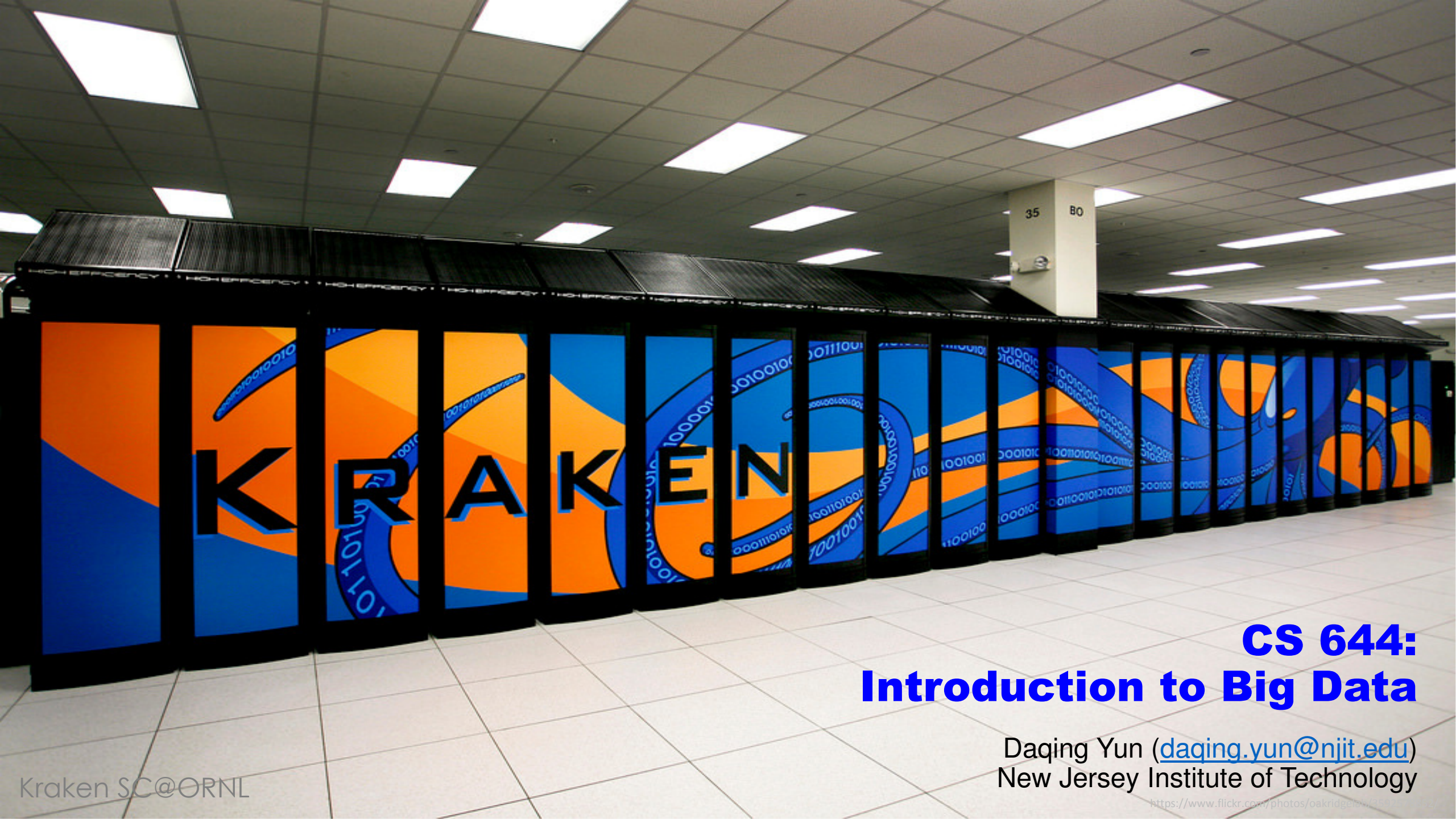


CS 644: Introduction to Big Data

Daqing Yun (daqing.yun@njit.edu)
New Jersey Institute of Technology

Kraken SC@ORNL

https://www.flickr.com/photos/oakridgelab/4501

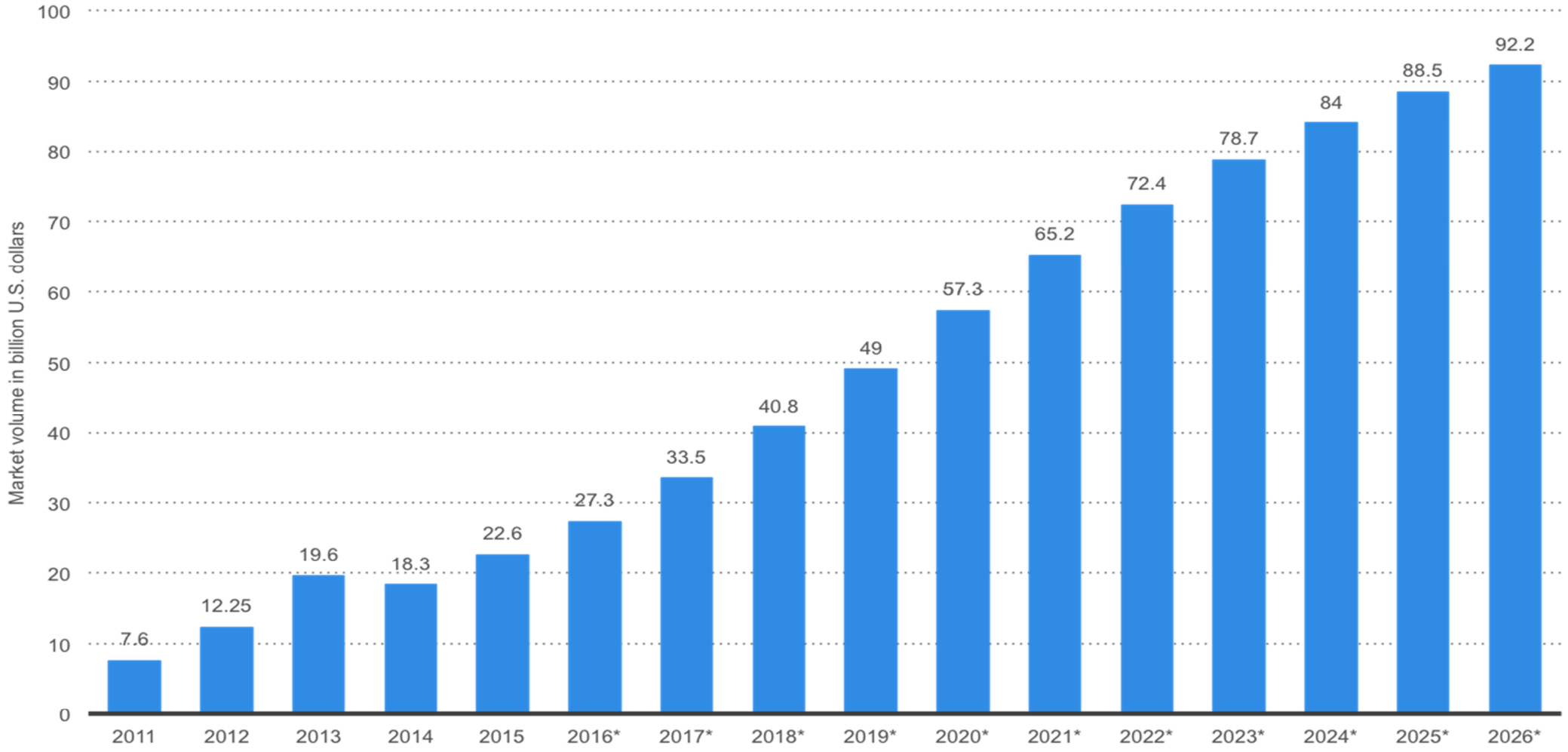
Outline

- Overview
- Big Data Analytics Example Use Cases
- MapReduce

Some of the slides were provided through the courtesy of Dr. Ching-Yung Lin at Columbia University; and based on many other online resources (whose links are included) such as http://www.mmds.org

Forecast revenue big data market worldwide 2011-2026

Big data market size revenue forecast worldwide from 2011 to 2026 (in billion U.S. dollars)



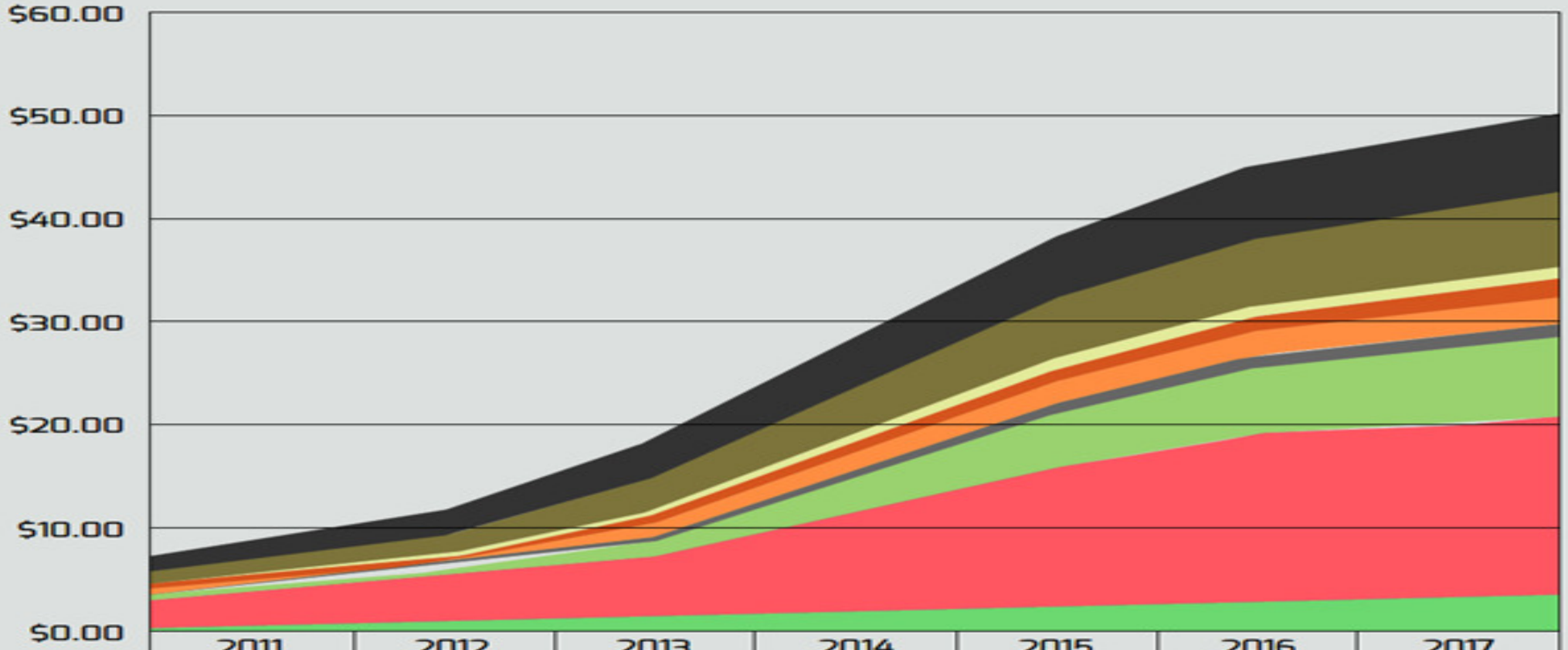
Note: Worldwide; 2014 to 2016

Source: Wikibon; ID 254266

statista

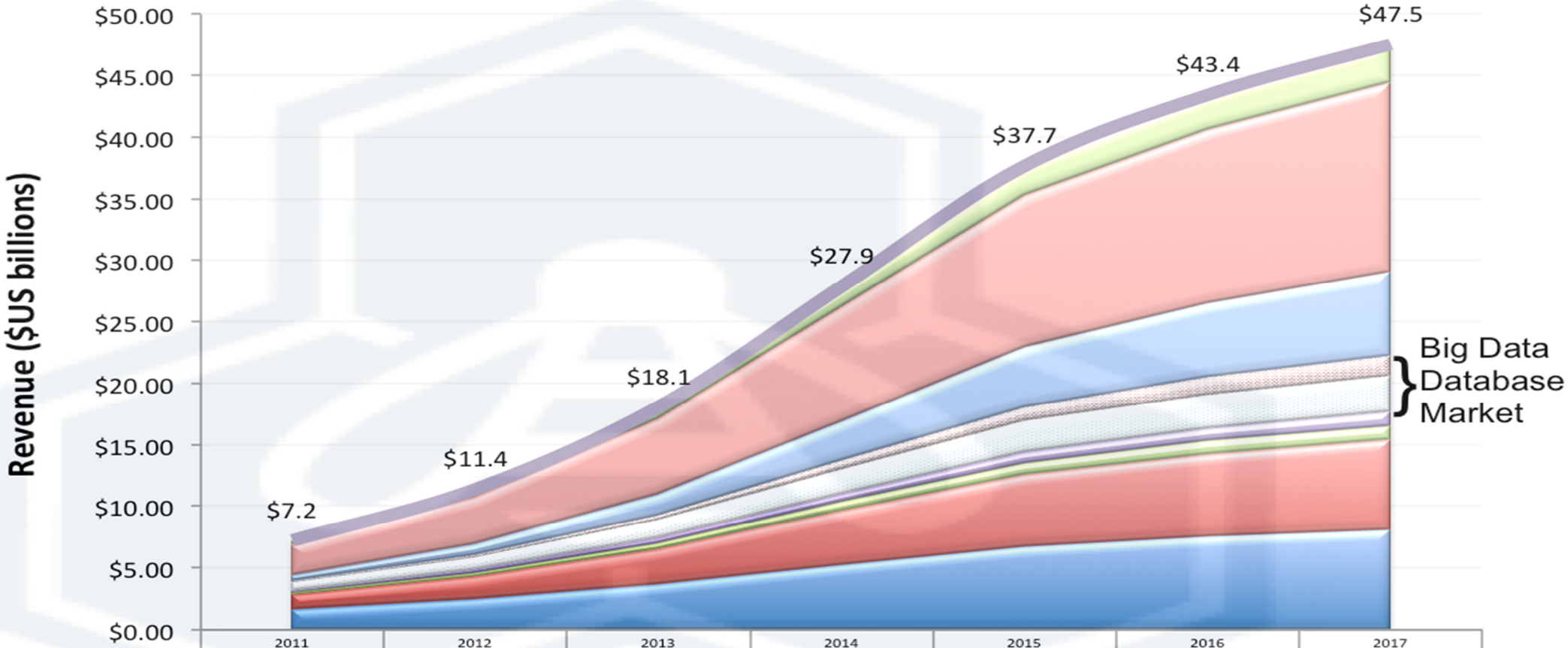
BIG DATA MARKET FORECAST BY SUB-TYPE, 2011-2017 (IN $US BILLIONS)

https://dataanalytics.report/



	2011	2012	2013	2014	2015	2016	2017
Compute	$1.53	$2.29	$3.65	$4.92	$6.40	$7.10	$7.60
Storage	$1.10	$1.75	$3.09	$4.20	$5.50	$6.40	$6.95
Networking	$0.15	$0.23	$0.42	$0.65	$0.85	$1.01	$1.15
Infrastructure Software	$0.14	$0.44	$0.83	$1.08	$1.25	$1.60	$1.90
SQL	$0.62	$0.88	$1.31	$1.75	$2.25	$2.45	$2.70
No SQL	$0.07	$0.13	$0.29	$0.50	$0.80	$1.00	$1.20
Apps & Analytics	$0.52	$0.99	$1.69	$3.45	$5.29	$6.65	$7.75
Professional Services	$2.80	$4.42	$6.15	$10.10	$13.50	$16.00	$17.20
Cloud	$0.36	$0.62	$1.19	$1.82	$2.52	$3.05	$3.65

Big Data Market Forecast by Component, 2011-2017 ($US billions)



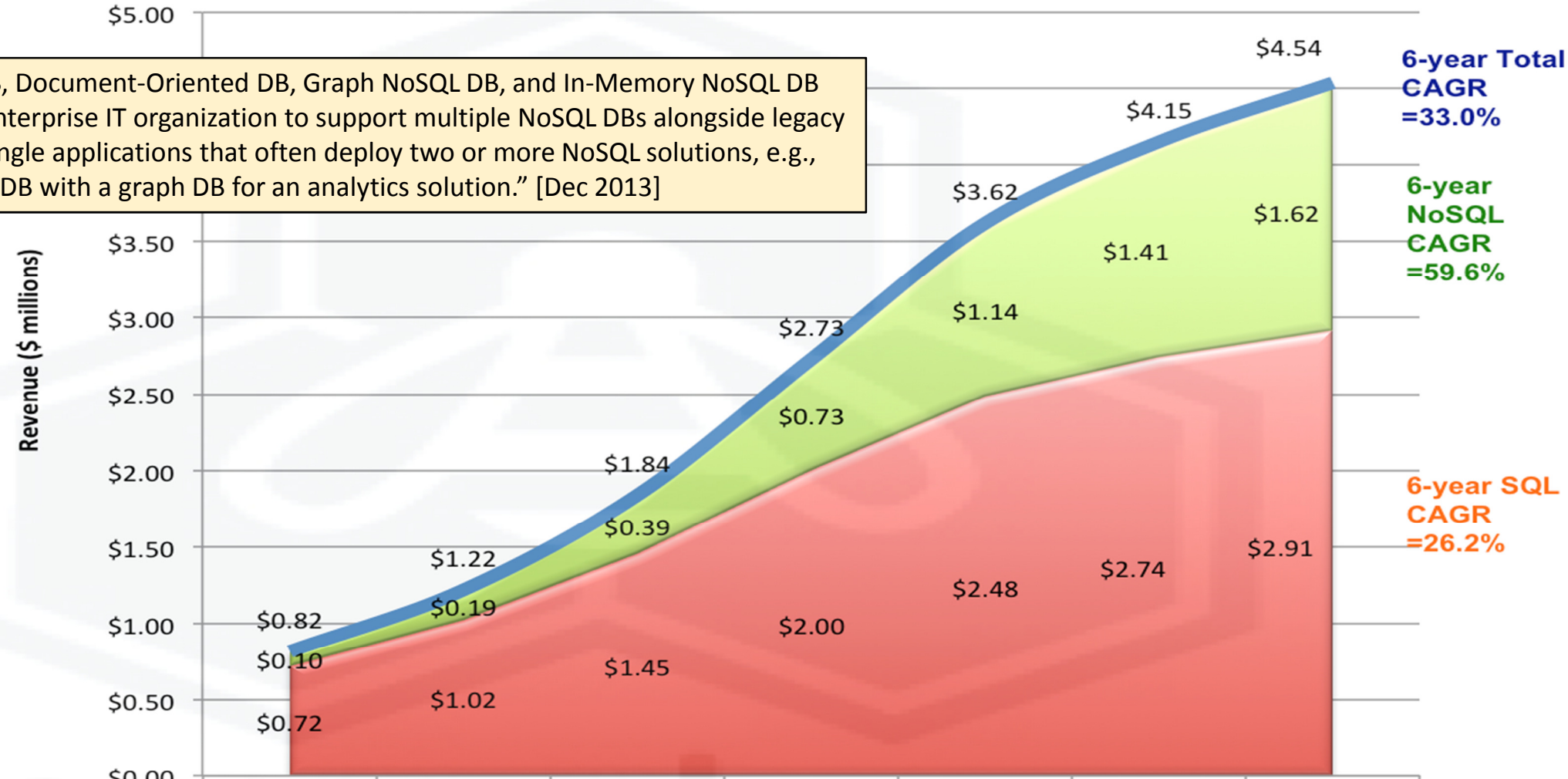
	2011	2012	2013	2014	2015	2016	2017
Big Data XaaS Revenue	$0.34	$0.60	$1.03	$1.71	$2.43	$2.87	$3.19
Big Data Professional Services Revenue	$2.43	$3.85	$6.07	$9.24	$12.31	$14.06	$15.30
Big Data Application (Analytic and Transactional) Revenue	$0.48	$0.93	$1.77	$3.24	$4.94	$6.05	$6.89
Big Data NoSQL Database Revenue	$0.10	$0.19	$0.39	$0.73	$1.14	$1.41	$1.62
Big Data SQL Database Revenue	$0.72	$1.02	$1.45	$2.00	$2.48	$2.74	$2.91
Big Data Infrastructure Revenue	$0.15	$0.25	$0.42	$0.67	$0.93	$1.08	$1.19
Big Data Networking Revenue	$0.18	$0.28	$0.44	$0.67	$0.89	$1.02	$1.11
Big Data Storage Revenue	$1.16	$1.83	$2.88	$4.39	$5.85	$6.68	$7.27
Big Data Compute Revenue	$1.64	$2.45	$3.64	$5.23	$6.70	$7.50	$8.06
Total Big Data Revenue	$7.2	$11.4	$18.1	$27.9	$37.7	$43.4	$47.5
Database as % of Total Big Data Market	11.4%	10.7%	10.2%	9.8%	9.6%	9.6%	9.5%

Source: © Wikibon Big Data Model 2011-2017

http://wikibon.org/wiki/v/Big_Data_Database_Revenue_and_Market_Forecast_2012-2017

Big Data SQL & NoSQL Database Revenue Projection 2011-2017 $M

NoSQL DB ==> Distributed DB, Document-Oriented DB, Graph NoSQL DB, and In-Memory NoSQL DB
"It is not uncommon for an enterprise IT organization to support multiple NoSQL DBs alongside legacy RDBMSs. Indeed, there are single applications that often deploy two or more NoSQL solutions, e.g., pairing a document-oriented DB with a graph DB for an analytics solution." [Dec 2013]



	2011	2012	2013	2014	2015	2016	2017
Big Data NoSQL Database Revenue	$0.10	$0.19	$0.39	$0.73	$1.14	$1.41	$1.62
Big Data SQL Database Revenue	$0.72	$1.02	$1.45	$2.00	$2.48	$2.74	$2.91
Total Big Data Database Revenue	$0.82	$1.22	$1.84	$2.73	$3.62	$4.15	$4.54

Source: © Wikibon Big Data Model 2011-2017

http://wikibon.org/wiki/v/Big_Data_Database_Revenue_and_Market_Forecast_2012-2017

Top 10 Business Intelligence Trends

01

GOVERNANCE AND SELF-SERVICE ANALYTICS BECOME BEST FRIENDS

Organizations have learned that data governance, when done right, can help nurture a culture of analytics and meet the needs of the business.

02

VISUAL ANALYTICS BECOMES A COMMON LANGUAGE

Visual analytics will serve as the common language, empowering people to reach insights quickly, collaborate meaningfully, and build a community around data.

03

THE DATA PRODUCT CHAIN BECOMES DEMOCRATIZED

Self-service analytics tools have changed people's expectations for good. In 2016, people will seek empowerment across the data continuum, especially as more millennials enter the workforce.

04

DATA INTEGRATION GETS EXCITING

With the rise of sophisticated tools and the addition of new data sources, companies will stop trying to gather every byte of data in the same place.

05

ADVANCED ANALYTICS IS NO LONGER JUST FOR ANALYSTS

Organizations will adopt platforms that let users apply statistics, ask a series of questions, and stay in the flow of their analysis.

06

CLOUD DATA AND CLOUD ANALYTICS TAKE OFF

In 2016, more people will transition to the cloud thanks, in part, to tools that help them consume web data. Early adopters are already learning from this data, and others are realizing they should.

07

THE ANALYTICS CENTER OF EXCELLENCE (COE) BECOMES EXCELLENT

An increasing number of organizations will establish a Centre of Excellence to foster adoption of self-service analytics. These centres play a critical role in implementing a data-driven culture.

08

MOBILE ANALYTICS STANDS ON ITS OWN

Mobile analytics has grown up and moved out. It's no longer just an interface to legacy business intelligence products.

09

PEOPLE BEGIN TO DIG INTO IOT DATA

As the volume of IoT data grows, so does the potential for insights. Companies will look for tools that allow users to explore the data, then share their findings in a secure, governed, and interactive way.

10

NEW TECHNOLOGIES RISE TO FILL THE GAPS

In 2016, we'll see the rise of the gap fillers, leading to a market consolidation. And organizations will continue to shift away from single solutions and embrace an open and flexible stack that includes these new technologies.

Year 2016, https://dataanalytics.report/

5 Key Big Data Use Case Categories – IBM's Perspective

Big Data Exploration

Find, visualize, and understand all big data to improve decision making

Enhanced 360º View of the Customer

Extend existing customer views (MDM, CRM, etc.) by incorporating additional internal and external information sources

Security/Intelligence Extension

Lower risk, detect fraud and monitor cyber security in real-time

Operations Analysis

Analyze a variety of machine data for improved business results

Data Warehouse Augmentation

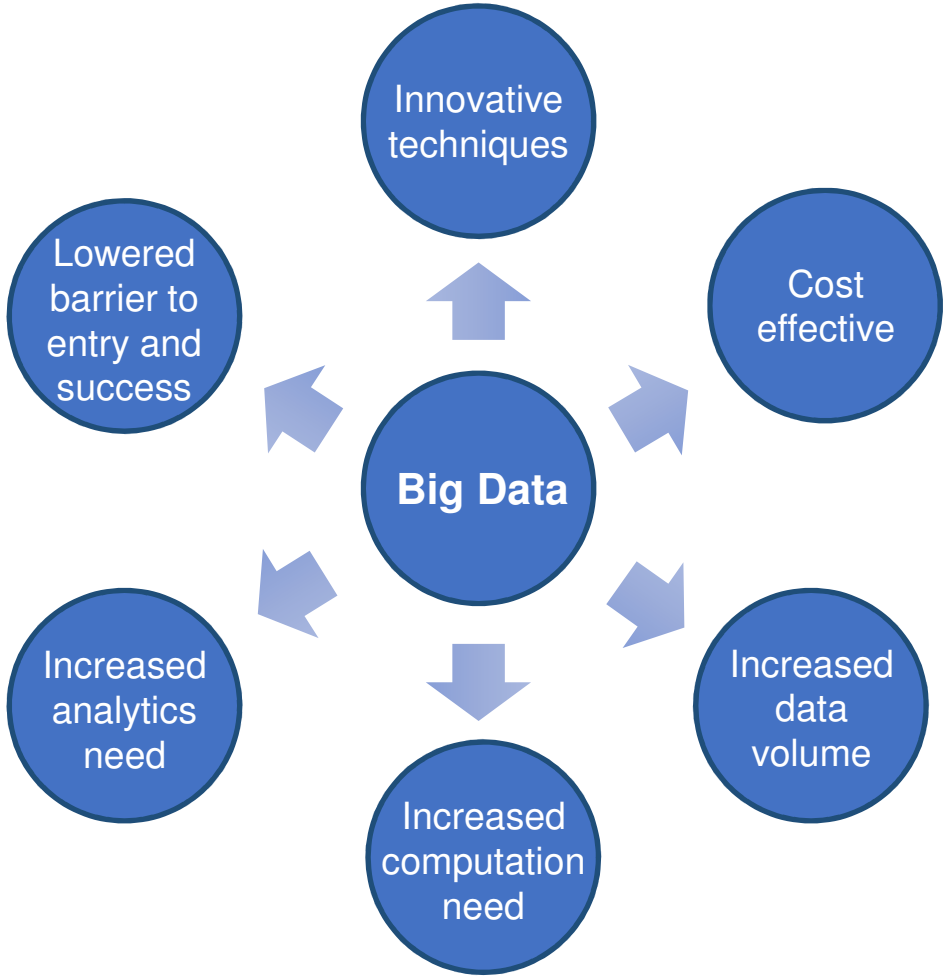
Integrate big data and data warehouse capabilities to increase operational efficiency

Techniques towards Big Data

- Massive Parallelism
- Huge Data Volumes Storage
- Data Distribution
- High-Speed Networks
- High-Performance Computing
- Task and Thread Management
- Data Mining and Analytics
- Data Retrieval
- Machine Learning
- Data Visualization

Techniques exist for years to decades.
Why did Big Data become **hot** now?

What made Big Data needed?



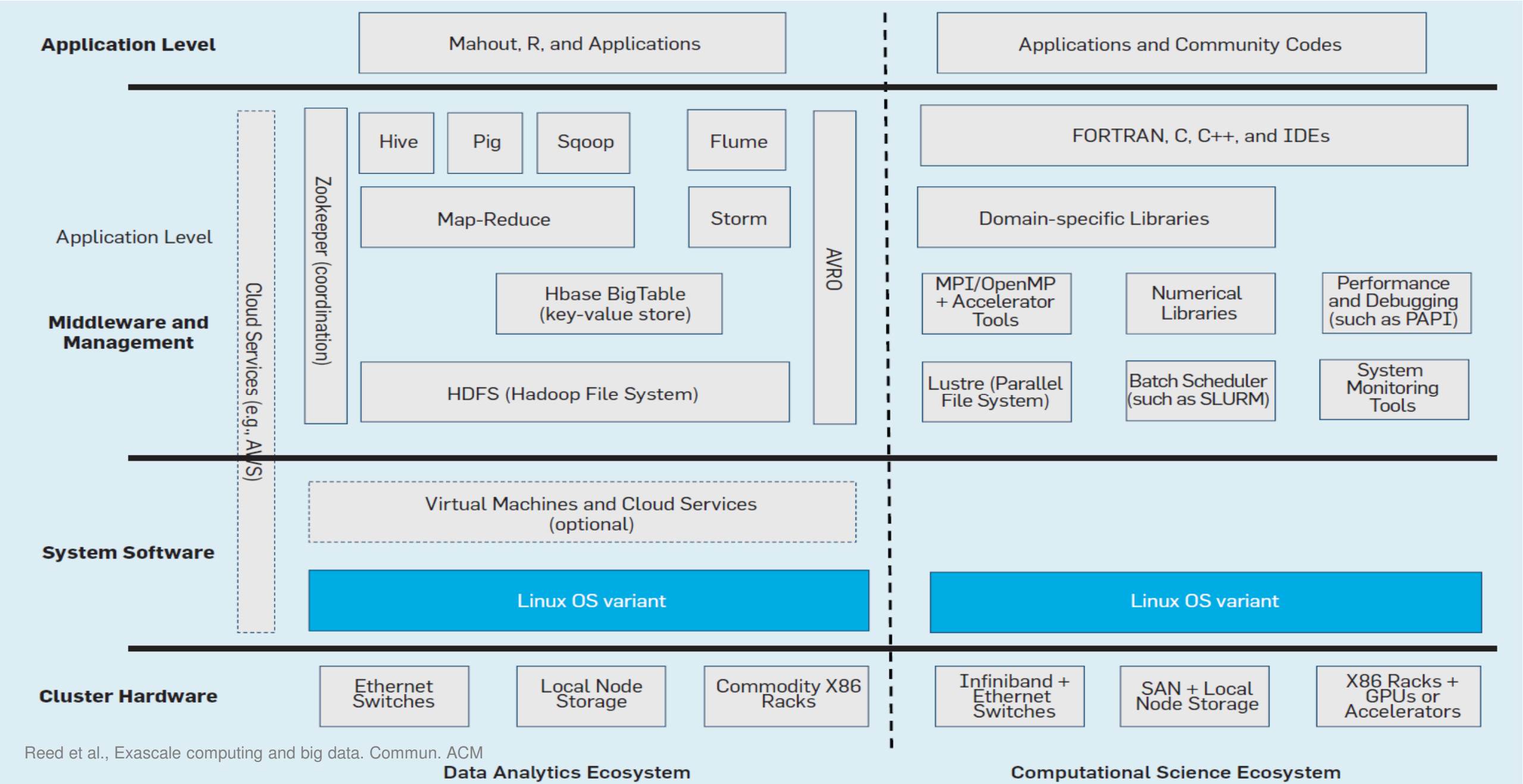
"Big Data Analytics", David Loshin, 2013

Contrasting Approaches in Adopting High-Performance Capabilities

Aspect	Typical Scenario	Big Data
Application development	Applications that take advantage of massive parallelism developed by **specialized developers** skilled in high-performance computing, performance optimization, and code tuning	A simplified application execution model encompassing a distributed file system, application programming model, distributed database, and program scheduling is packaged within **Hadoop, an open source framework for reliable, scalable, distributed, and parallel computing**
Platform	Uses **high-cost** massively parallel processing (MPP) computers, utilizing high-bandwidth networks, and massive I/O devices	Innovative methods of creating scalable and yet elastic virtualized platforms **take advantage of clusters of commodity hardware** components (either cycle harvesting from local resources or through cloud-based utility computing services) coupled with open source tools and Technology
Data management	Limited to file-based or relational database management systems (**RDBMS**) using standard row-oriented data layouts	Alternate models for data management (often referred to as **NoSQL or "Not Only SQL"**) provide a variety of methods for managing information to best suit specific business process needs, such as in-memory data management (for rapid access), columnar layouts to speed query response, and graph databases (for social network analytics)
Resources	Requires large capital investment in **purchasing high-end hardware** to be installed and managed in-house	The ability to **deploy systems like Hadoop on virtualized platforms** allows small and medium businesses to utilize cloud-based environments that, from both a cost accounting and a practical perspective, are much friendlier to the bottom line

"Big Data Analytics", David Loshin, 2013

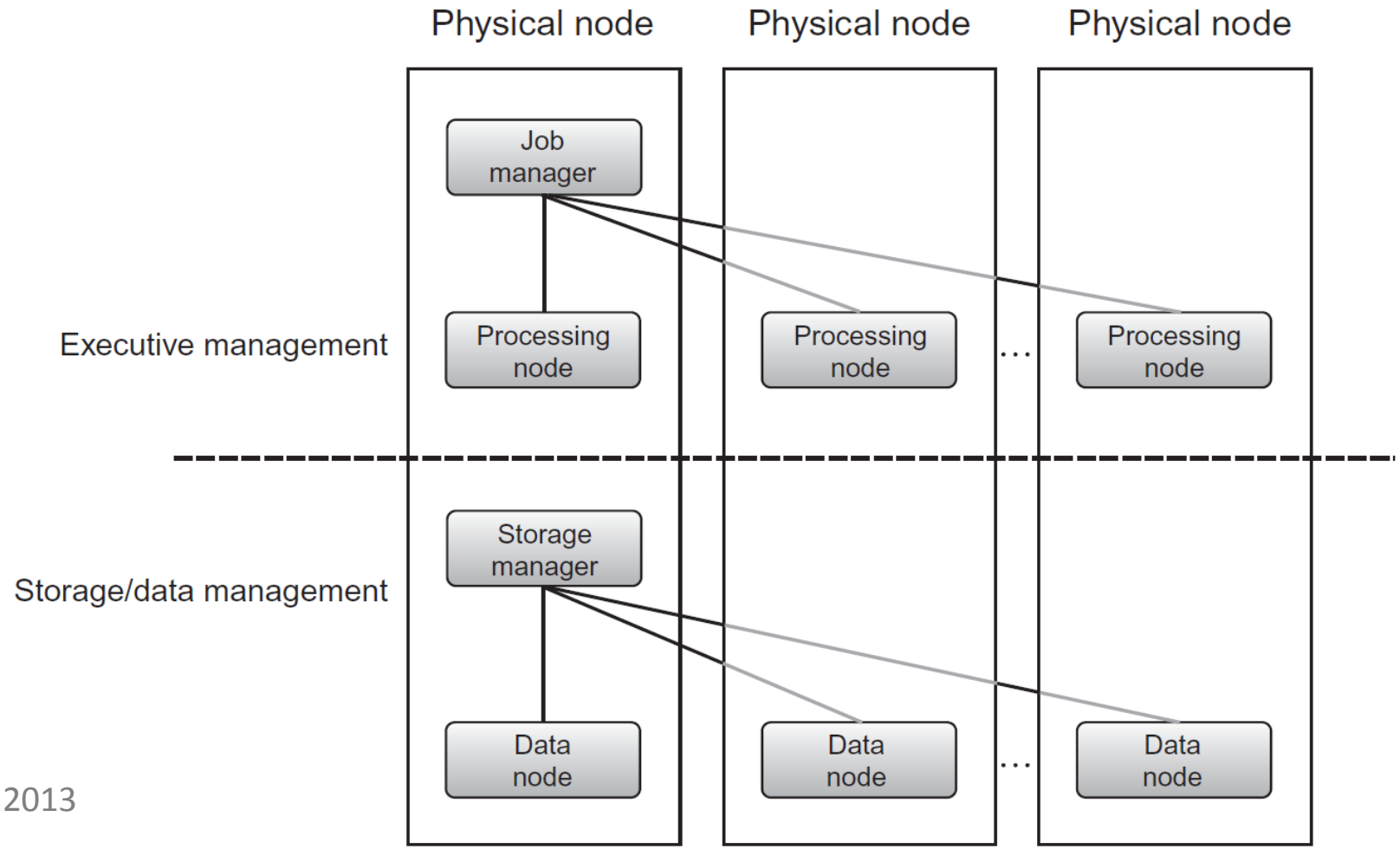
Comparison of Data Analytics and Computing Ecosystems



Reed et al., Exascale computing and big data. Commun. ACM

Key Computing Resources for Big Data

- Processing capability: CPU, processor, or node.
- Memory
- Storage
- Network



"Big Data Analytics", David Loshin, 2013

Apache Hadoop

- The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.
- The Apache Hadoop software library is **a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models**. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.
- This project includes these modules:
  - **Hadoop Common**: The common utilities that support the other Hadoop modules.
  - **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
  - **Hadoop YARN**: A framework for job scheduling and cluster resource management.
  - **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

http://hadoop.apache.org

Hadoop-related Apache Projects

- **Ambari™**: A web based tool for provisioning, managing, and monitoring Hadoop clusters. It also provides a dashboard for viewing cluster health and ability to view MapReduce, Pig, and Hive applications visually.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **HBase™**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive™**: A data warehouse infrastructure that provides data summarization and ad hoc querying
- **Mahout™**: A scalable machine learning and data mining library.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation
- **Spark™**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- **Tez™**: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases.
- **ZooKeeper™**: A high-performance coordination service for distributed applications.

Big Data Analytics Example Use Cases

1. Recommendation
2. Sentiment Analysis
3. Financial Analysis
4. Social Media Monitoring
5. Data Exploration and Visualization
6. Personalized Search
7. Anomaly Detection (Espionage, Sabotage, etc.)
8. Fraud Detection
9. Cybersecurity
10. Celluar Network Monitoring
11. Cloud Monitoring
12. Code Life Cycle Management
13. Traffic Navigation
14. Image and Video Semantic Understanding
15. Genomic Medicine
16. Data Curation
17. Brain Network Analysis
18. Near Earth Object Analysis

Daqing's Amazon

YOUR ORDERS
0 recent orders
View orders

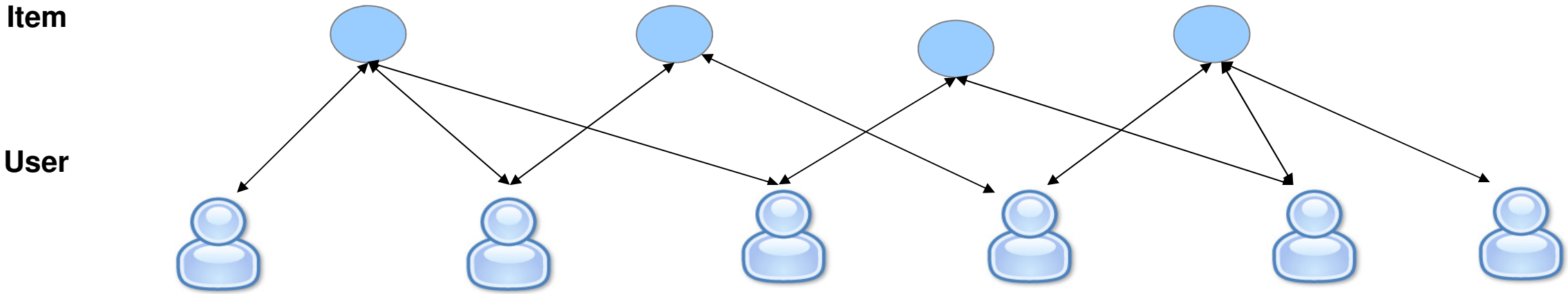
AMAZON PRIME
Try Prime
View benefits

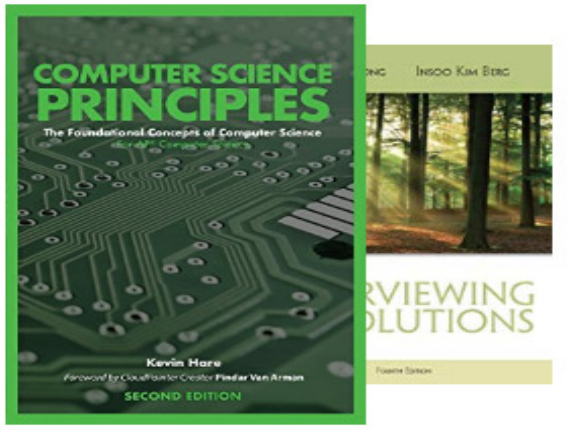
AUDIBLE AUDIOBOOKS
Try Audible
Get 2 free audiobooks

CUSTOMER SINCE
2012

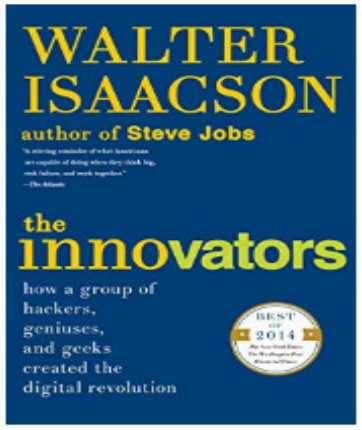
Use Case 1: Recommendation

Recommended for you, Daqing



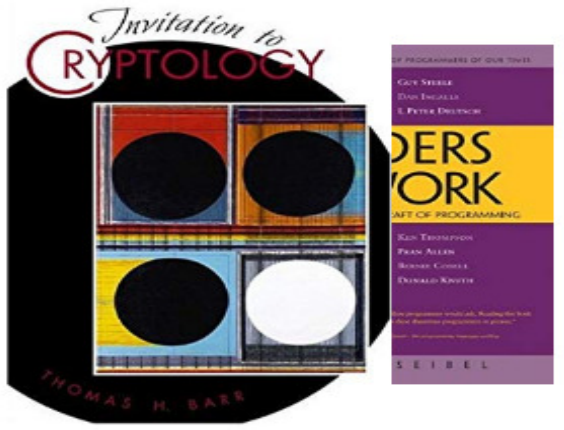


Education & Teaching Books
17 ITEMS



Engineering Books
30 ITEMS

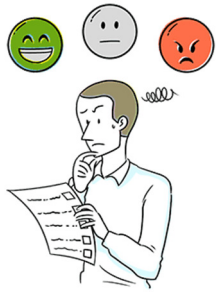
Science & Math Books
64 ITEMS



Business Books
12 ITEMS

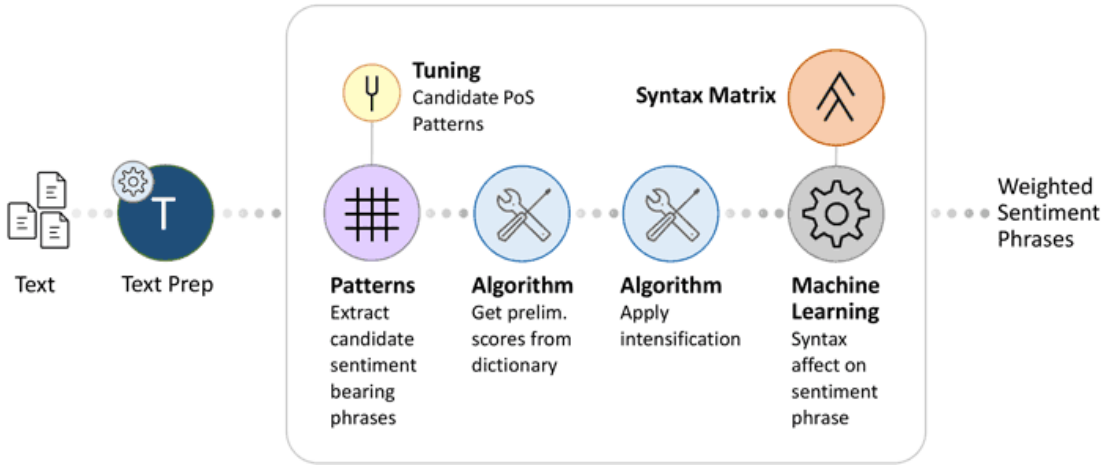
Use Case 2: Sentiment Analysis

- The explosion of social media and the proliferation of mobile devices have created opportunities for customers to express their feelings and attitudes about anything and everything at anytime in the form of reviews, chats, shares, likes tweets, etc., often includes comments that can be invaluable for businesses looking to improve products and services, make more informed decisions, and better promote their brands.
- The key to business success with sentiment data lies in the ability to mine vast stores of unstructured social data for actionable insights.
- That requires sophisticated tools such as Natural Language Processing (NLP) to carry out comprehensive examinations of the sentiments of social media users
- Big data analytics platforms such as cloud-based Hadoop are up to the challenge.



Sentiment analysis

- Enhancing the customer experience
- Gaining competitive advantage
- Gaining Business Intelligence
- Revitalizing a brand



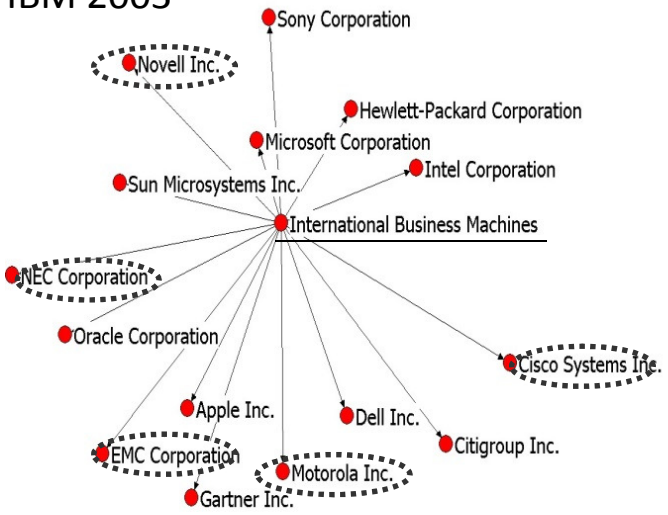
Hybrid sentiment analysis systems combine natural language processing with machine learning to identify weighted sentiment phrases within their larger context

https://www.lexalytics.com/technology/sentiment

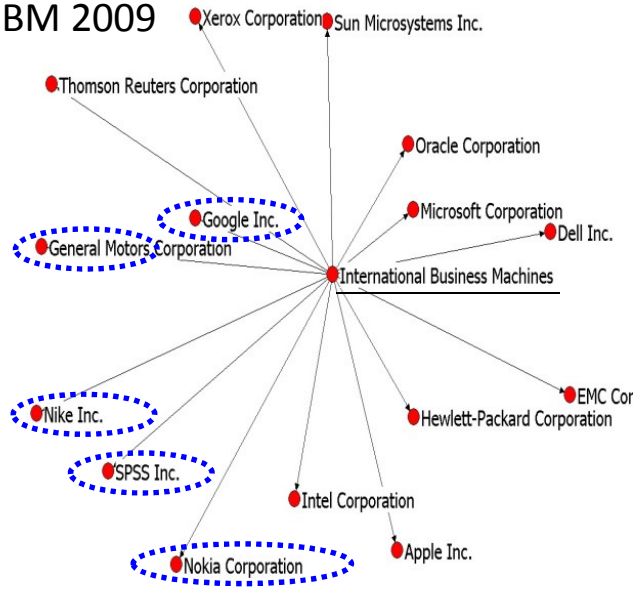
Use Case 3: Graph Analytics for Financial Analysis

Goal: Injecting Network Graph Effects for Financial Analysis. Estimating company performance considering correlated companies, network properties and evolutions, causal parameter analysis, etc.

- IBM 2003



- IBM 2009

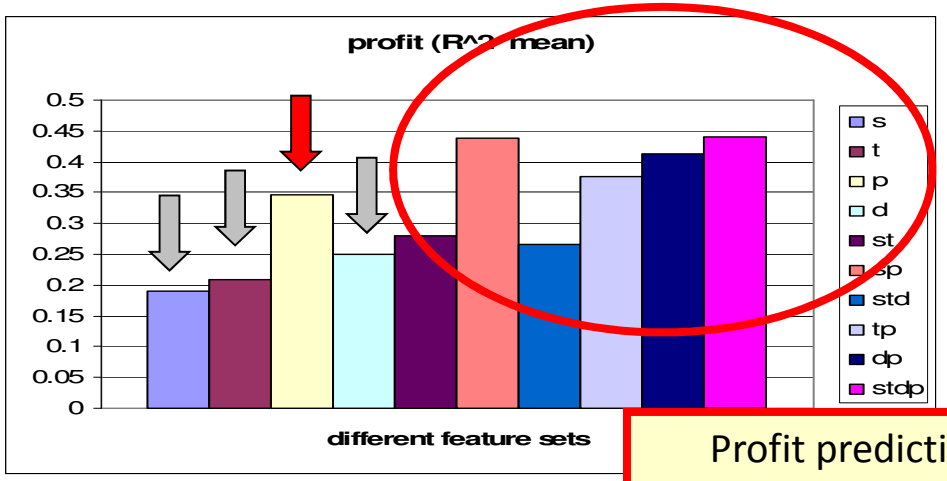


- **Data Source:**
  - Relationships among 7594 companies, data mining from NYT 1981 ~ 2009

Targets: 20 Fortune companies' normalized Profits

Goal: Learn from previous 5 years, and predict next year

Model: Support Vector Regression (RBF kernel)



Network feature:
s (current year network feature),
t (temporal network feature),
d (delta value of network feature)

Financial feature:
p (historical profits and revenues)

Profit prediction by joint network and financial analysis outperforms network-only by 130% and financial-only by 33%.

Use Case 4: Social Media Monitoring

Social media monitoring is a process of using social media channels to track, gather and mine the information and data of certain individuals or groups to assess their reputation and discern how they are perceived online.

Application examples:

- Public safety
  - Seemingly-unrelated case files connection
  - Crime prediction
  - Network oversight and cyber security
  - Threats migration
- Social media marketing
  - Visio recognition
  - Personality insights
  - Promotion
  - Placement
  - Product
- …

- The case files can be related and ranked based on their content, telling users not only that the documents are related, but to present them in a list ranked by how closely they are related.

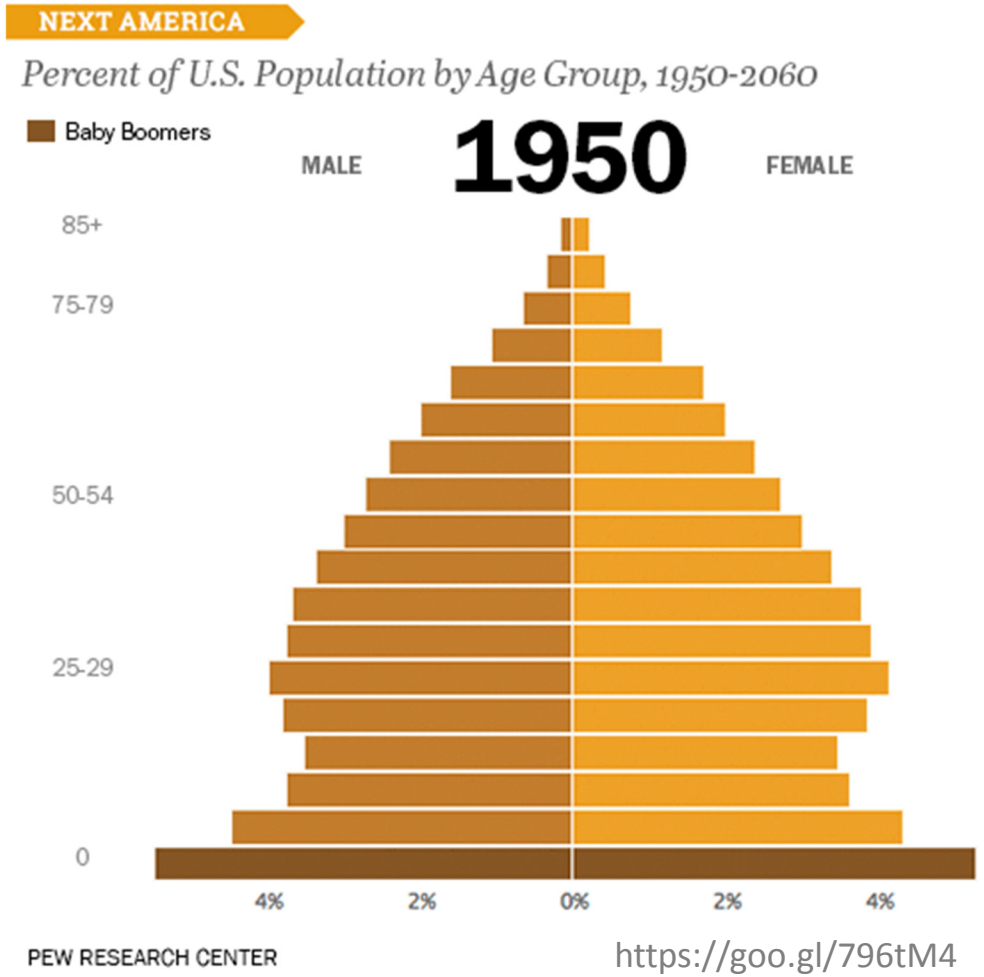
- The benefit to police officers, investigators, and intelligence officials is that instead of reaching out across offices and departments, these documents are automatically connected by the key words and phrases they contain.

- Explore the trends and relationships that exist between the surrounding circumstances and the criminal act itself.

- Reach out to customers for information gathering.
- Customer intelligence to determine current industry trends via mining blogs, wikis, news sites, etc. available outlets, social networking sites, forums, video/photo sharing sites and message boards
- Using a variety of social media monitoring tools, e.g., customized data mining software and Web crawlers.
- Search for specific keywords on various websites and blogs.
- Transfers the desired words and phrases from unstructured to structured database data for analysis with traditional data mining techniques.

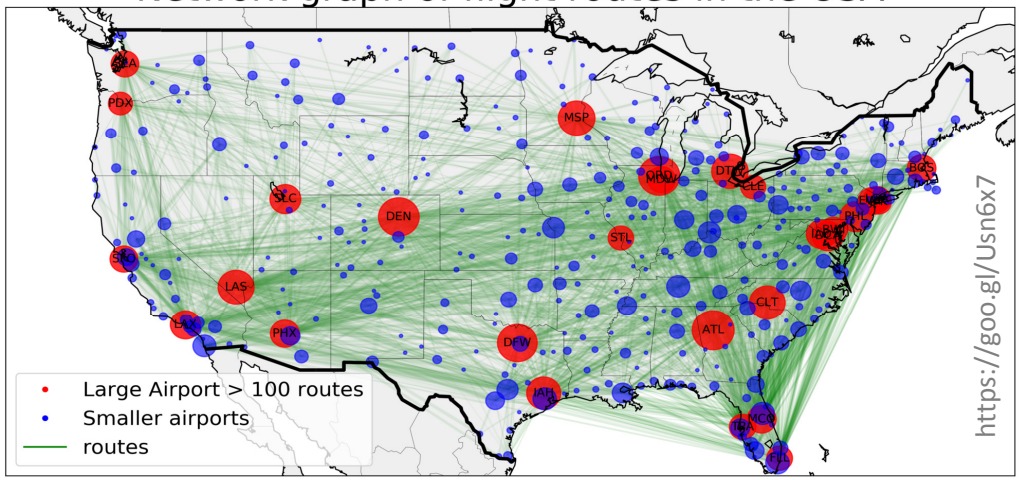
https://webstarkmedia.com/what-is-social-media-marketing/ https://tech.co/big-data-social-media-marketing-2017-09 https://emcien.com/res_guides/Big-Data-for-Public-Safety.pdf

User Case 5: Data Visualization and Exploration

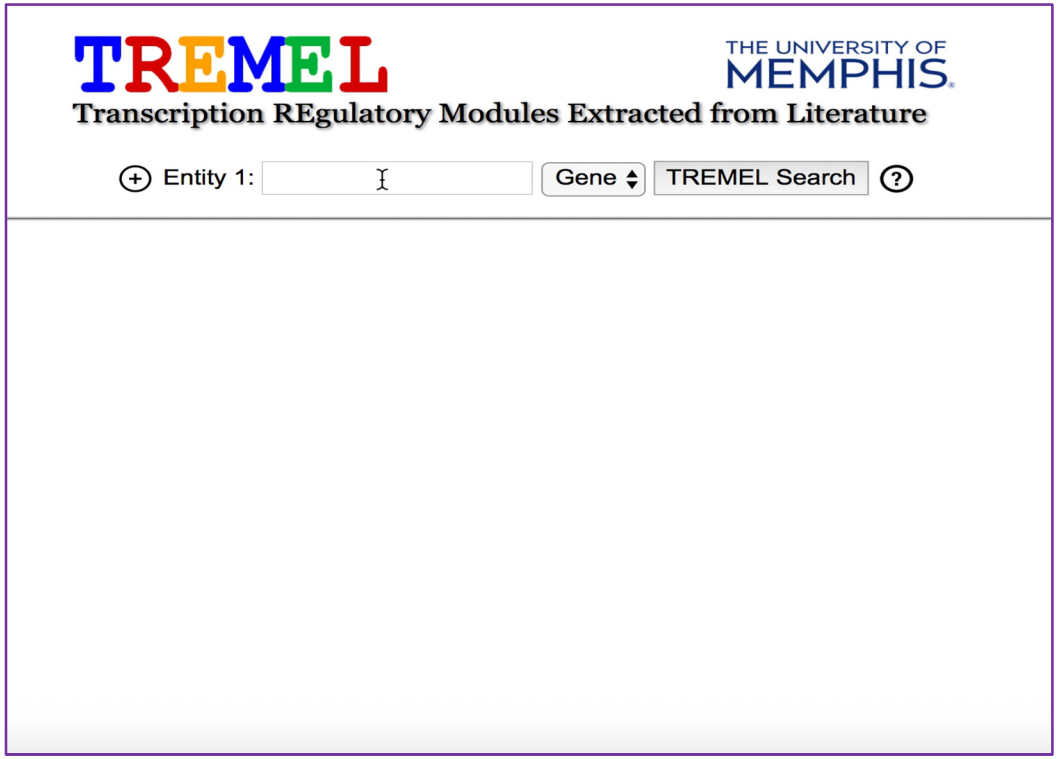


https://goo.gl/796tM4

Network graph of flight routes in the USA

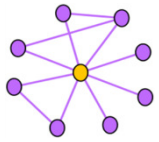


Query based (huge) graph visualization and interaction

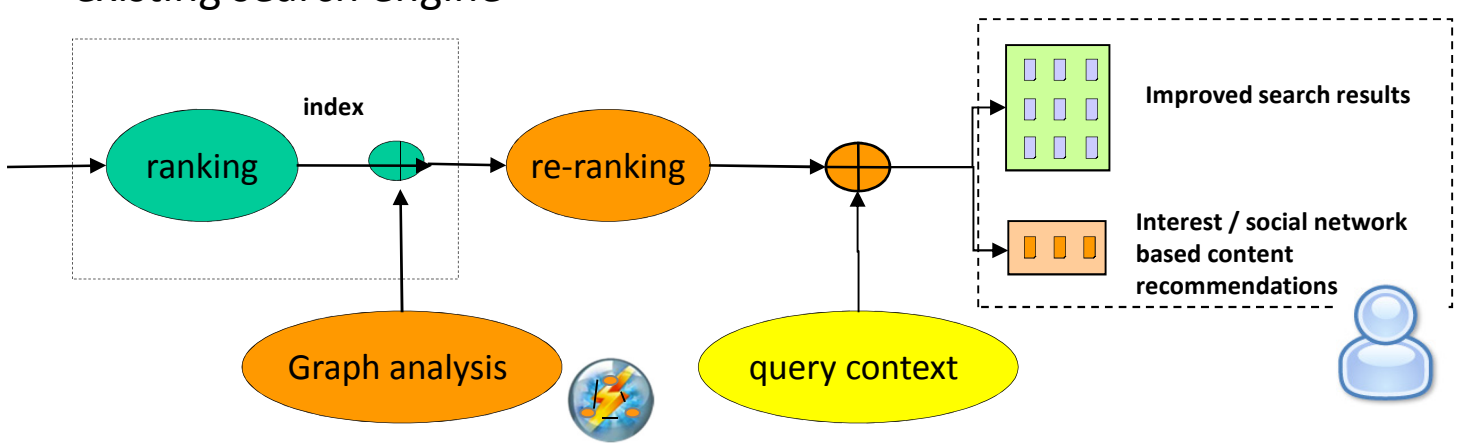


http://binf1.memphis.edu/tremel/

User Case 6: Personalized Search



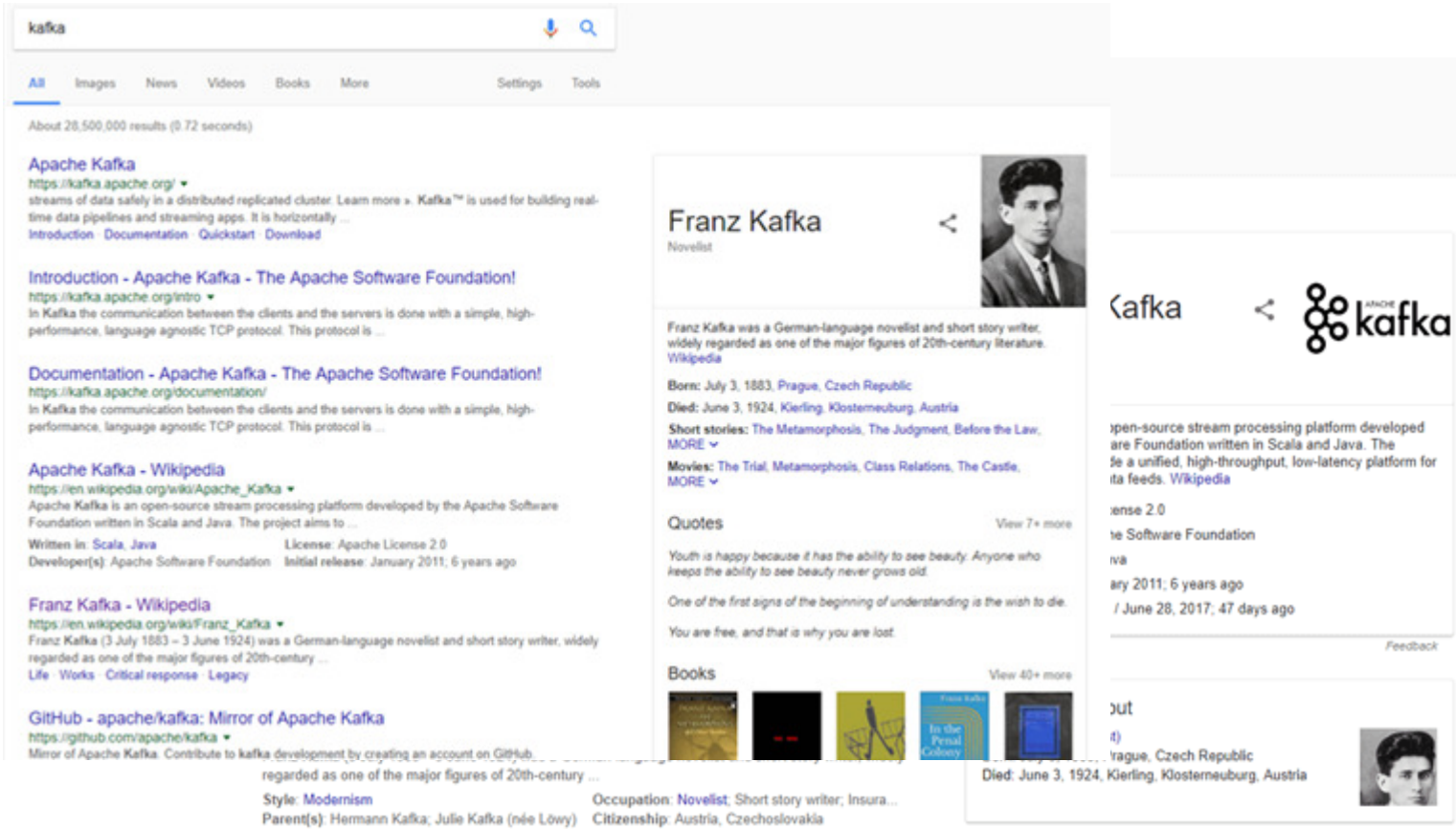
Info-Socio networks



- Personalized search results are the results a user sees in a search engine that aren't just based on the traditional ranking factors (such as the relevance of the web pages to the search term or their authority), but also on the information that the search engine has about the user at the given time, such as their location, search history, demographics, or interests.

- Although widely debated, the purpose of personalized search is to increase the relevance of the results for the particular user.

- Both Google and Bing (and hence Yahoo, too) are personalizing their search results, but Google seems to be doing it for a larger share of searches. Back in 2011, a small experiment showed that over 50% of Google searches were being personalized; that number has likely only gone up since.



https://www.link-assistant.com/news/personalized-search.html

User Case 7: Anomaly Detection at Multiple Scale

Based on President Executive Order 13587

Goal: System for Detecting and Predicting Abnormal Behaviors in Organization, through large-scale social network & cognitive analytics and data mining, to decrease insider threats such as espionage, sabotage, colleague-shooting, suicide, etc.


The White House
Office of the Press Secretary

For Immediate Release October 07, 2011

Executive Order 13587 -- Structural Reforms to Improve the Security of Classified Networks and the Responsible Sharing and Safeguarding of Classified Information

By th and the laws of the United States of America and in order to ensure the responsible sharing and


THE WALL STREET JOURNAL.

Many Past Espionage Cases Had Links to

U.S. Ups Ante for Spying On Firms


npr

To Catch Worker Misconduct, Companies Hire Corporate Detectives

by AILSA CHANG

January 10, 2013 6:25 PM

"Enterprise Information Leakage Impacted economy and jobs" Feb 2013

"What's emerged is a multibillion dollar detective industry" *npr Jan 10, 2013*

Emails

Instant Messaging

Web Access

Executed Processes

Printing

Copying

Log On/Off

Social sensors

Click streams capturer

Feed subscription

Database access

Graph analysis

Behavior analysis

Semantics analysis

Psychological analysis

Multimodality Analysis

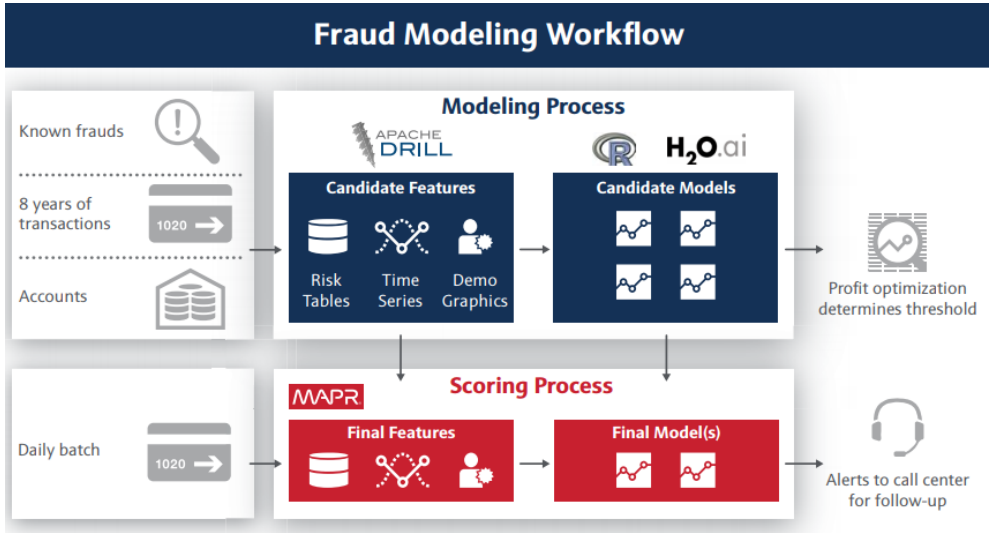
Detection, Prediction & Exploration Interface

Infrastructure + ~ 70 Analytics

Use Case 8: Fraud Detection for Bank

Fraud Detection Solution at a Large Regional Bank

- **Discovery**: understand the nuances of the bank's business and its customers
- **Solution Building**: data scientists work with domain experts to develop solution (model) iteratively
- **Transition**: knowledge transfers to help the client understand what was built and how to use it

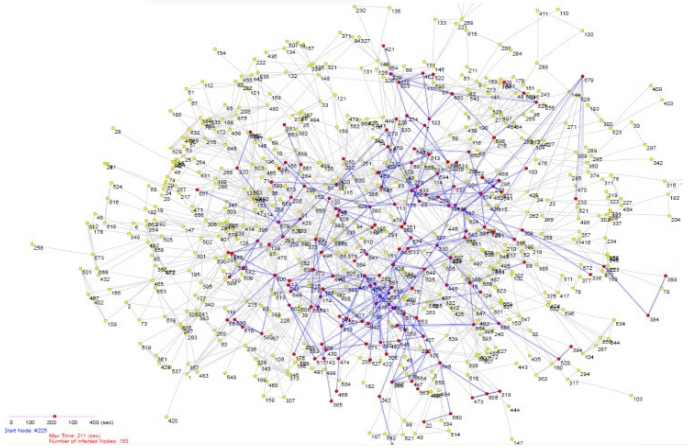


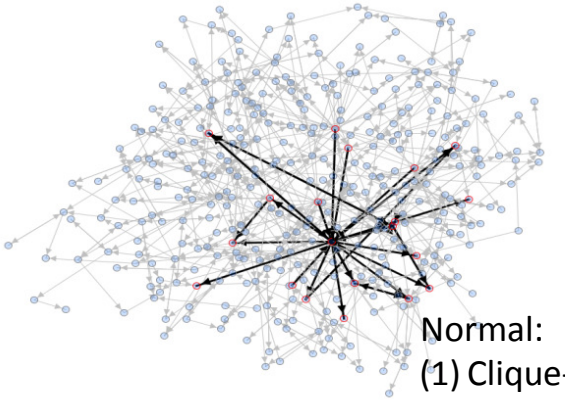
https://mapr.com/resources/fraud-detection-solution-large-regional-bank/

Network Info Flow

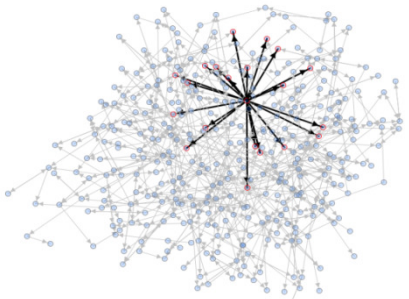
Ponzi scheme Detection

Ego Net Features





Normal:
(1) Clique-like
(2) Two-way links



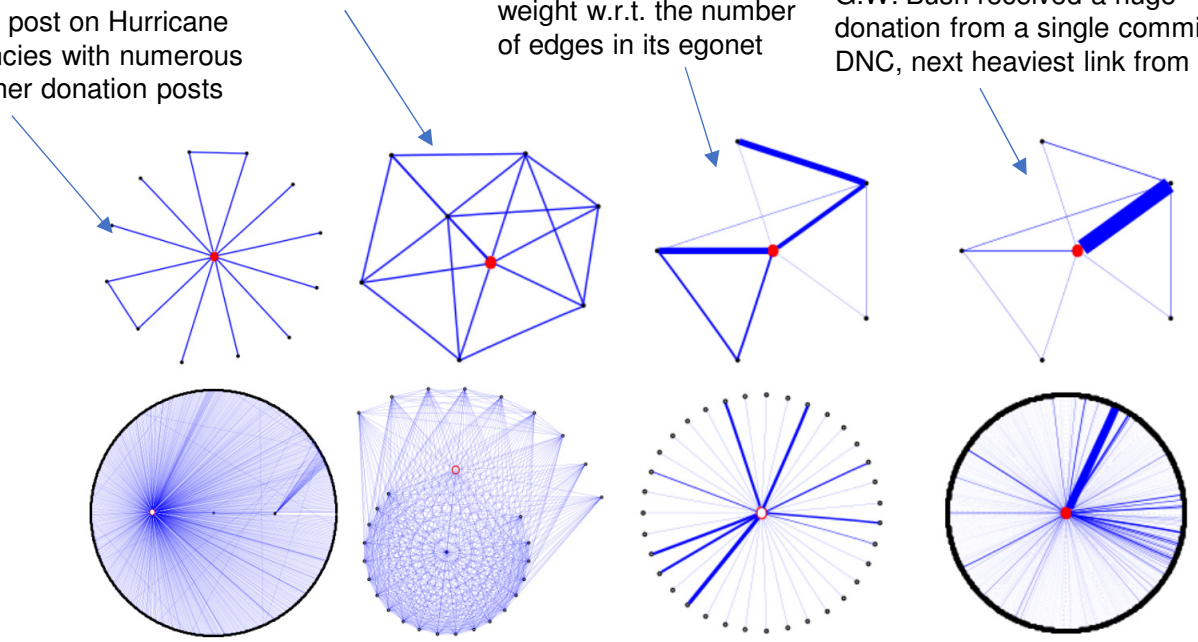
Attacker:
Near-Star

often linked to its own posts, as well as to its own posts in other blogs

An extremely long post on Hurricane Katrina relief agencies with numerous links to diverse other donation posts

has abnormally high weight w.r.t. the number of edges in its egonet

G.W. Bush received a huge donation from a single committee: DNC, next heaviest link from RNC



Akoglu et al.,. OddBall: spotting anomalies in weighted graphs. In Proc. PAKDD 2010.

Use Case 9: Detecting Cyber Attacks

- Intelligence driven security is being increasingly adopted by the industry, and promises a radically different, much more effective model of security using Big Data thinking and technologies.
- When we comprehensively understand the context of normal behavior of people and the flow of data over networks, we are able to transcend the reactive models of the past to more clearly and quickly spot even the faint signal of any impending attack or intrusion in the midst of an increasingly noisy environment.

Big data solution

- Maintain a log of malicious and suspicious IPs (preferably enormous amount) that has initiated attacks in the past -- lead to more accurate detection

- Handle huge chunks of data and have complete logs of raw data -- to derive exploratory analytics and help defenders stay ahead of the attackers

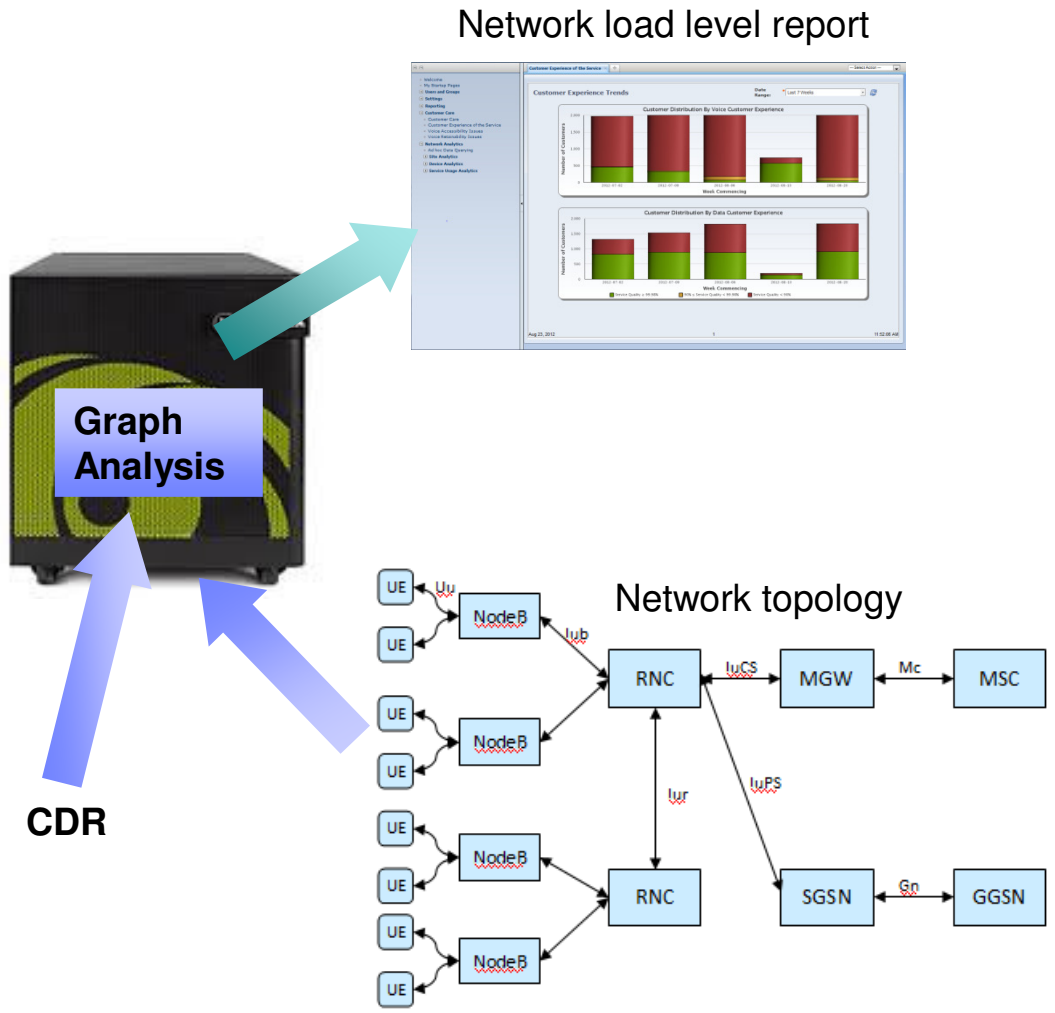
- Cloud-based solutions operated through APIs allowing for specific responses, rather than static policies -- can effectively collaborate with multiple vendors and low-cost mitigation systems

- Provide unified visibility to gain insights into other information in addition to the attack data -- e.g., help monitoring traffic flow, network performance, routing data, and device/interface data

Use Case 10: Cellular Network Analytics

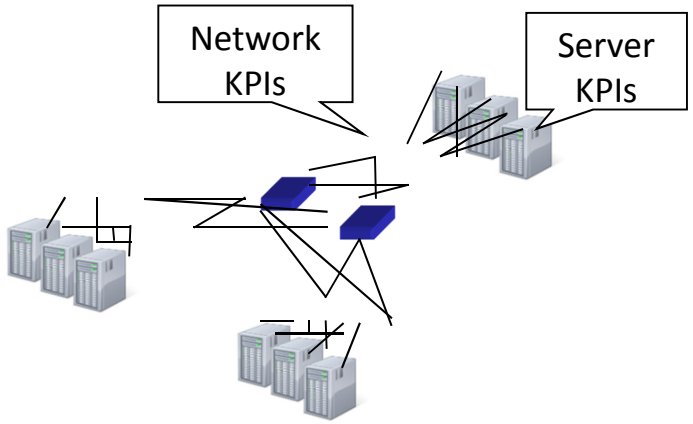
Goal: Efficiently and uniquely identify *internal* state of Cellular/Telco networks (e.g., performance and load of network elements/links) using probes between monitors placed at selected network elements & endhosts

- Applied Graph Analytics to telco network analytics based on CDRs (call detail records): estimate traffic load on CSP network with low monitoring overhead
  - CDRs, already collected for billing purposes, contain information about voice/data calls
  - Traditional NMS* and EMS** typically lack of end-to-end visibility and topology across vendors
  - Employ graph algorithms to analyze network elements which are not reported by the usage data from CDR information
- Approach
  - Cellular network comprises a hierarchy of network elements
  - Map CDR onto network topology and infer load on each network element using graph analysis
  - Estimate network load and localize potential problems



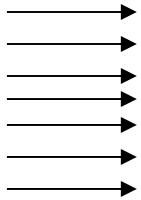
Use Case 11: Monitoring Large Cloud

Goal: Monitoring technology that can track the time-varying state (e.g., causality relationships between KPIs) of a large Cloud when the processing power of monitoring system cannot keep up with the scale of the system & the rate of change

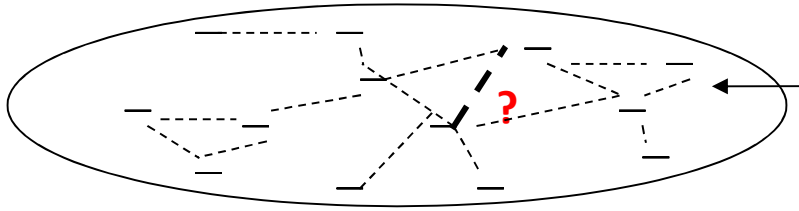


- *Causality relationships (e.g., Granger causality) are crucial in performance monitoring & root cause analysis*
- *Challenge: easy to test pairwise relationship, but hard to test multi-variate relationship (e.g., a large number of KPIs)*

KPI time series (e.g., server performance/load, network performance/load)



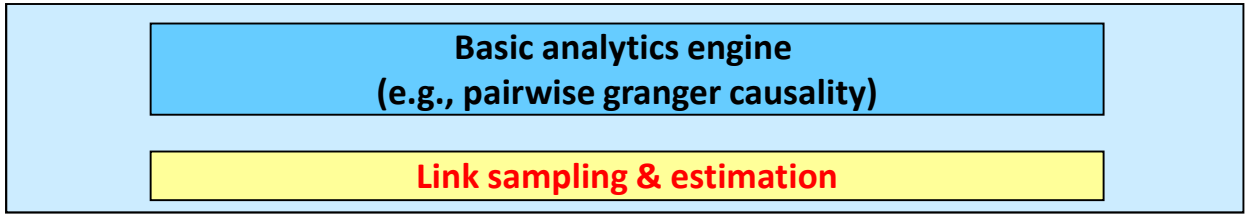
Causality analyzer



KPI (a time series)

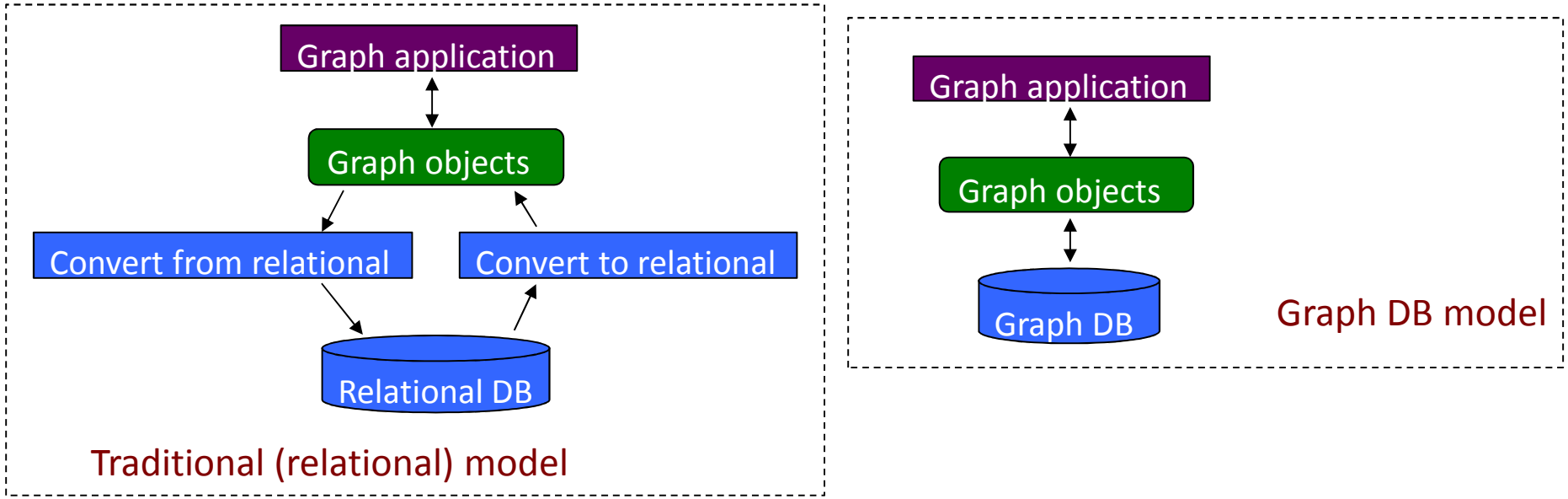
(potential) pairwise relationship (e.g., causality)

Our approach: Probabilistic monitoring via sampling & estimation



Select KPI pairs (sampling)→ Test link existence → Estimate unsampled links based on history → Overall graph

Use Case 12: Code Life Cycle Improvement



- Advantages of working directly with graph DB for graph applications
  - Smaller and simpler code
  - Flexible schema → easy schema evolution
  - Code is easier and faster to write, debug and manage
  - Code and Data is easier to transfer and maintain

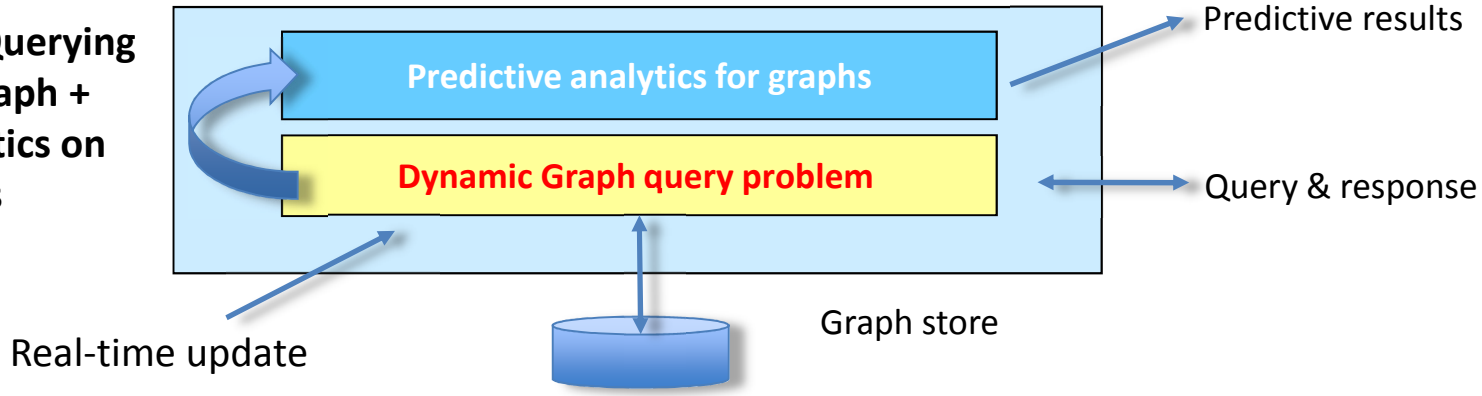
Use Case 13: Smart Navigation Utilizing Real-time Road Information

Goal: *Enable unprecedented level of accuracy in **traffic scheduling** (for a fleet of transportation vehicles) and navigation of individual cars utilizing the **dynamic real-time information** of changing road condition and predictive analysis on the data*

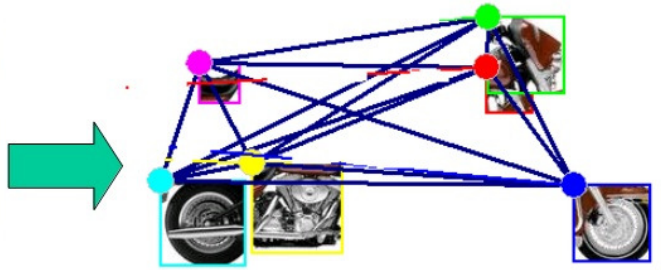
- *Dynamic graph algorithms implemented in System G provide **highly efficient graph query computation** (e.g. shorted path computation) on time-varying graphs (order of magnitudes improvement over existing solutions)*

- *High-throughput **real-time predictive analytics** on graph makes it possible to estimate the future traffic condition on the route to make sure that the decision taken now is optimal overall*

Our approach: Querying over dynamic graph + predictive analytics on graph properties

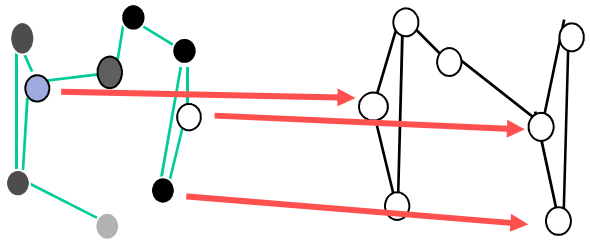


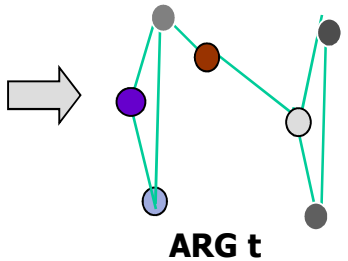
Use Case 14: Graph Analysis for Image and Video Analysis



Vertex Correspondence

Attribute Transformation

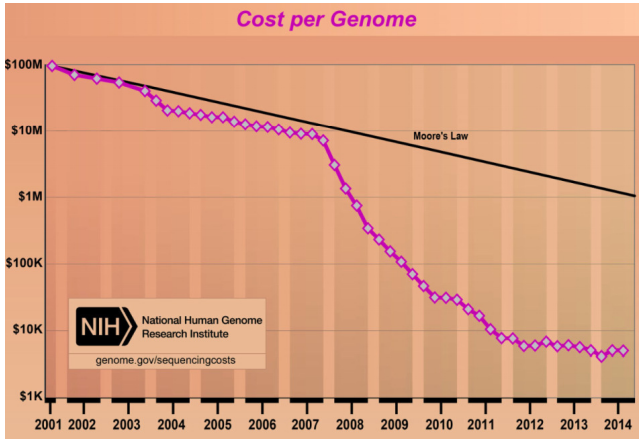




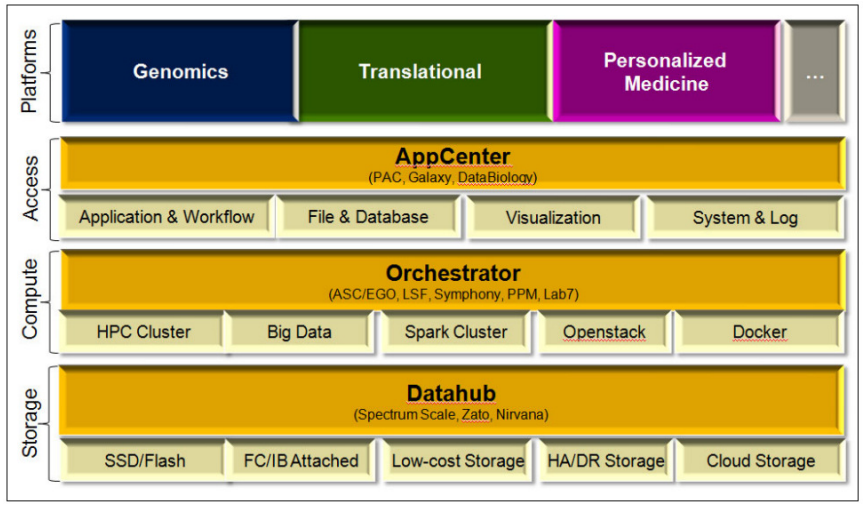
Y_t

Use Case 15: Graph Matching for Genomic Medicine

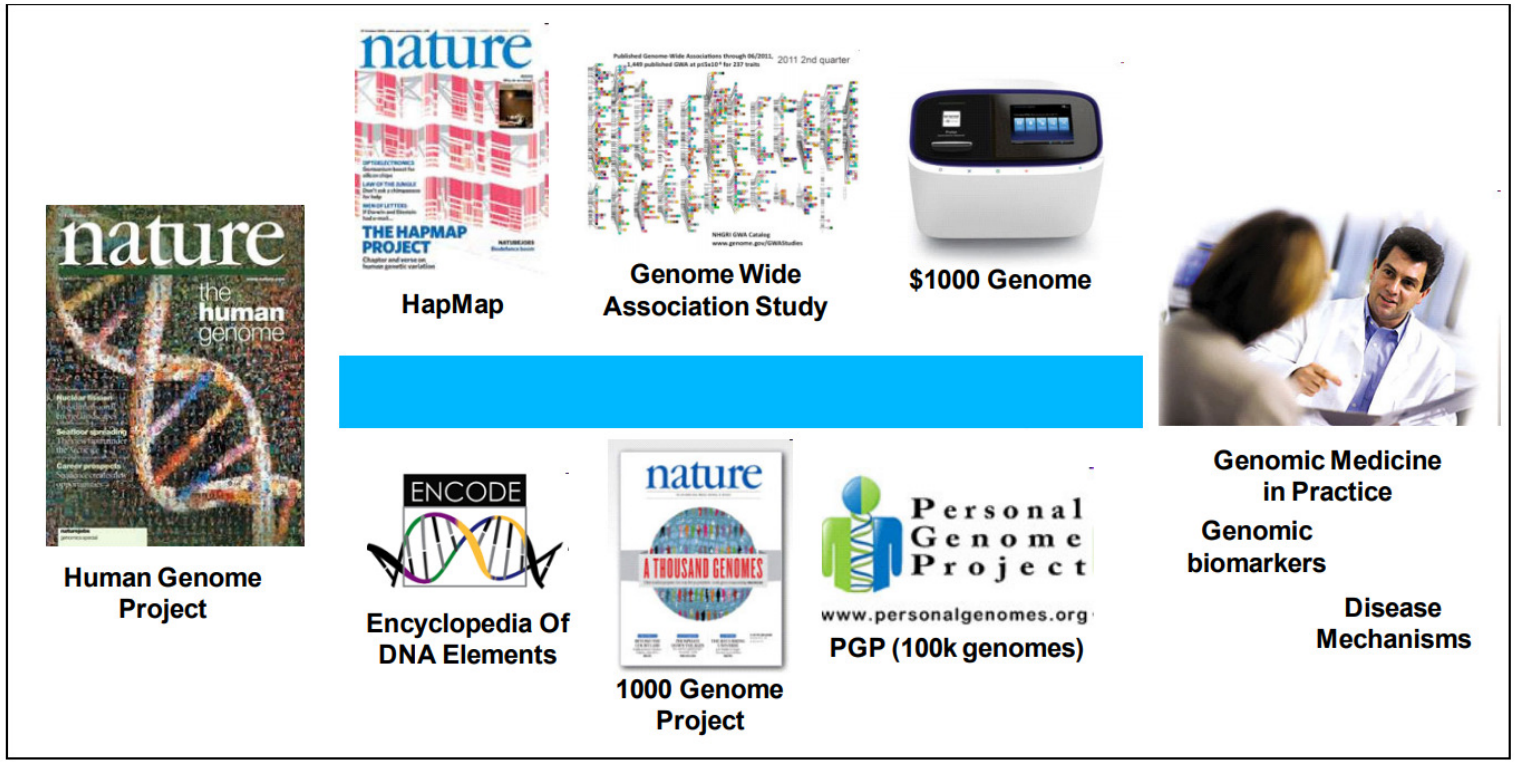
Genomic medicine promises to revolutionize biomedical research and clinical care. By investigating the human genome in the context of biological pathways, drug interaction, and environmental factors, it is now possible for genomic scientists and clinicians to identify individuals at risk of disease, provide early diagnoses based on biomarkers, and recommend effective treatments



Fast drop of DNA sequencing cost



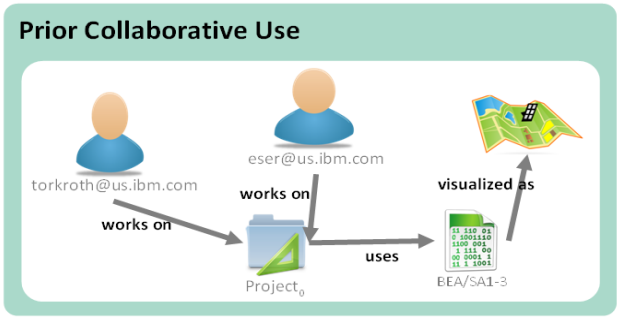
Ongoing discussion:

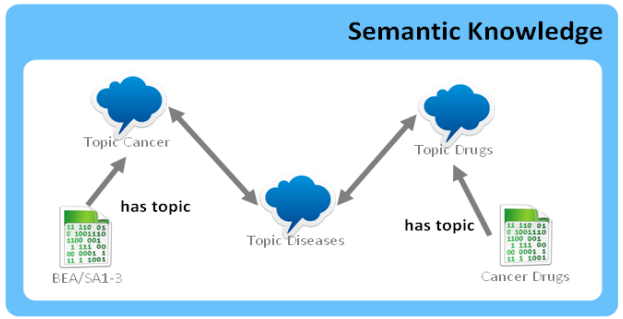


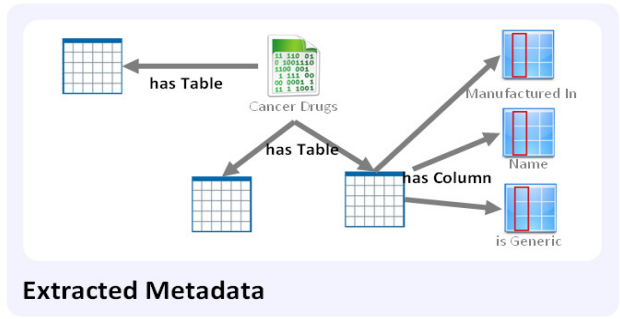
A decade of technological advancement for genomic medicine

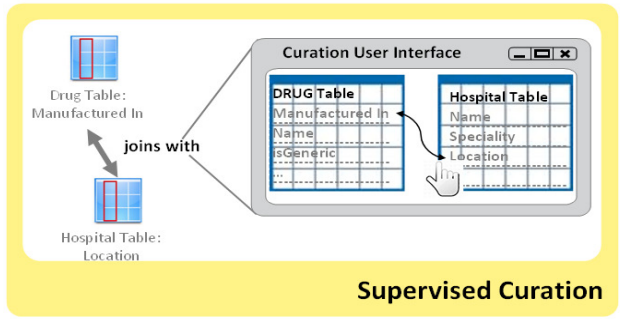
http://www.redbooks.ibm.com/redpapers/pdfs/redp5210.pdf

Use Case 16: Data Curation for Enterprise Data Management

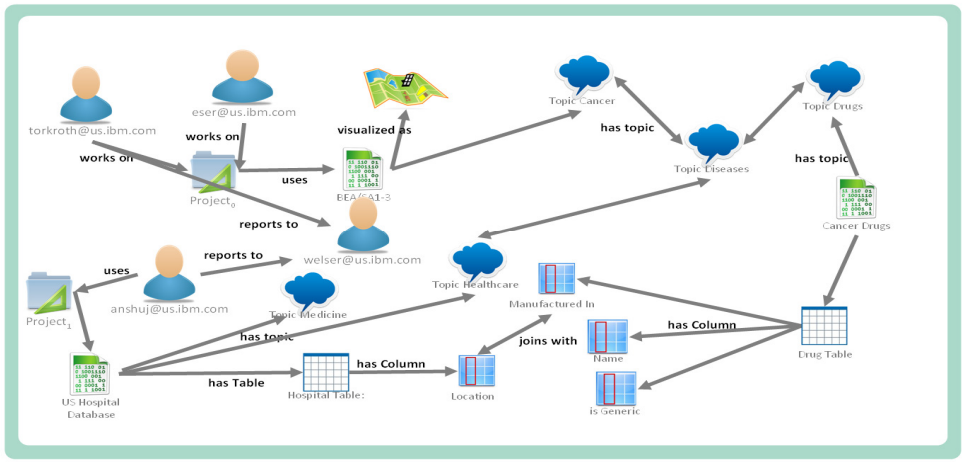




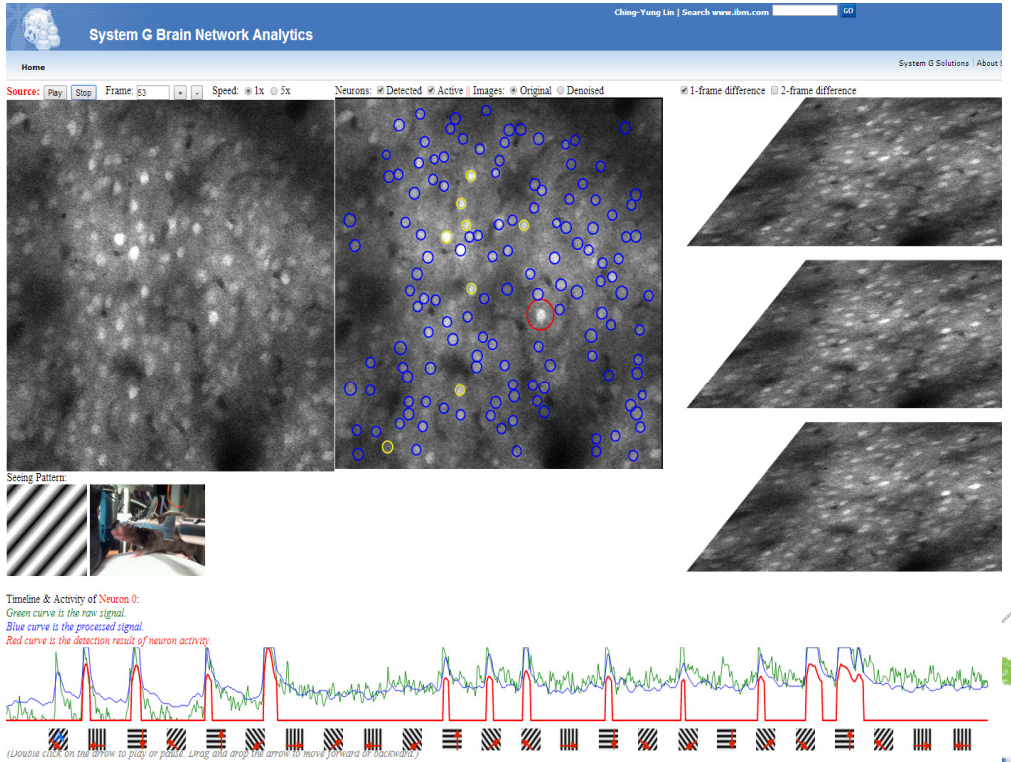


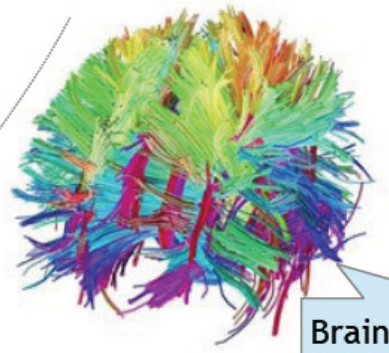


- **Preserving**
  Collecting and taking care of research data
- **Sharing**
  Revealing data's potential across domains
- **Discovering**
  Promoting the re-use and new combinations of data



Use Case 17: Understanding Brain Network

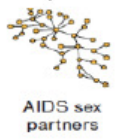




Brain neural network
100 billion neurons
100 trillion connections

Seven bridge problem in 1735

AIDS partner graph

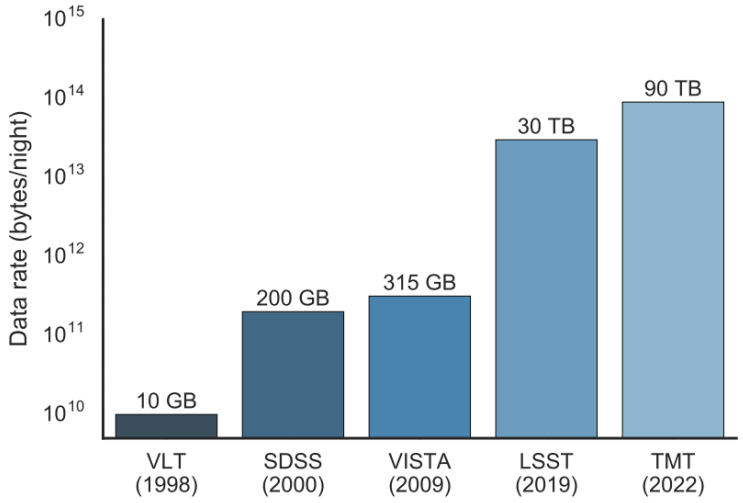


US road map
129,164 nodes
330,870 edges

Social graph, Relation graph
Facebook : 1.23 billion users (2013/12)
WhatsApp : 1 billion users (2015/08)
Our Twitter graph: 120M nodes, 2B edges (2012-14)
Our Bitcoin graph: 70M nodes, 200M edges (2014)

Hamilton graph

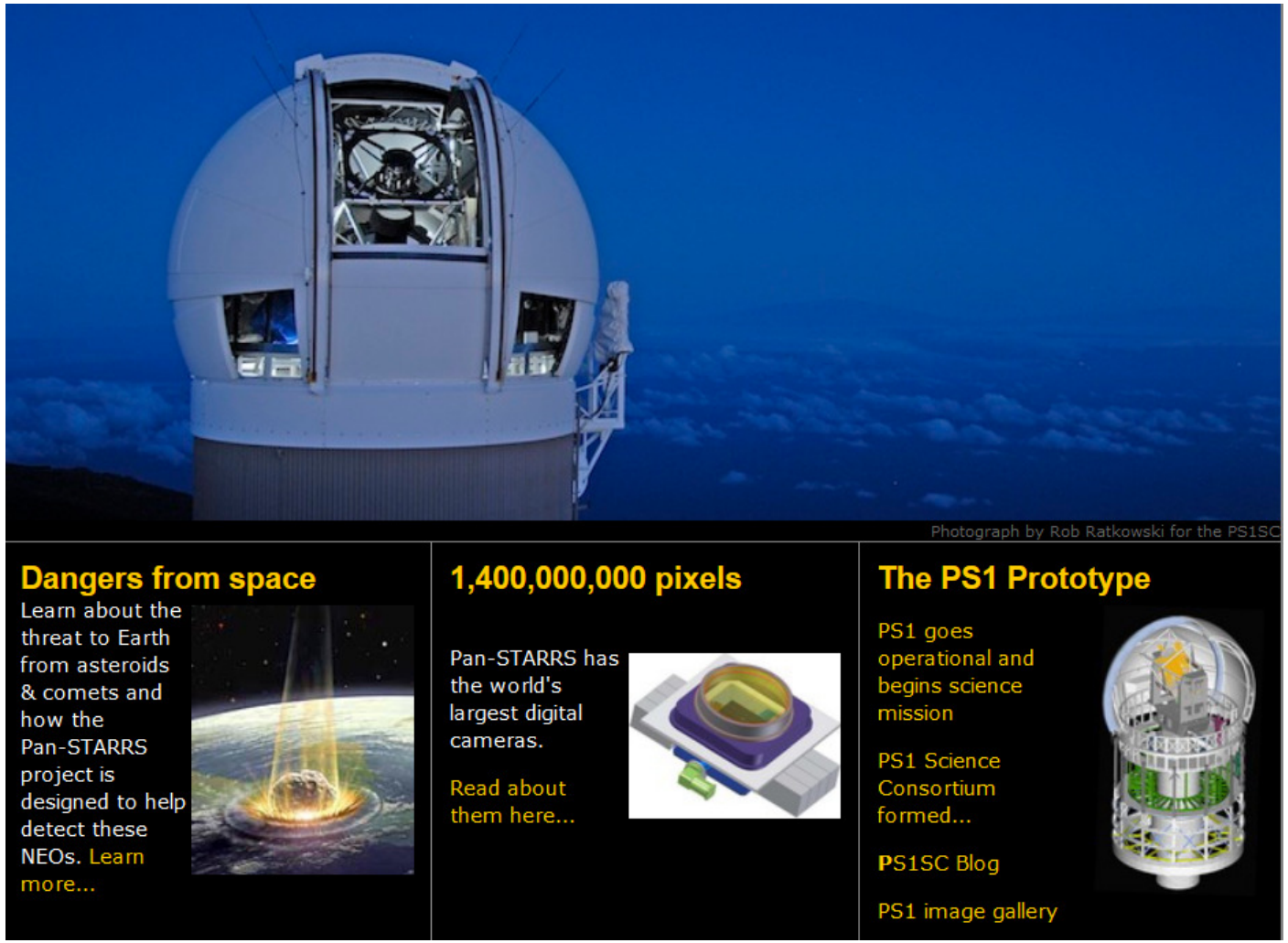
Use Case 18: Planet Security



Increasing data volumes of existing and upcoming telescopes

Kremer et al. Big Universe, Big Data: Machine Learning and Image Analysis for Astronomy," IEEE Intelligent Systems, 2017.

Big Data on Large-Scale Sky Monitoring



Resources

More Big Data Analytics Use Cases:

- https://www.experfy.com/blog/twenty-big-data-use-cases
- https://www.qubole.com/resources/best-use-cases-for-big-data-analytics/
- https://towardsdatascience.com/7-use-cases-for-data-science-and-predictive-analytics-e3616e9331f9
- https://www.searchtechnologies.com/blog/big-data-use-cases-for-business
- https://www.datamation.com/big-data/big-data-use-cases.html
- …

MapReduce

Why?

- Cloud computing (and MapReduce in particular) will be the normal way (if it is not currently) to conduct the analysis of very large amounts of data.

- MapReduce methodology for exploiting parallelism in computing clouds (racks of interconnected processors).

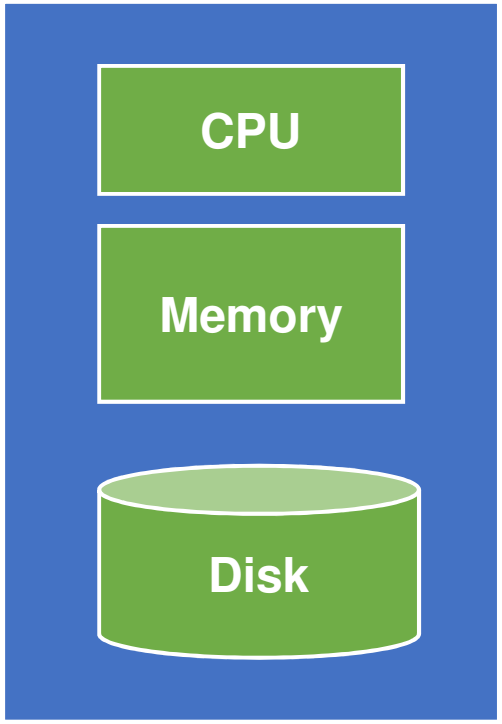
MapReduce

- Problems that are hard (or impossible) to solve at a single site can be solved with the right kind of **parallelization** and **distribution** of the tasks involved

- **Challenges**
  - How to distribute computation
  - Distributed/parallel programming is hard
    - How do you facilitate communication between nodes?
    - How do you scale to more machines?
    - How do you handle machine failures?

- **MapReduce** addresses all of the above
  - Google's computational/data manipulation model
  - Elegant way to work with big data

Single Node Architecture



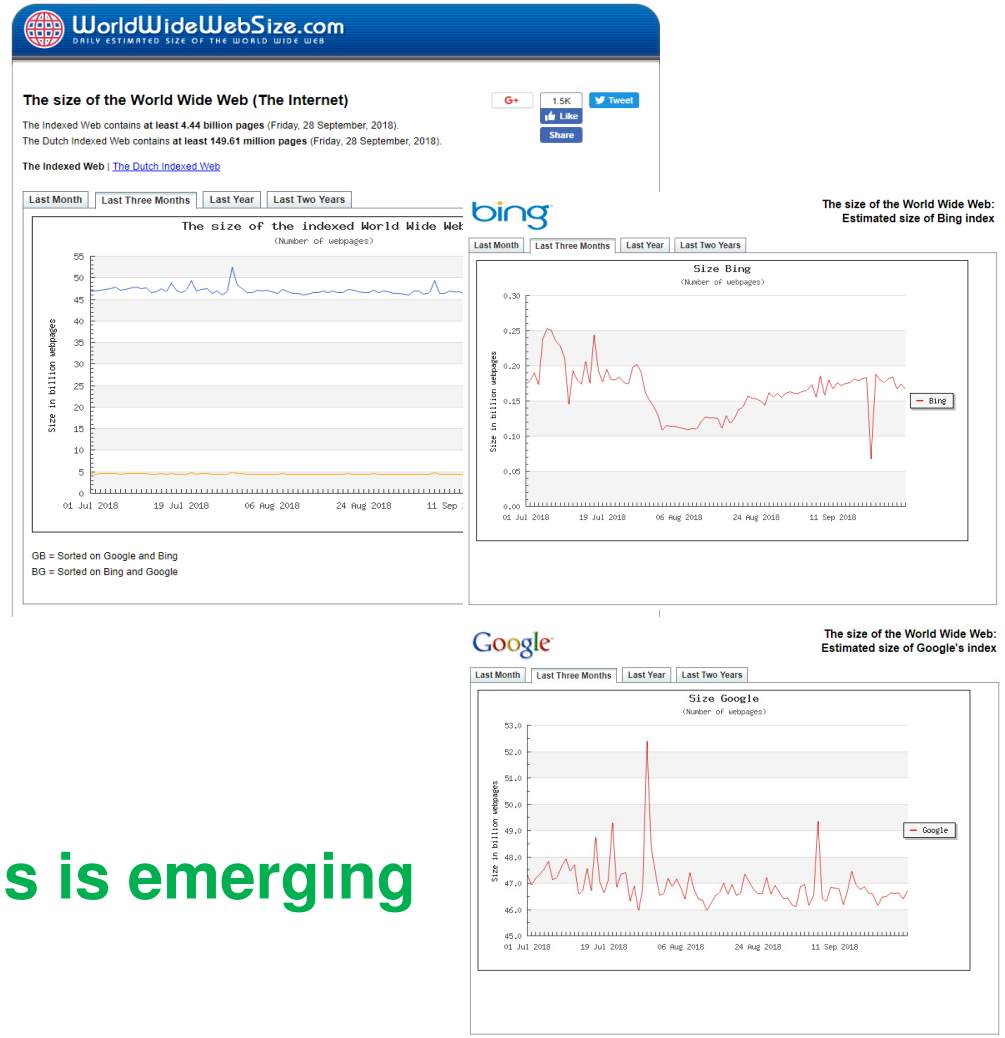
Machine Learning, Statistics

"Classical" Data Mining

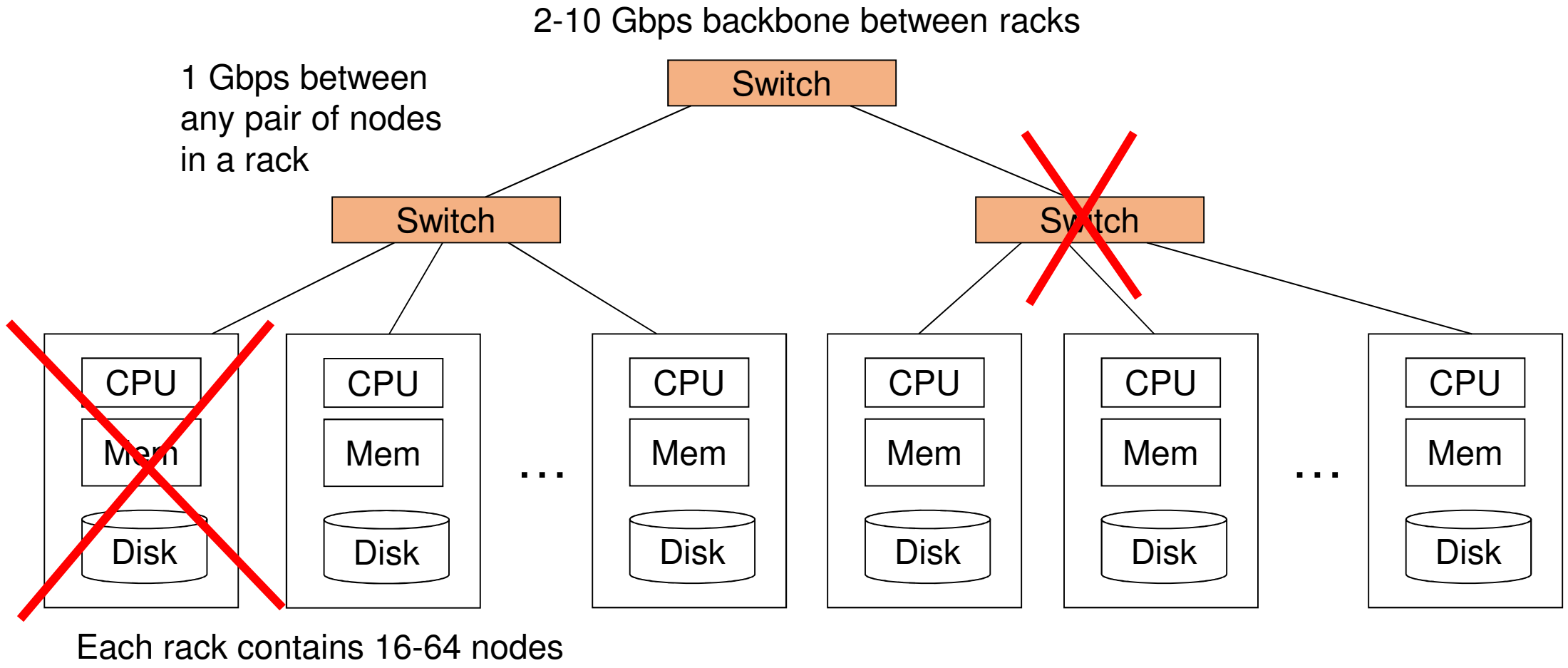
Motivation – Google Example

http://www.worldwidewebsize.com/

- 20+ billion web pages x 20KB = 400+ TB
- 1 computer reads 30-35 MB/sec from disk
  - ~4 months to read the web
- ~1,000 hard drives to store the web
- Takes even more to **do** something useful
- **A standard architecture for such problems is emerging**
  - Cluster of commodity Linux nodes
  - Commodity network (Ethernet) to connect them



Cluster Architecture



In 2011 it was guestimated that Google had 1M machines, http://bit.ly/Shh0RO


The cluster looks like...
http://www.mmds.org

The cluster looks like...

http://aggregate.org/press001122.html

Large-scale Computing

- Large-scale computing for data mining problems on **commodity hardware**

- **Challenges**
  - How do you **distribute** computation?
  - How can we **make it easy** to write distributed programs?
  - How do we handle **machines fail**?
    - One server may stay up 3 years (1,000 days)
    - If you have 1,000 servers, expect to loose 1/day
    - People estimated Google had ~1M machines in 2011
      - 1,000 machines fail every day!

Idea and Solution

- **Issue**
  - Copying data over a network takes time

- **Idea**
  - Bring computation close to data
  - Store files multiple times for reliability

- **Map-reduce** addresses these problems
  - Google's computational/data manipulation model
  - Elegant way to work with big data
  - Storage Infrastructure (File System): Google: GFS; Hadoop: HDFS
  - Programming model: MapReduce

Storage Infrastructure

- **Problem**
  - If nodes fail, how to store data persistently?

- **Answer**
  - Distributed File System
    - Provides global file namespace
    - Google GFS, Hadoop HDFS

- **Typical usage pattern**
  - Huge files (100s of GB to TB)
  - Data is rarely updated in place
  - Reads and appends are common

Distributed File Systems

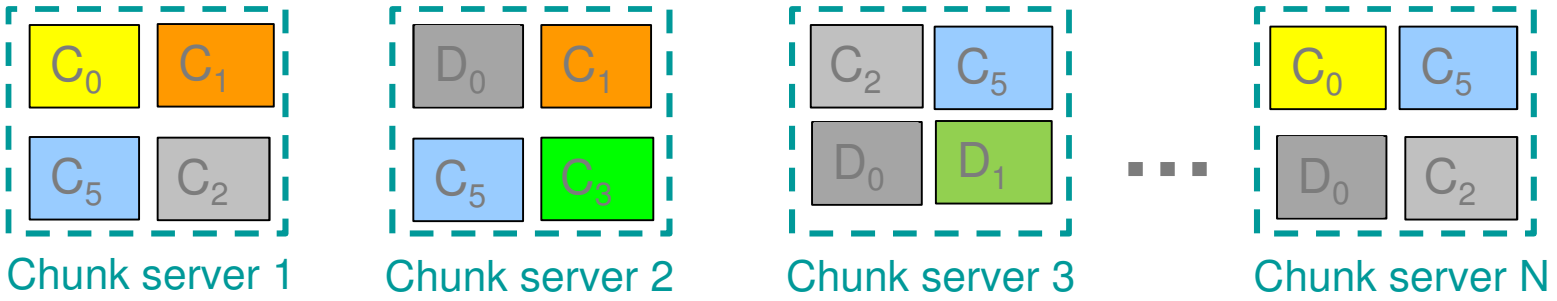
- **Chunk servers**
  - File is split into contiguous chunks
  - Typically each chunk is 16-64MB
  - Each chunk replicated (usually 2x or 3x)
  - Try to keep replicas in different racks

- **Master node**
  - a.k.a. Name Node in Hadoop's HDFS
  - Stores metadata about where files are stored
  - Might be replicated

- **Client libraries for file access**
  - Talks to master to find chunk servers
  - Connects directly to chunk servers to access data

Distributed File Systems

- **Reliable distributed file system**
- Data kept in “chunks” spread across machines
- Each chunk **replicated** on different machines
  - Seamlessly recovery from disk or machine failure



Bring computation directly to the data

Chunk servers also server as compute nodes

Programming Model – MapReduce

Warm-up task

- **Given**
  - A huge text document

- **Task**
  - Count the number of times each distinct word appears in the document

- **Example application**
  - Analyze web server logs to find popular URLs

Task: Word Count

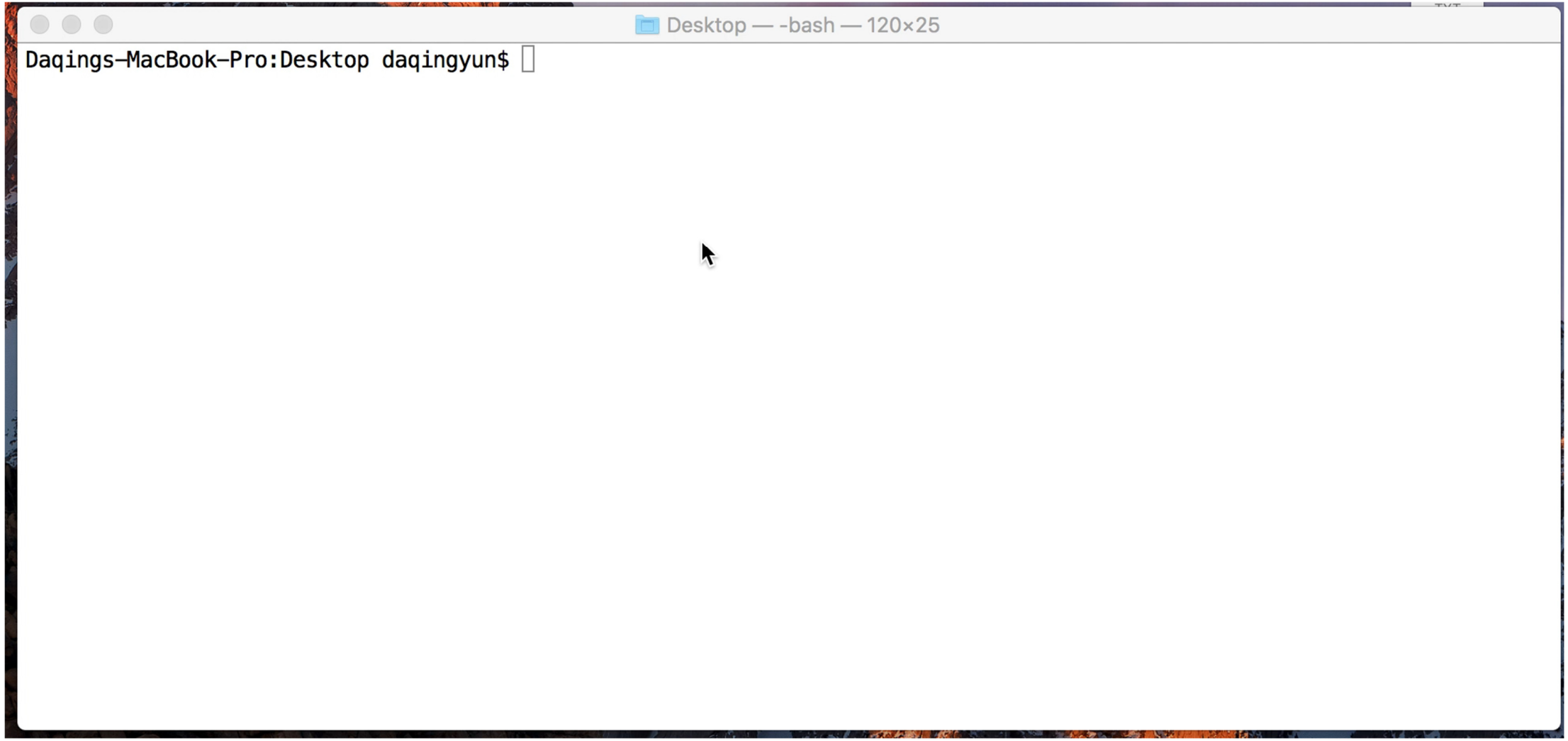
- File too large for memory, but all <word, count> pairs fit in memory

- Count occurrences of words
  - **words(doc.txt) | sort | uniq -c**
  - **words(doc.txt)** takes a file (doc.txt) as input and outputs the words in it, one per a line

- This captures the essence of **MapReduce**
  - Great thing is that it is naturally parallelizable

Let us try this on a single computer

Task: Word Count



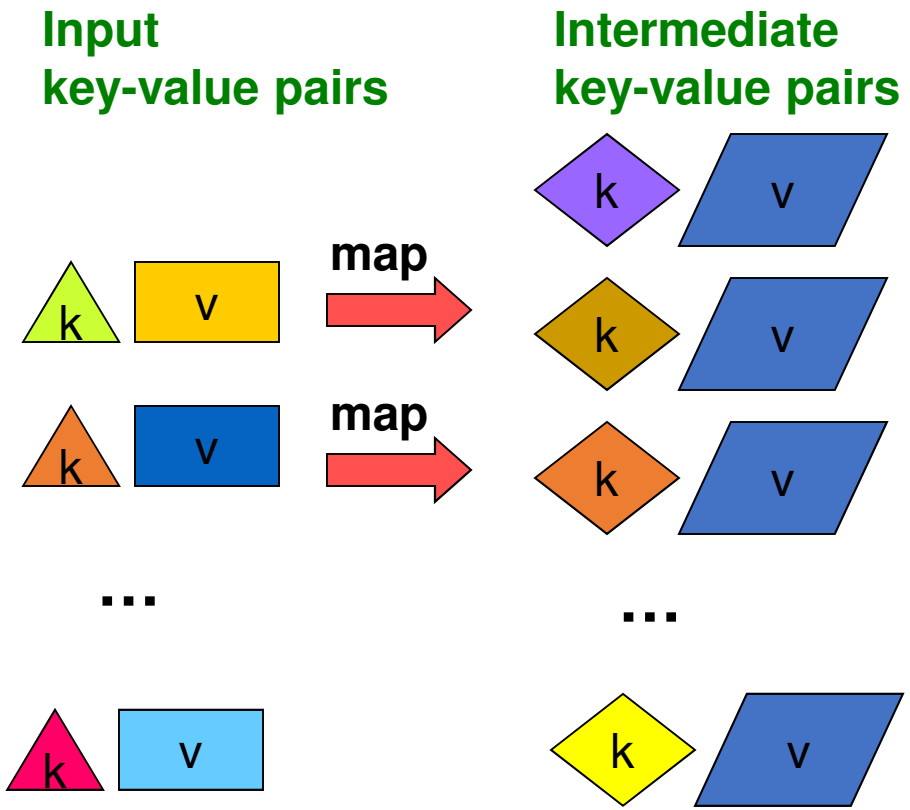
MapReduce – Overview

```
words(doc.txt) | sort | uniq -c
```

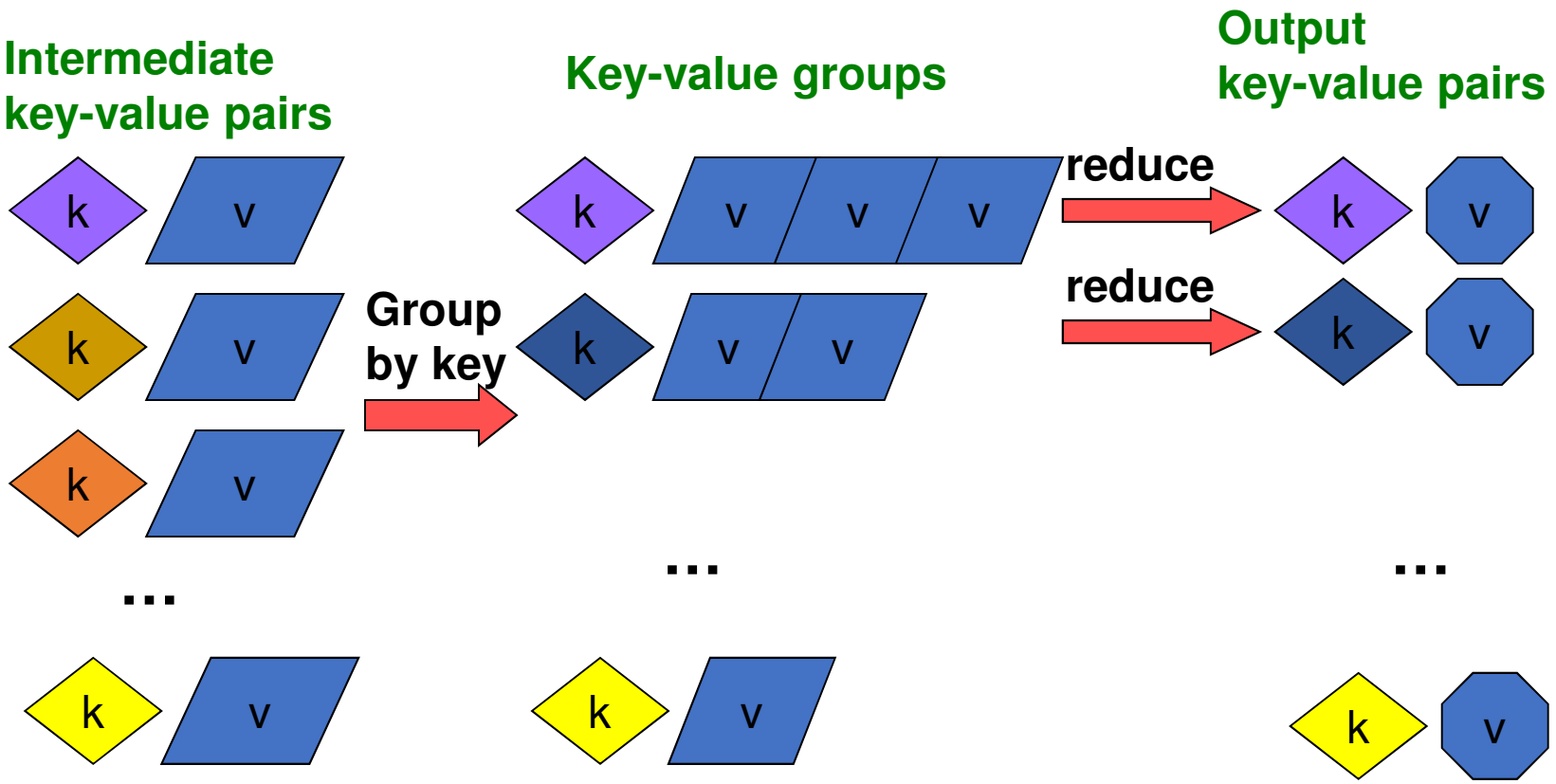
- **Sequentially read a lot of data**
- **Map**
  - Scan input file (read sequentially)
  - **Extract** something you care about
- **Group by key**
  - Sort and Shuffle
- **Reduce**
  - Aggregate, summarize, filter, or transform
- Write the output

Outline stays the same, **Map** and **Reduce** change to fit the problem

MapReduce – The Map Step



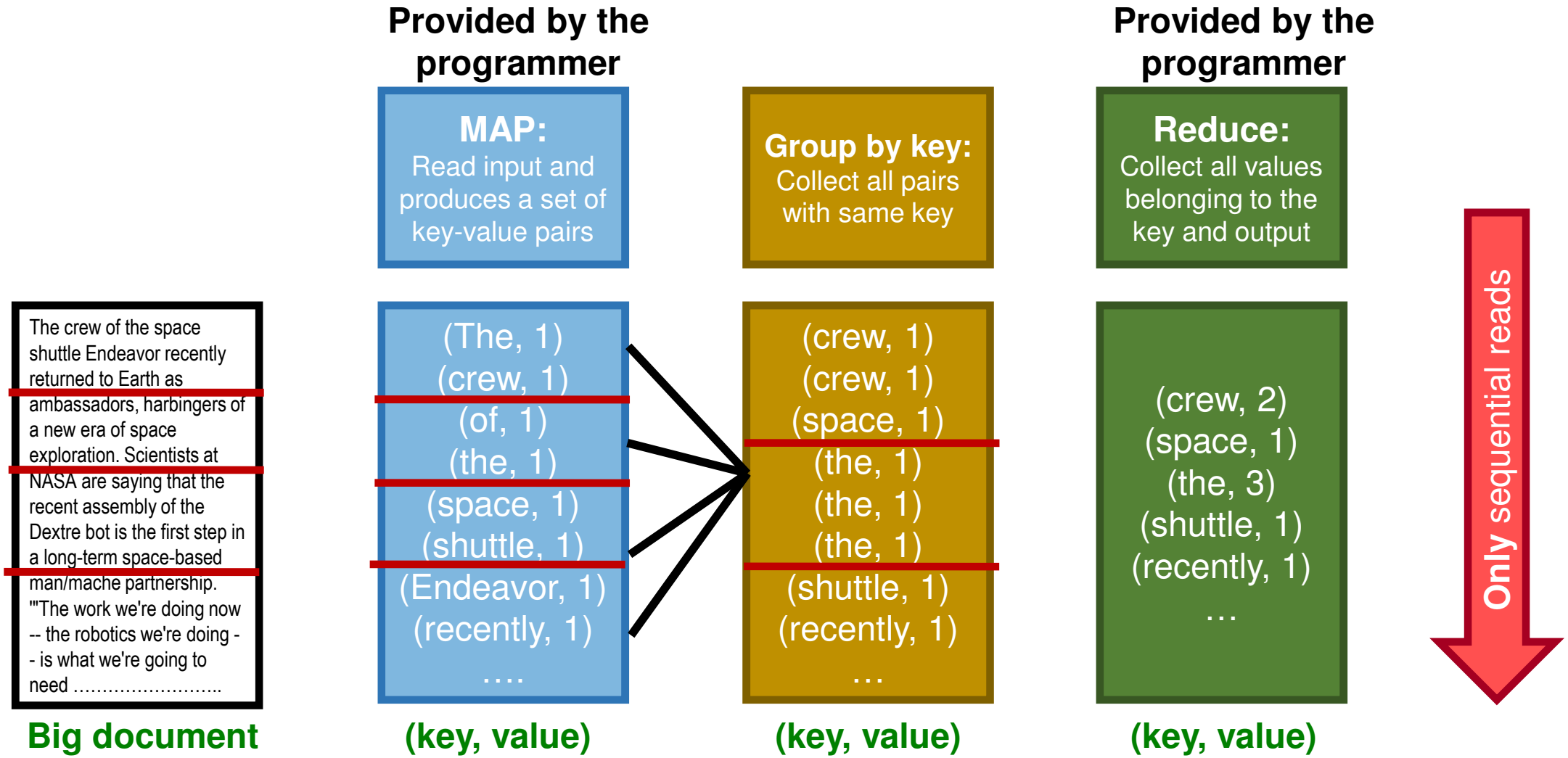
MapReduce – The Reduce Step



More Specifically

- **Input**
  - A set of key-value pairs
- Programmer specifies two methods:
  - **Map(k, v)** → <k', v'>*
    - Takes **a key-value pair** and outputs **a set of key-value pairs**
      - E.g., key is the filename, value is a single line in the file
    - There is one Map call for every (k, v) pair
  - **Reduce(k', <v'>*)** → <k', v''>*
    - All values **v'** with the same **k'** are reduced together and processed in v' order
    - There is one Reduce function call per **unique** key **k'**

MapReduce – Word Counting



MapReduce – Word Counting

```
map(key, value):
// Takes a key-value pair and outputs a set of key-value pairs
// key: document name; value: text of the document
  for each word w in value:
       emit(w, 1)
```

```
reduce(key, values):
// All values with the same key are reduced and processed together
// key: a word; value: an iterator over counts
       result = 0
       for each count v in values:
             result += v
       emit(key, result)
```

MapReduce – Environments

MapReduce environment takes care of:

- Partitioning the input data
- Scheduling the program's execution process across a set of machines
- Performing the **group by key** step
- Handling machine failures -- detect failure via periodic heartbeats
- Managing required inter-machine communication

MapReduce – Environments

Need to handle more data? Just add more Mappers and Reducers:

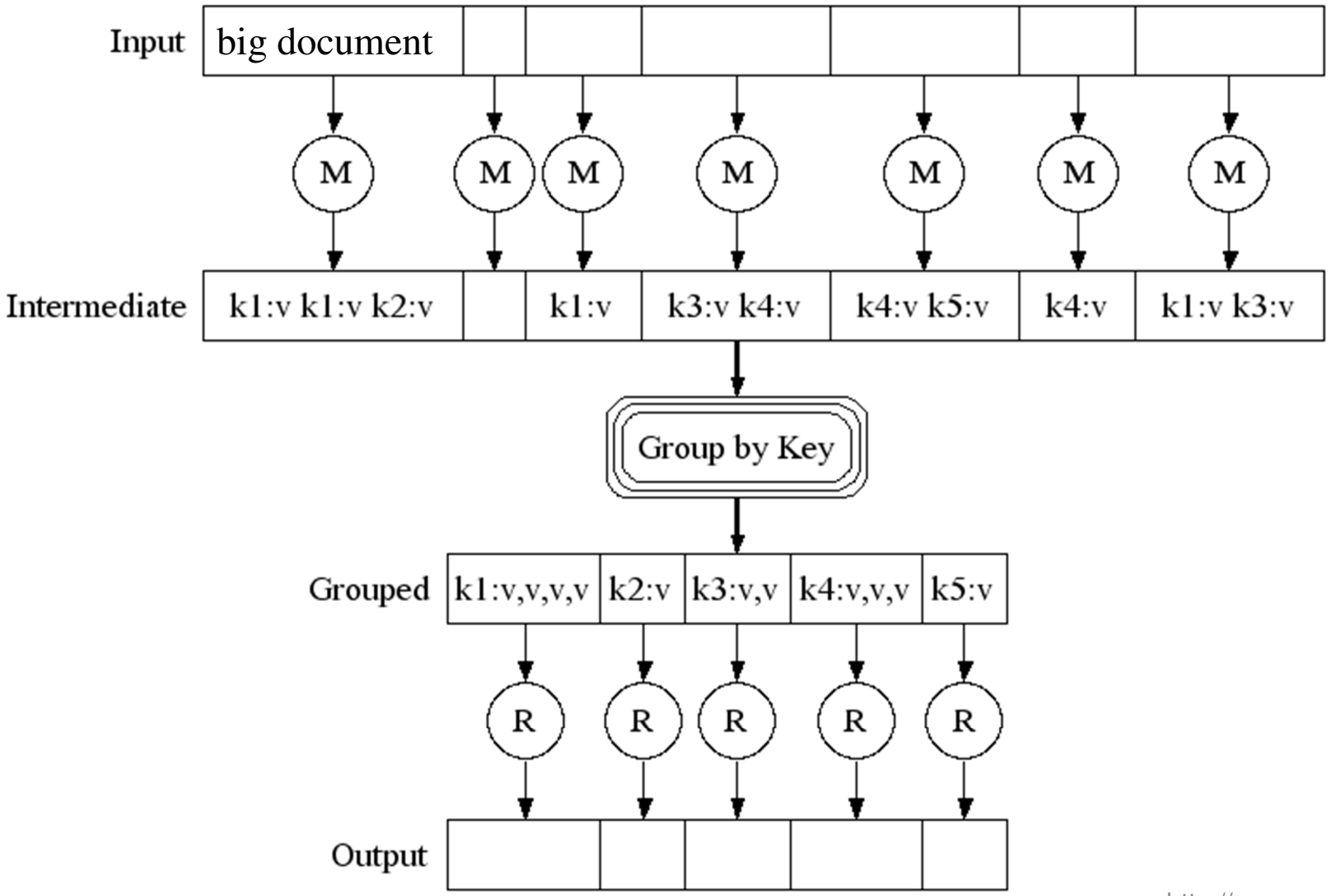
- No need to handle **multithreaded** code ;-)
- Mappers and Reducers are typically single threaded and deterministic
  - Allows failed jobs to restart
- Mappers and Reducers run **entirely independent**
  - In Hadoop, they run on different JVMs

MapReduce – A Diagram

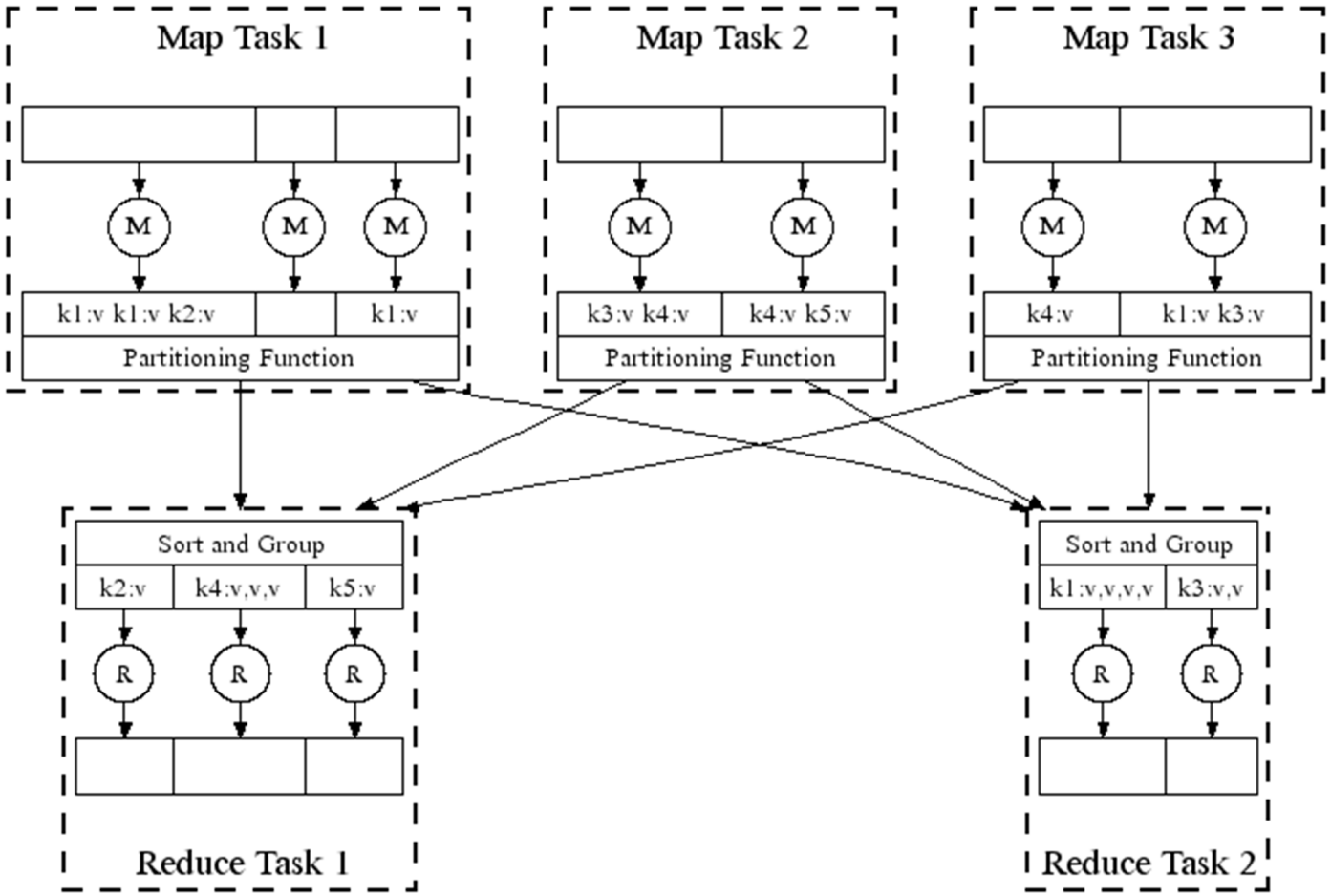
MAP:
Read input and produces a set of key-value pairs

Group by key:
Collect all pairs with same key

Reduce:
Collect all values belonging to the key and output



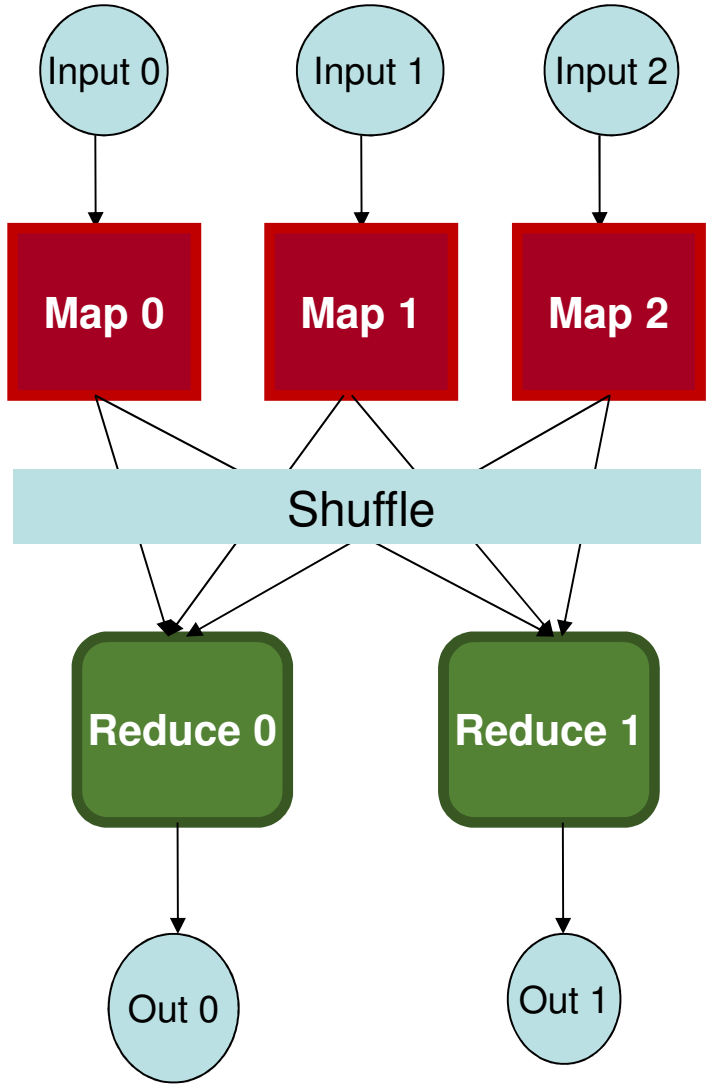
MapReduce – In Parallel



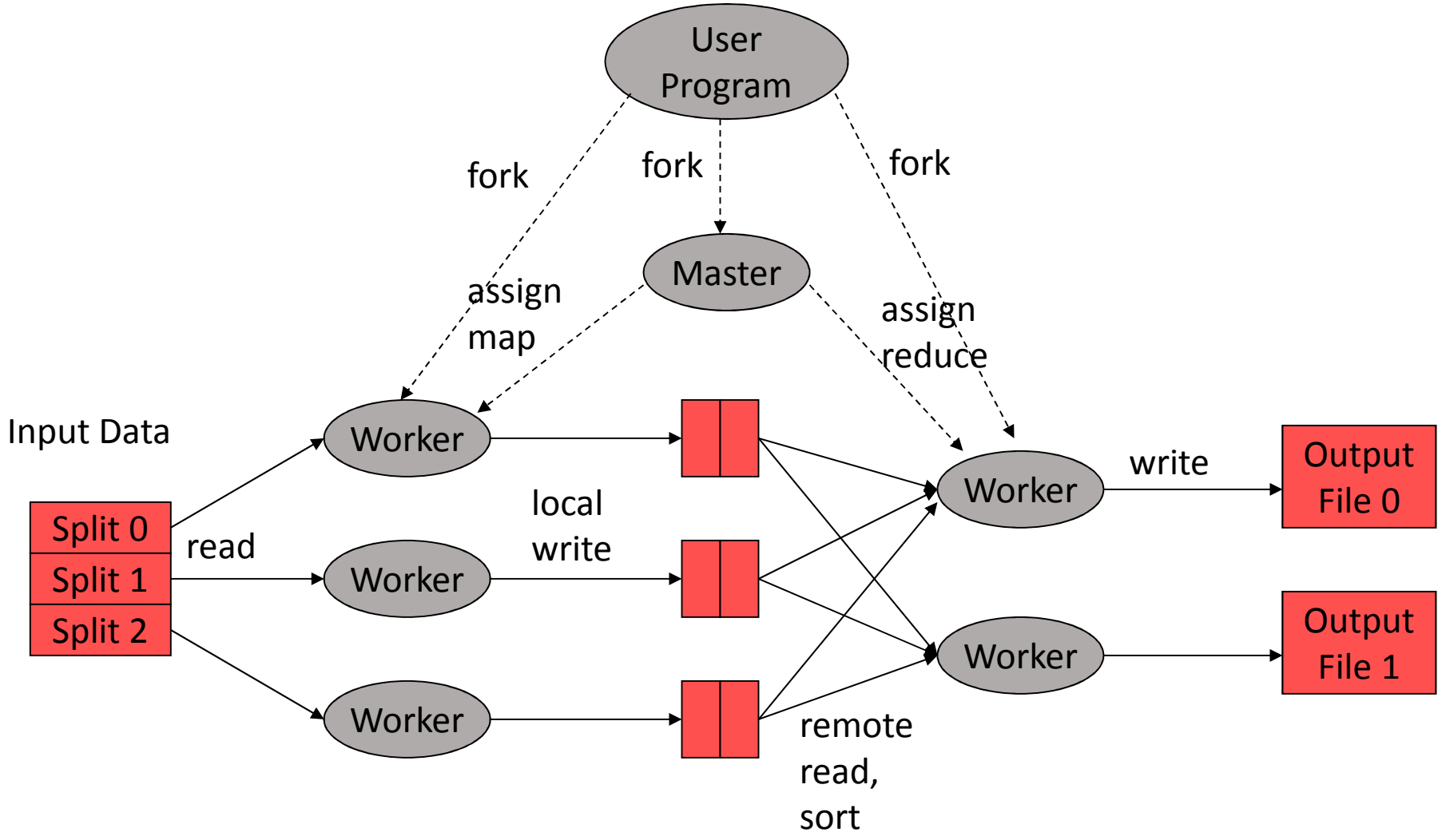
All phases are distributed with many tasks doing the work

MapReduce – In Parallel

- **Programmer specifies:**
  - Map and Reduce and input files
- **Workflow:**
  - Read inputs as a set of key-value-pairs
  - **Map** transforms input kv-pairs into a new set of k'v'-pairs
  - Sorts & Shuffles the k'v'-pairs to output nodes
  - All k'v'-pairs with a given k' are sent to the same **reduce**
  - **Reduce** processes all k'v'-pairs grouped by key into new k''v''-pairs
  - Write the resulting pairs to files
- **All phases are distributed with many tasks doing the work**



Distributed Execution Overview



Data Flow

- **Input and final output** are stored on a **distributed file system (FS)**
  - Scheduler tries to schedule map tasks “close” to physical storage location of input data

- **Intermediate results** are stored on **local FS** of Map and Reduce workers

- Output is often input to **another** MapReduce task

Coordinator – Master

- **Master node takes care of coordination:**
  - **Task status**: (idle, in-progress, completed)
  - **Idle tasks** get scheduled as workers become available
  - When a map task completes, it sends the master the location and sizes of its R intermediate files, one for each reducer
  - Master pushes this info to reducers

- Master pings workers periodically to detect failures

Dealing with Failures

- **Map worker failure**
  - Map tasks completed or in-progress at worker are reset to idle
  - Reduce workers are notified when task is rescheduled on another worker
- **Reduce worker failure**
  - Only in-progress tasks are reset to idle
  - Reduce task is restarted
- **Master failure**
  - MapReduce task is aborted and client is notified; Resume from execution log

How many Map and Reduce jobs?

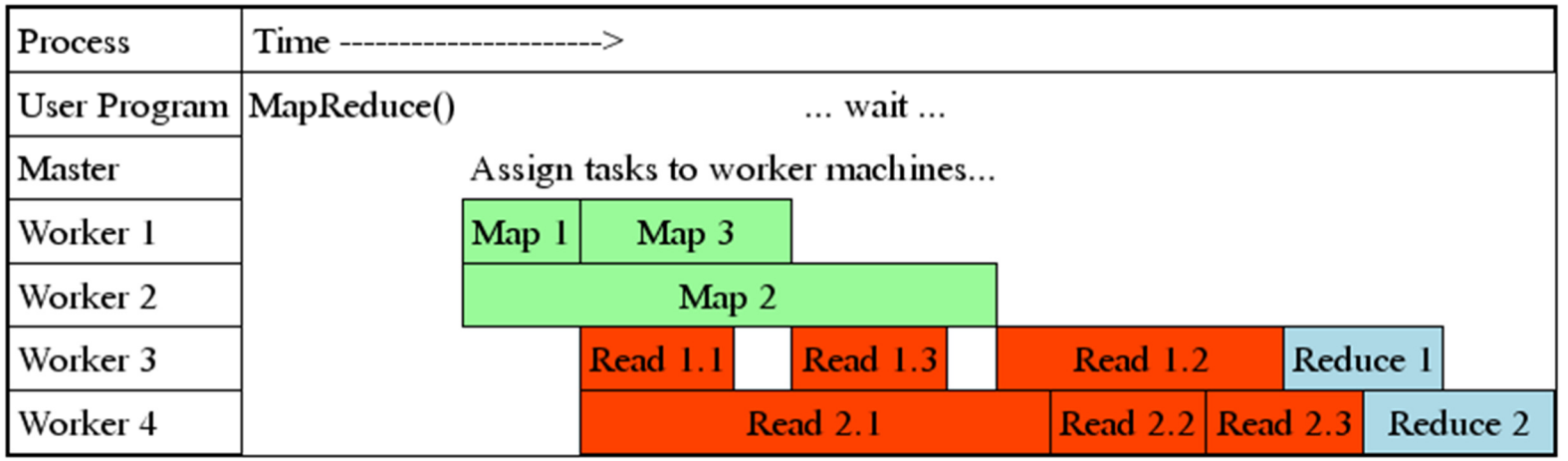
- M map tasks, R reduce tasks

- **Rule of a thumb:**
  - Make M and R much larger than the number of nodes in the cluster
  - One DFS chunk per map is common
  - Improves dynamic load balancing and speeds up recovery from worker failures

- **Usually R is smaller than M**
  - Because output is spread across R files

Task Granularity & Pipelining

- **Fine granularity tasks:** map tasks >> machines
  - Minimizes time for fault recovery
  - Can do pipeline shuffling with map execution
  - Better dynamic load balancing



Refinements: Backup Tasks

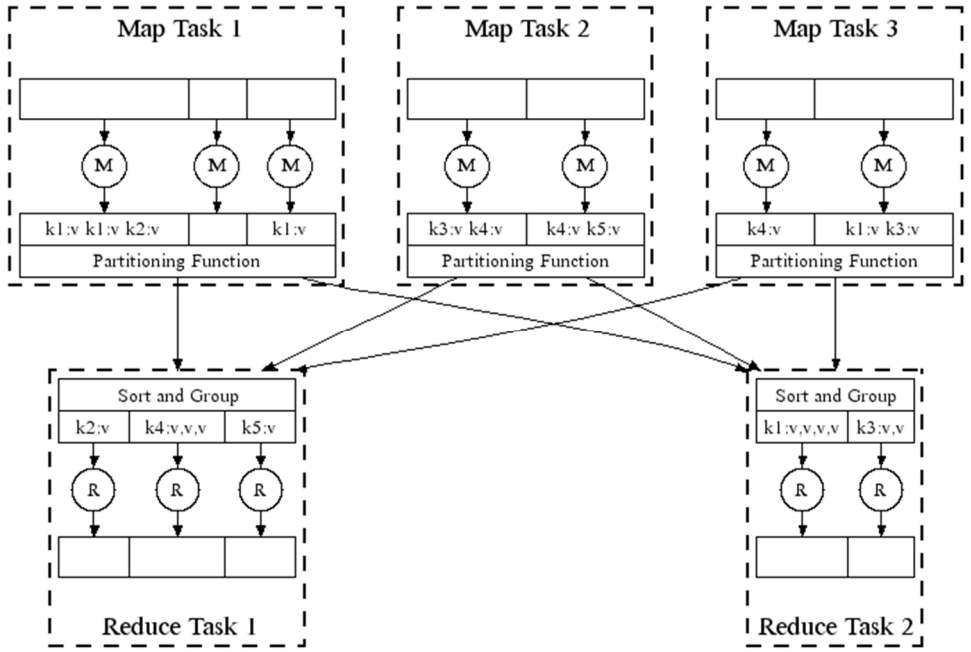
- **Problem**
  - Slow workers significantly lengthen the job completion time:
    - Other jobs on the machine
    - Bad disks
    - Weird things

- **Solution**
  - Near end of phase, spawn backup copies of tasks
    - Whichever one finishes first “wins”

- **Effect**
  - Dramatically shortens job completion time

Refinements: Combiners

- Often a Map task will produce many pairs of the form (k,v_1), (k,v_2), … for the same key k
  - E.g., popular words in the word count example

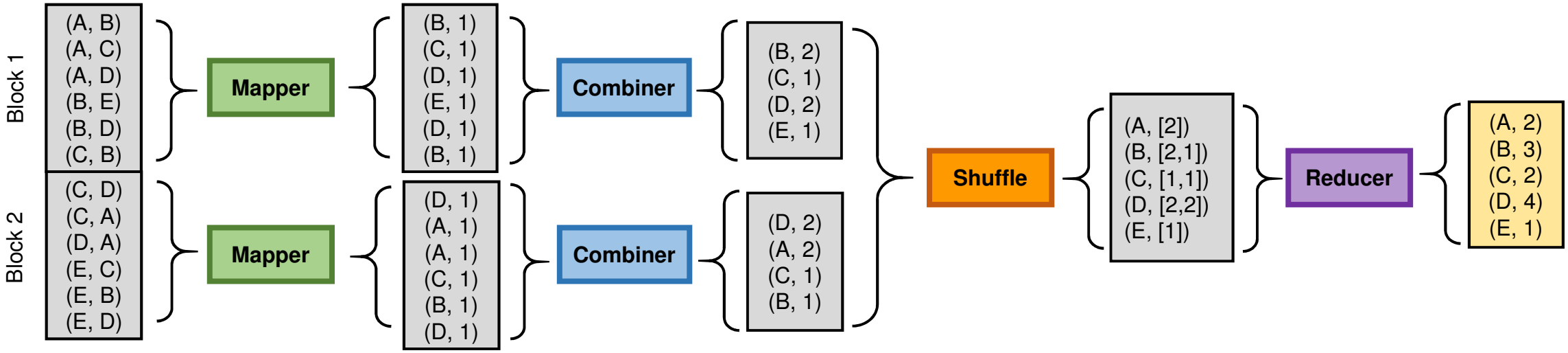


- **Can save network time by pre-aggregating values in the mapper:**
  - combine(k, list(v_1)) → v_2
  - Combiner is usually same as the reduce function

- Works only if reduce function is **commutative (a*b=b*a)** and **associative (a+b)+c=a+(b+c)**

Refinements: Combiners

- **Back to our word counting example:**
  - Combiner combines the values of all keys of a single mapper (single machine):



  - Much less data needs to be copied and shuffled!

Refinements: Partition Function

- **Want to control how keys get partitioned**
  - Inputs to map tasks are created by contiguous splits of input file
  - Reduce needs to ensure that records with the same intermediate key end up at the same worker

- **System uses a default partition function:**
  - **hash(key) mod R**

- **Sometimes useful to override the hash function:**
  - E.g., **hash(hostname(URL)) mod R** ensures URLs from a host end up in the same output file

Problems Suited for MapReduce
Example1: Host Size

- **Suppose we have a large web corpus**
- Look at the metadata file
  - Lines of the form: **(URL, size, date, …)**
- **For each host, find the total number of bytes**
  - That is, the **sum of the page sizes** for all URLs from that particular host
  - **Map:** for each record, output <hostname(URL), size>
  - **Reduce:** sum the sizes for each host

- **Other examples:**
  - Link analysis and graph processing
  - Machine Learning algorithms

Problems Suited for MapReduce
Example 2: Language Model

- **Statistical machine translation:**
  - Need to count number of times every **5-word sequence** occurs in a large corpus of documents

- **Very easy with MapReduce:**
  - **Map:** extract <5-word sequence, count> from document
  - **Reduce:** combine the counts

Problems Suited for MapReduce
Example 3: Distributed Grep

- Find all occurrences of **a given pattern** in a very large data set
  - Distributed grep is used to search for a given pattern in a large number of files.
  - For example, a web administrator can use distributed grep to search web server logs in order to find the top requested pages that match a given pattern

- MapReduce
  - **Map:** take input as (input file, line) and generate one of the following outputs
    - An empty list <>, if there is no match found
    - A key value pair <line, 1> if a match is found
  - **Reduce:** take input as <line, (1, 1, 1, ...)> and generate output as <line, n> where “n” is the number of 1’s in the list

Problems Suited for MapReduce
Example 4: Graph Reversal

- Given a directed (e.g., a web-link) graph as an adjacency list:
  - src-1: dest-1-1, dest-1-2, …
  - src-2: dest-2-1, dest-2-2, …

- Construct the graph in which all the links are reversed
  - **Map:** outputs <*destination*, *source*> pairs for each edge to a *destination* node found in a source node named *source*
  - **Reduce:** concatenates the list of all source nodes **associated** with a given destination node and emits the pair <*destination*, *list of source nodes*>

Problems Suited for MapReduce
Example 5: Frequent Pairs

- Given a large set of market baskets, find all frequent pairs
  - Basket: a set of items someone bought together at one time
    - E.g., {apple, milk, coffee, orange} -- remember definitions from Association Rules
  - If frequency(<a, b>) is larger than a threshold, <a, b> is a frequent pair, then frequency(a) and frequency(b) both should be larger than the threshold
  - Finding the frequent individual items first
  - Two **Map**-**Reduce** passes
    - First: to divide the entire file of baskets into segments small enough that all frequent items for the segment can be found in main memory, i.e., look for frequent items -- candidate items are those found frequent in at least one segment
    - Second: second pass allows us to count all the candidates and find the exact collection of frequent item -- look for frequent pairs

Problems Suited for MapReduce

Example 6: Join By MapReduce

- **Compute the natural join *R(A,B)* ⋈ *S(B,C)***
- *R* and *S* are each stored in files
- Tuples are pairs *(a,b)* or *(b,c)*

A	B
a_1	b_1
a_2	b_1
a_3	b_2
a_4	b_3

R

⋈

B	C
b_2	c_1
b_2	c_2
b_3	c_3

S

=

A	C
a_3	c_1
a_3	c_2
a_4	c_3

Problems Suited for MapReduce
Example 6: Join By MapReduce

- **Use a hash function *h* from B-values to *1...k***
- **A Map process turns:**
  - Each input tuple *R(a,b)* into key-value pair *(b,(a,R))*
  - Each input tuple *S(b,c)* into *(b,(c,S))*
- **Map processes** send each key-value pair with key *b* to Reduce process *h(b)*
  - Hadoop does this automatically; just tell it what *k* is.
- Each **Reduce process** matches all the pairs *(b,(a,R))* with all *(b,(c,S))* and outputs *(a,b,c)*.

Problems Not Suited for MapReduce

- If the computation of a value depends on previously computed values, then MapReduce cannot be used.
  - Fibonacci series where each value is summation of the previous two values. i.e., f(k+2) = f(k+1) + f(k)
  - If the data set is small enough to be computed on a single machine, then it is better to do it as a single reduce(map(data)) operation rather than going through the entire map reduce process
- Following multiple pointer hops
- Iterative algorithms
- Algorithms with global state

Cost Measures for Algorithms

- **In MapReduce we quantify the cost of an algorithm using:**
  1. **Communication cost** = total I/O of all processes
  2. **Elapsed communication cost** = max of I/O along any path
  3. (**Elapsed**) **computation cost** analogous, but count only running time of processes

- Note that here the big-O notation is not the most useful (adding more machines is always an option)

Example: Cost Measures

- **For a map-reduce algorithm:**
  - **Communication cost =** input file size + 2 × (sum of the sizes of all files passed from Map processes to Reduce processes) + the sum of the output sizes of the Reduce processes.
  - **Elapsed communication cost** is the sum of the largest input + output for any map process, plus the same for any reduce process

What Cost Measures Mean

- Either the I/O (communication) or processing (computation) cost dominates
  - Ignore one or the other

- Total cost tells what you pay in rent from your friendly neighborhood cloud

- Elapsed cost is **wall-clock time** using parallelism

Cost of Map-Reduce Join

- **Total communication cost** = $O(|R|+|S|+|R \bowtie S|)$
- **Elapsed communication cost** = $O(s)$
  - We're going to pick ***k*** and the number of Map processes so that the I/O limit ***s*** is respected
  - We put a limit ***s*** on the amount of input or output that any one process can have ***s* could be:**
    - What fits in main memory
    - What fits on local disk
- With proper indexes, computation cost is linear in the input + output size
  - So computation cost is like communication cost

MapReduce – Implementations

- Google
  - Not available outside Google

- **Hadoop**
  - An open-source implementation in Java
  - Uses HDFS for stable storage
  - Download: http://hadoop.apache.org/

- Aster Data
  - Cluster-optimized SQL Database that also implements MapReduce

Cloud Computing

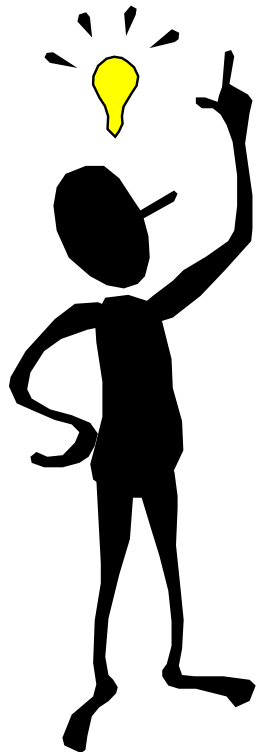
- Ability to rent computing by the hour
  - Additional services e.g., persistent storage
- Amazon's "Elastic Compute Cloud" (EC2)
- Aster Data and Hadoop can both be run on EC2
- **Use the VMs you created in AWS**

Further Reading

- J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51, 1, pp. 107-113.
  - https://goo.gl/hglEvH

- S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google File System. In *Proc. of the 19th ACM Symp. on Operating Systems Principles*, New York, NY, pp. 29-43.
  - https://goo.gl/dcluda

- J. Leskovec, A. Rajaraman, and J. Ullman. Chapter 2 MapReduce and New Software Stack, *Mining of Massive Datasets*.



Thanks ! ☺

Questions ?