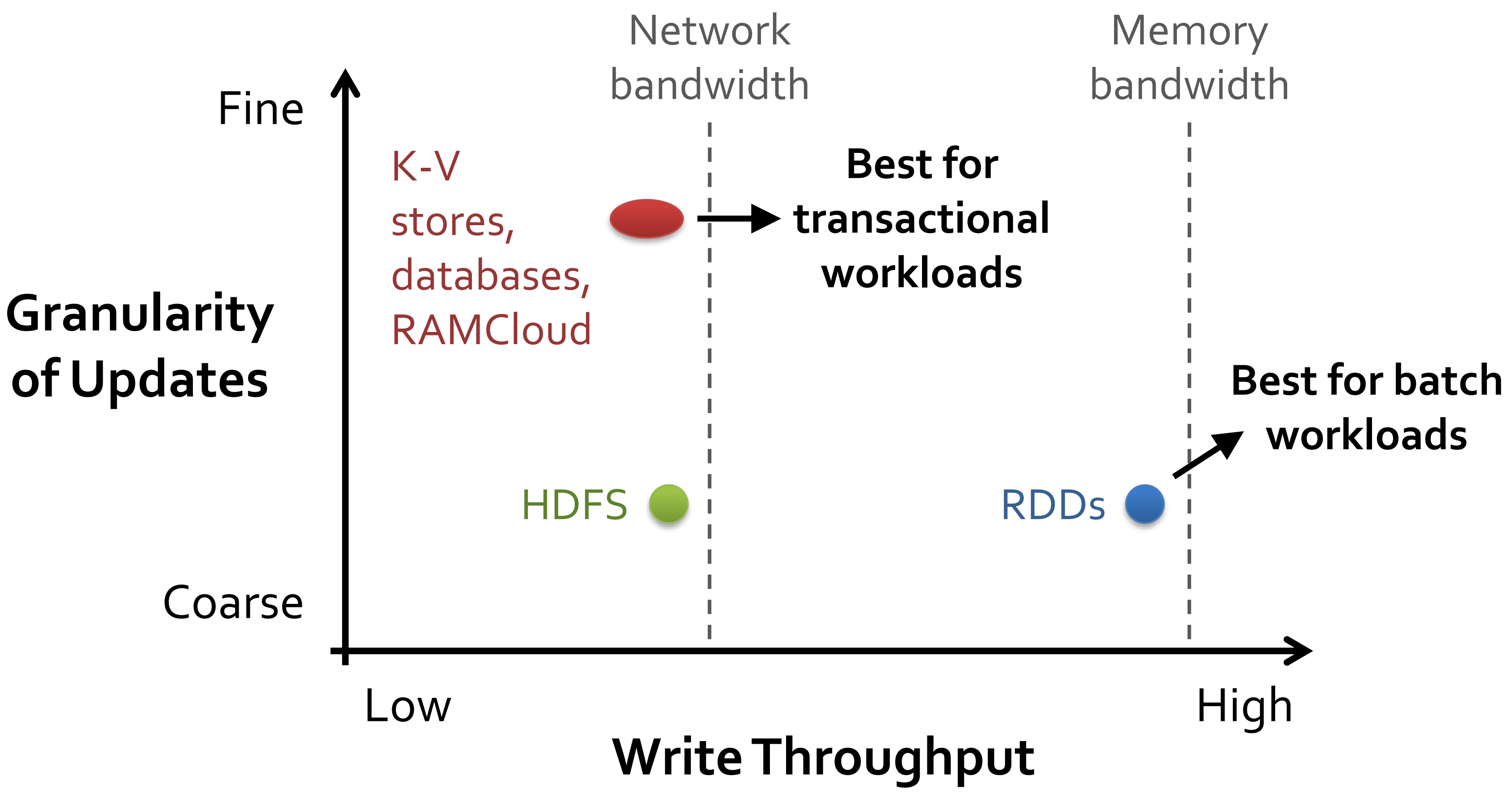
MLlib algorithms

- Classification: logistic regression, linear SVM, naïve Bayes, classification tree
- Regression: generalized linear models (GLMs), regression tree
- Collaborative filtering: alternating least squares (ALS), non-negative matrix factorization (NMF)
- Clustering: k-means
- Decomposition: SVD, PCA
- Optimization: stochastic gradient descent, L-BFGS

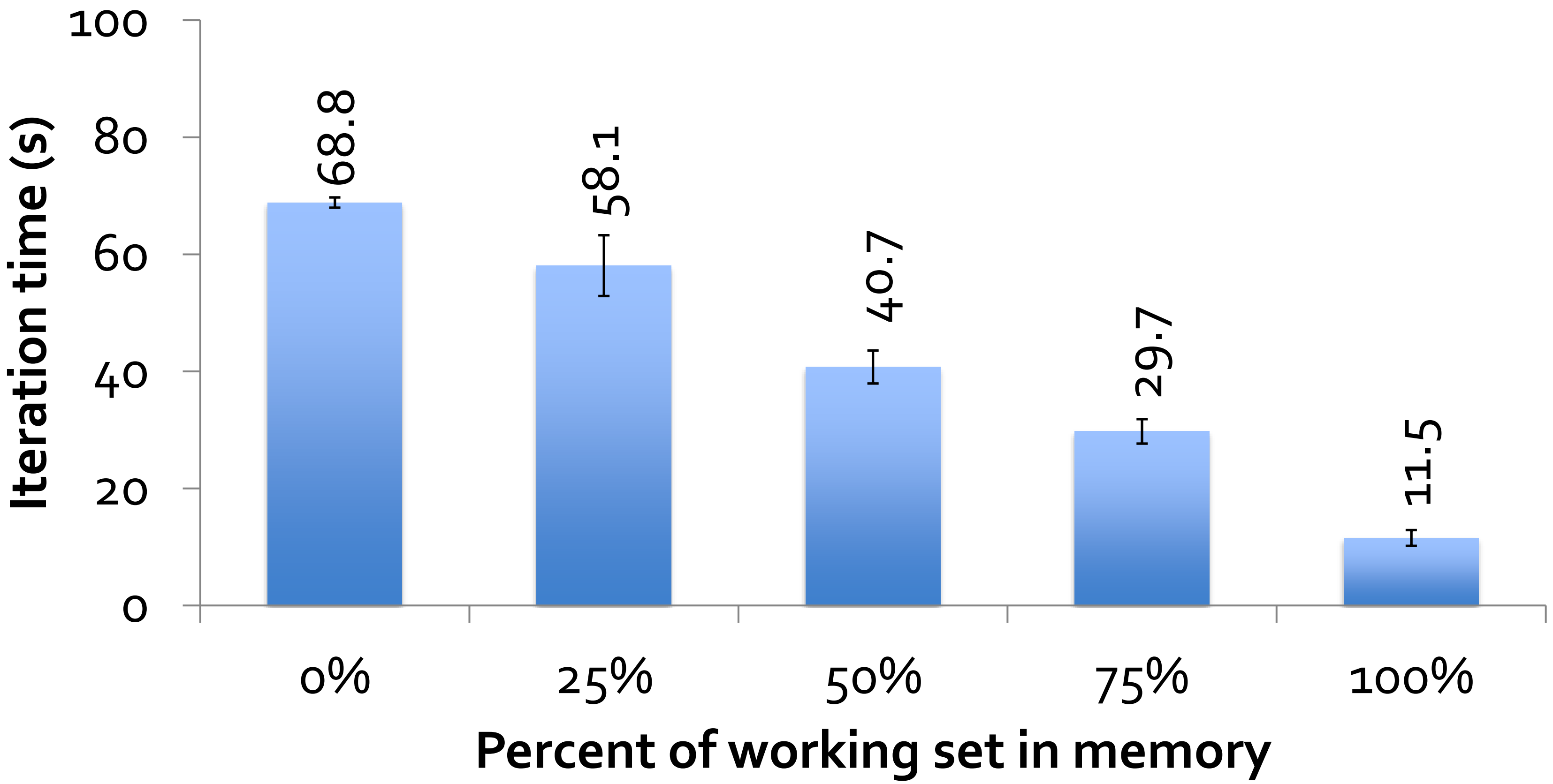
Logistic regression performance



Tradeoff Space



Behavior with Insufficient RAM



Scalability