UVA HPC & BIG DATA COURSE

Introduction to Big Data

Adam Belloum

Content

- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

Jim Gray Vision in 2007

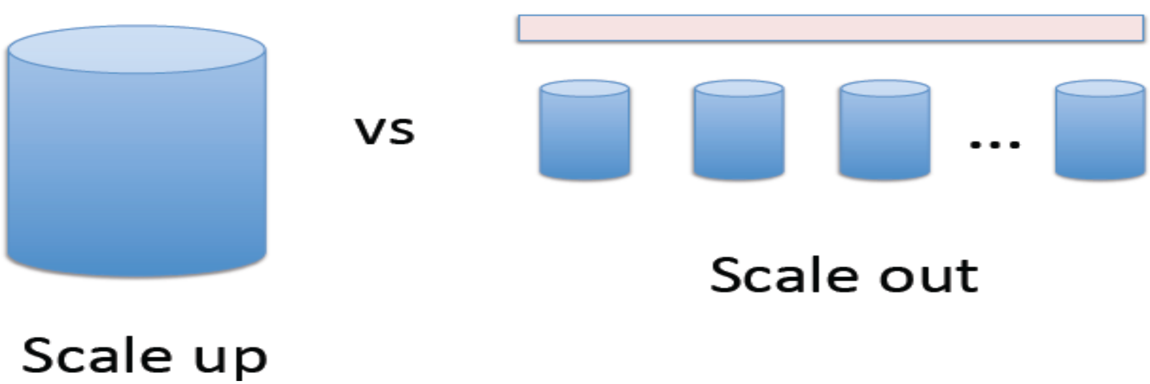
- “ We have to **do better at producing tools** to support the whole research cycle—from data capture and data curation to data analysis and data visualization. Today, the **tools** for capturing data both at the mega-scale and at the milli-scale are just **dreadful**. After you have captured the data, you need to curate it before you can start doing any kind of data analysis, and we lack good tools for both data curation and data analysis.”

- “Then comes the publication of the results of your research, and the published literature is just the tip of the data iceberg. By this I mean that people collect a lot of data and then reduce this down to some number of column inches in Science or Nature—or 10 pages if it is a computer science person writing. So what I mean by data iceberg is that there is **a lot of data** that is **collected** but not curated or published in any systematic way.“

Based on the transcript of a talk given by Jim Gray
to the NRC-CSTB1 in Mountain View, CA, on January 11, 2007

How to deal with Big Data
Advice From Jim Gray

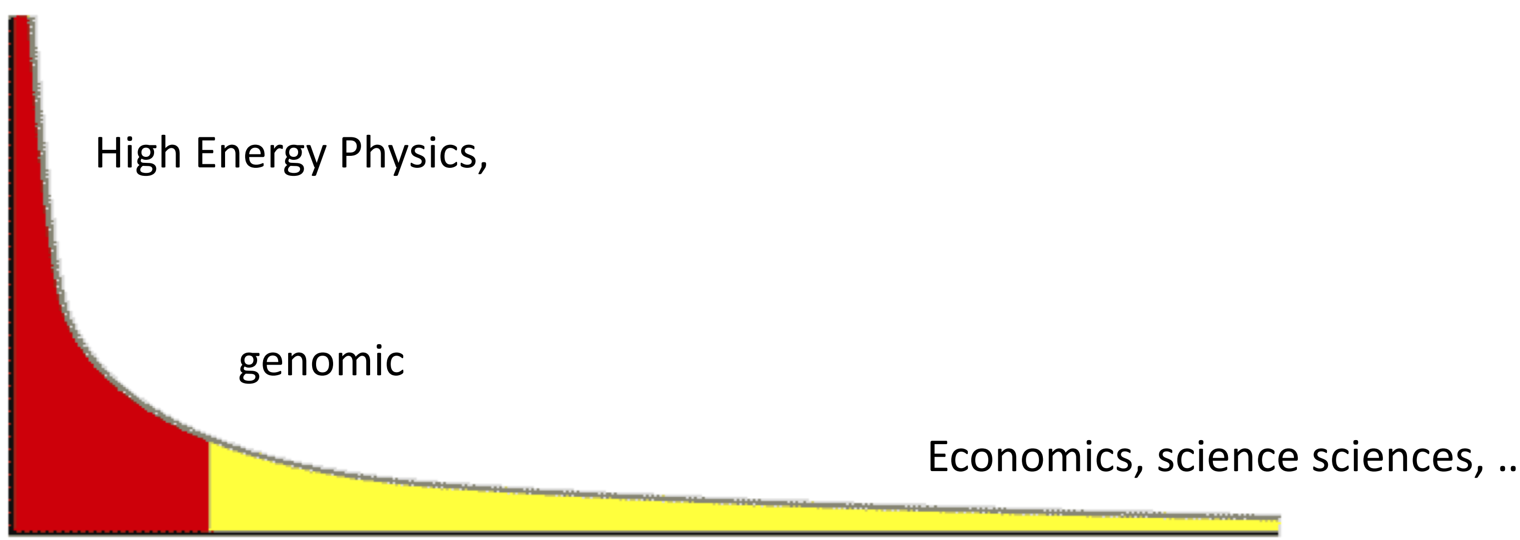
1. Analysing Big data requires scale-out solutions not scale-up solutions
2. **Move** the analysis to the data.
3. Work with scientists to find the most common "**20 queries**" and make them fast.
4. Go from "**working to working.**"



Source: Robert Grossman, Collin Bennec University of Chicago Open Data Group

Data keep on growing

- Google processes **20 PB a day** (2008)
- Wayback Machine has 3 PB + **100 TB/month** (3/2009)
- Facebook has 2.5 PB of user data + **15 TB/day** (4/2009)
- eBay has 6.5 PB of user data + **50 TB/day** (5/2009)
- CERN's Large Hydron Collider (LHC) generates **15 PB/year**



Collectively "Long Tail" science is generating **a lot of data**

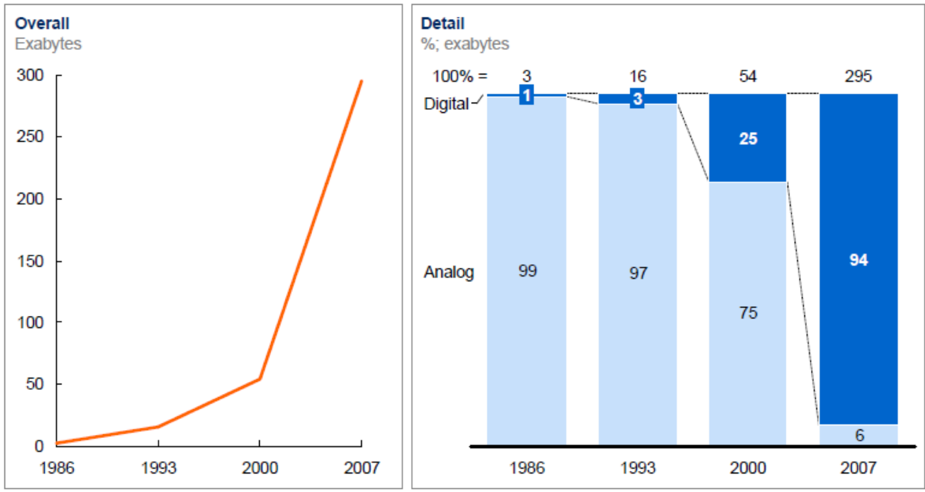
Big data was big news in 2012

- The Harvard Business Review talks about it as "*The Management Revolution*".

- The Wall Street Journal
  "*Meet the New Big Data*",
  "*Big Data is on the Rise, Bringing Big Questions*".

Data storage has grown significantly, shifting markedly from analog to digital after 2000

Global installed, optimally compressed, storage

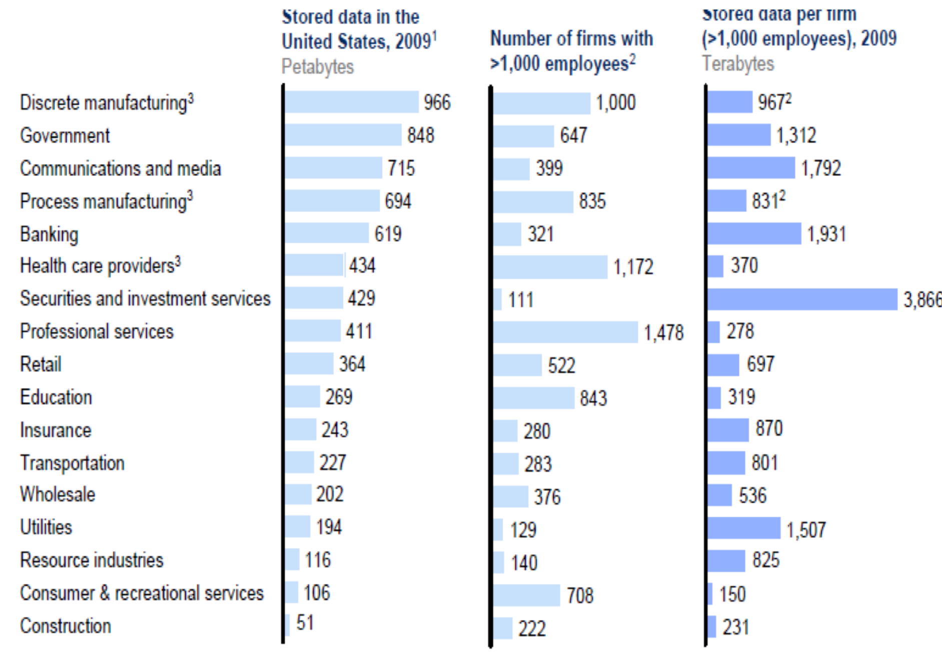


NOTE: Numbers may not sum due to rounding.
SOURCE: Hilbert and López, "The world's technological capacity to store, communicate, and compute information," *Science*, 2011

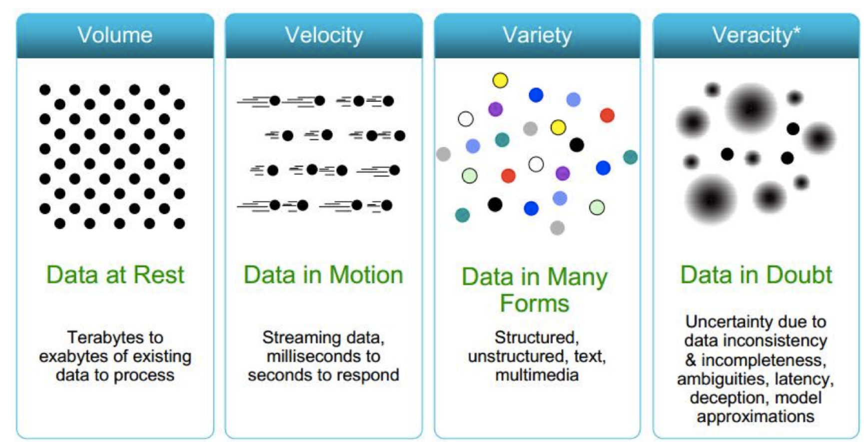
Where Big Data Comes From?

- Big Data is not Specific application type, but rather a trend –or even a collection of Trends- napping multiple application types

- Data growing in multiple ways
  - More data (**volume** of data)
  - More Type of data (**variety** of data)
  - Faster Ingest of data (**velocity** of data)
  - More Accessibility of data (internet, instruments , …)
  - Data Growth and availability exceeds organization ability to make intelligent decision based on it



1 Storage data by sector derived from IDC.



Addison Snell CEO. Intersect360, Research

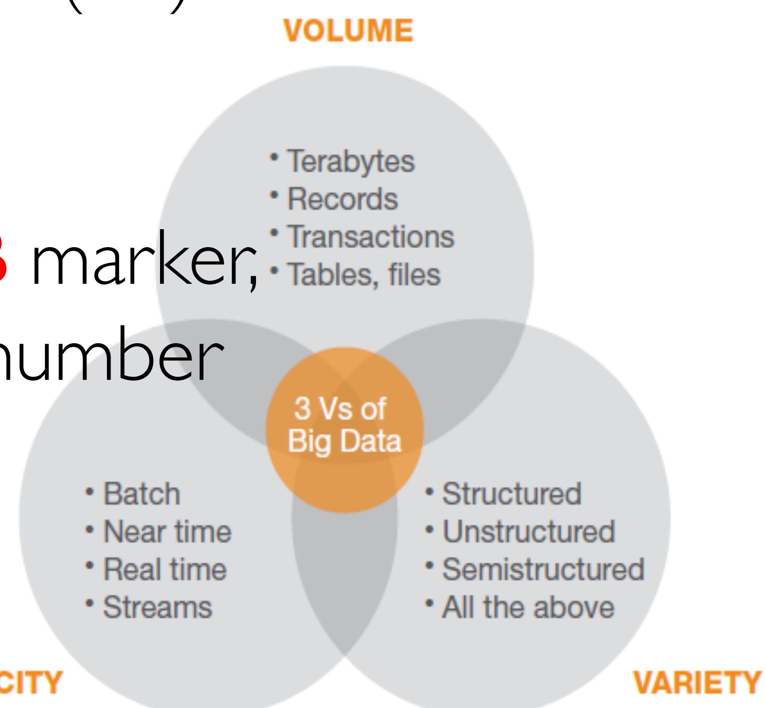
content

- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

volume, variety, and velocity

- Aggregation that used to be measured in petabytes (PB) is now referenced by a term: zettabytes (ZB).
  - A **zettabyte** is a **trillion gigabytes** (GB)
  - or a **billion terabytes**

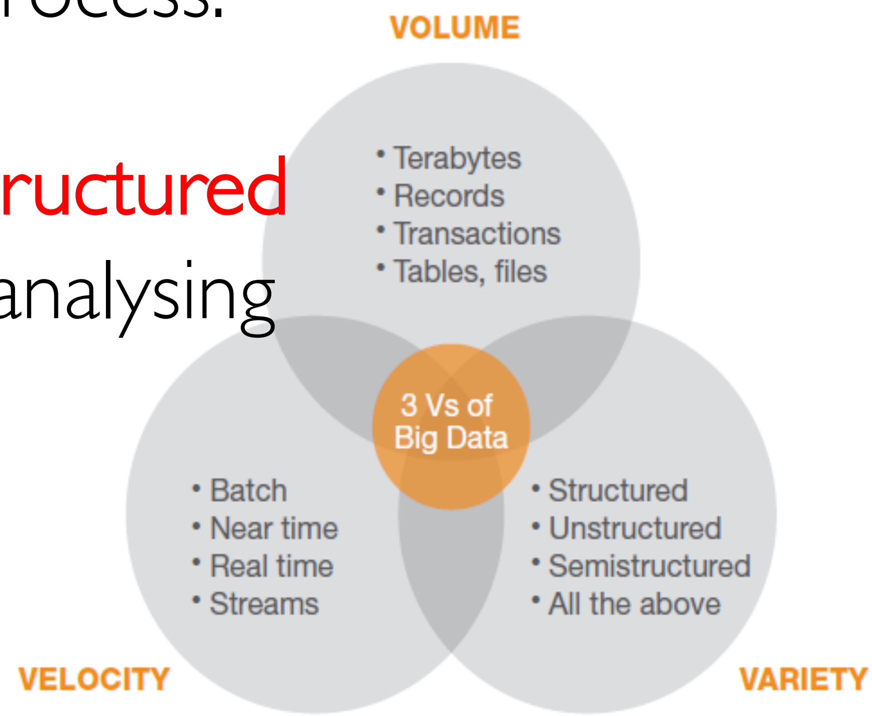
- in 2010, we crossed the 1ZB marker, and at the end of 2011 that number was estimated to be 1.8ZB



volume, variety, and velocity

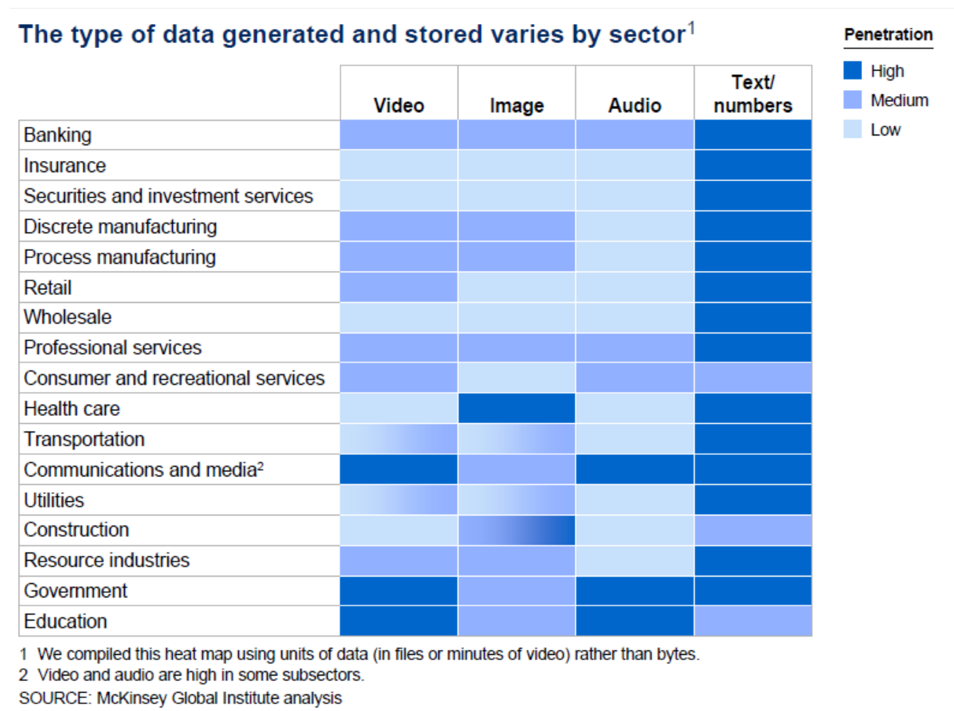
- The variety characteristic of Big Data is really about trying to **capture all** of the data that pertains to our decision-making process.

- Making sense out of unstructured data, such as opinion, or analysing images.



volume, variety, and velocity
(Type of Data)

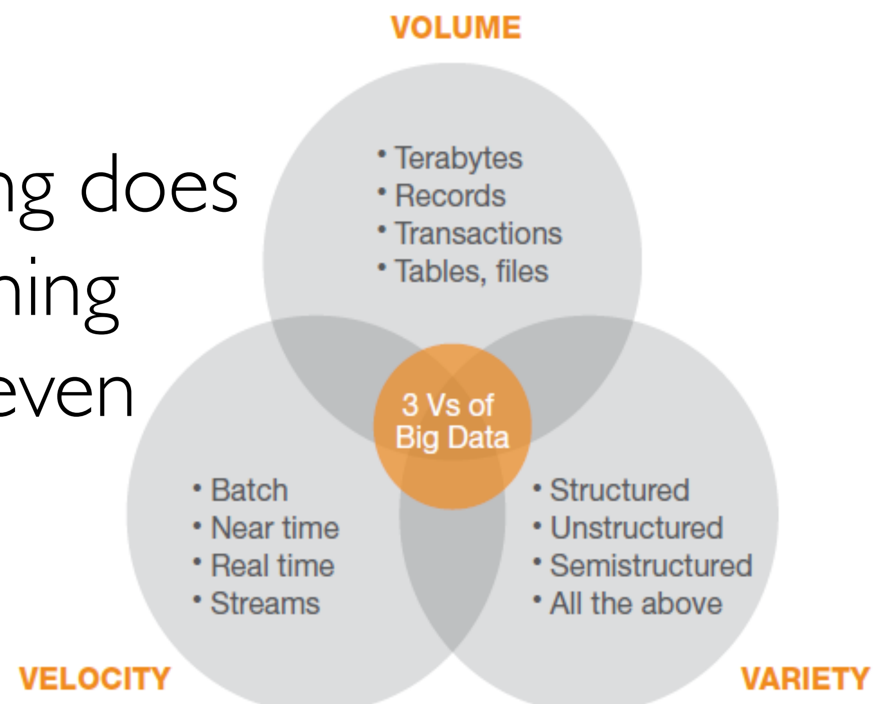
- Relational Data (Tables/Transaction/Legacy Data)
- Text Data (Web)
- Semi-structured Data (XML)
- Graph Data
  - Social Network, Semantic Web (RDF), …
- Streaming Data
  - You can only scan the data once



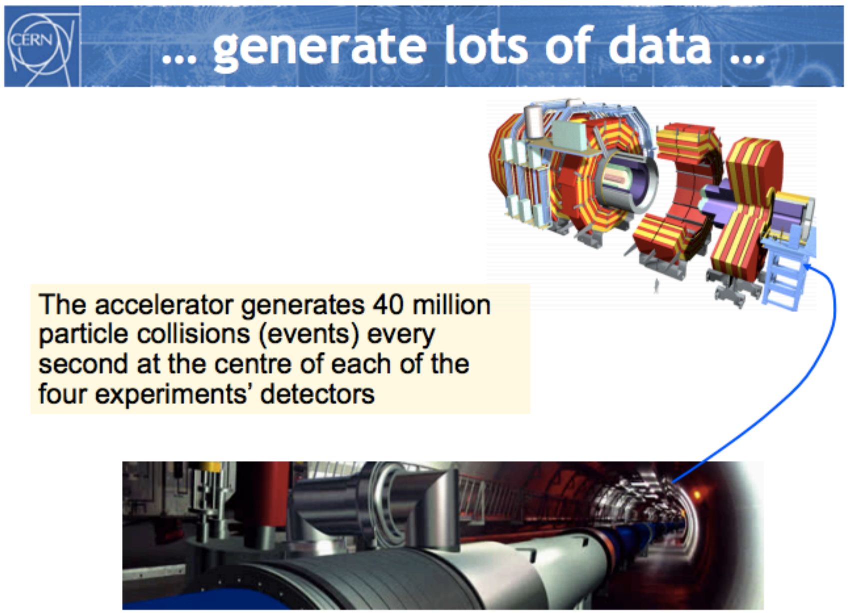
volume, variety, and velocity

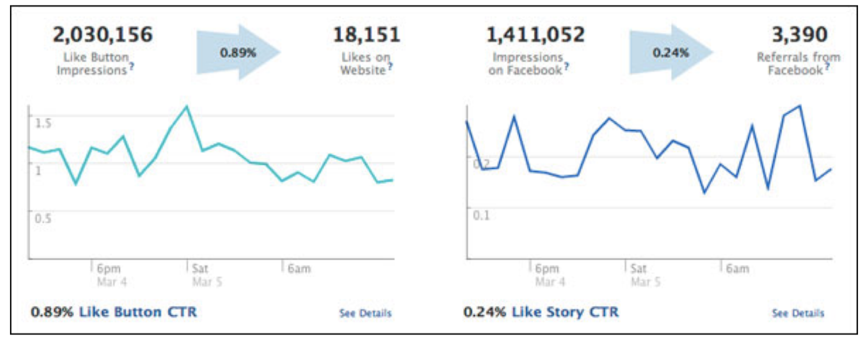
- velocity is the rate at which data arrives at the enterprise and is processed or well understood

- In other terms "How long does it take you to do something about it or know it has even arrived?"



volume, variety, and velocity





Today, it is possible using real-time analytics to optimize 

buttons across both website and on Facebook.

FaceBook use anonymised data to show the number of times people:

- saw Like buttons,
- clicked Like buttons,
- saw Like stories on Facebook,
- and clicked Like stories to visit a given website.

Not All analytics are real time
(from Analytics @ Twitter)

- Counting
  - How many request?
  - What's the average latency?
  - How many signups, sms, tweets?

Real time (msec/sec)

- Correlating
  - Desktop vs Mobile user ?
  - What devices fail at the same time?
  - What features get user hooked?

Near real time(Min/Hours)

- Researching
  - What features get re-tweeted
  - Duplicate detection
  - Sentiment analysis

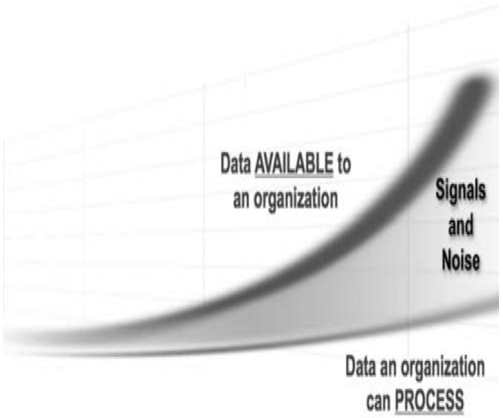
Batch (Days..)

volume, variety, velocity, and veracity

- Veracity refers to the quality or trustworthiness of the data.

- A common complication is that the data is saturated with both **useful signals** and lots of noise (data that can't be trusted)

LHC ATLAS detector generates about 1 Petabyte raw data per second, during the collision time (about 1 ms)



Big Data platform must include the **key imperatives**

	Big Data Platform Imperatives	Technology Capability
1	Discover, explore, and navigate Big Data sources	Federated Discovery, Search, and Navigation
2	Extreme performance–run analytics closer to data	Massively Parallel Processing Analytic appliances
3	Manage and analyze unstructured data	Hadoop File System/MapReduce Text Analytics
4	Analyze data in motion	Stream Computing
5	Rich library of analytical functions and tools	In-Database Analytics Libraries Big Data Visualization
6	Integrate and govern all data sources	Integration, Data Quality, Security, Lifecycle Management, MDM, etc

The Big Data platform manifesto: imperatives and underlying technologies

content

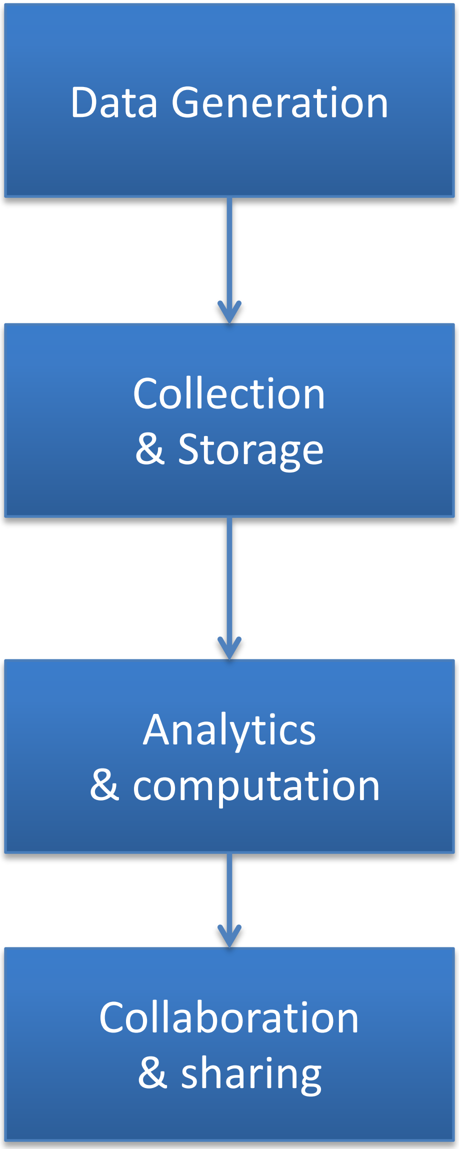
- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

Data Analytics pipeline

Analytics Characteristics are not new

- **Value**: produced when the analytics output is put into action
- **Veracity**: measure of accuracy and timeliness
- **Quality**:
  - well-formed data
  - Missing values
  - cleanliness

Data types have differing pre-analytics needs



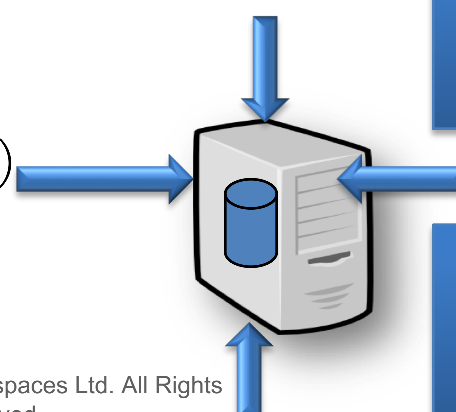
content

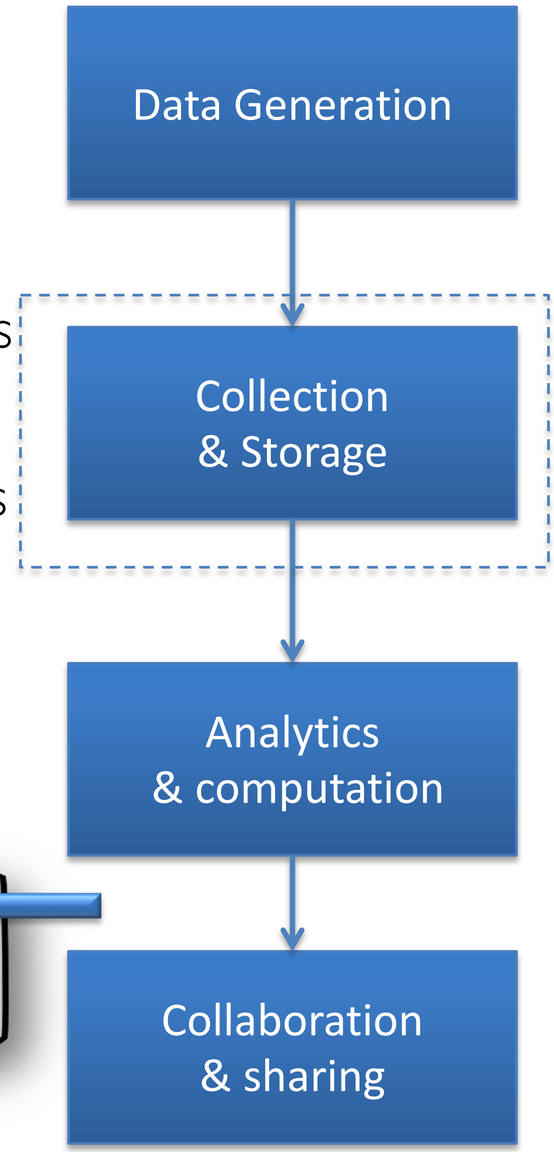
- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

Traditional analytics applications

- Scale-up Database
  - Use traditional SQL database
  - Use stored procedure for event driven reports
  - Use flash-based disks to reduce disk I/O
  - Use read only replica to scale-out read queries

- Limitations
  - Doesn't scale on write
  - Extremely expensive (HW + SW)

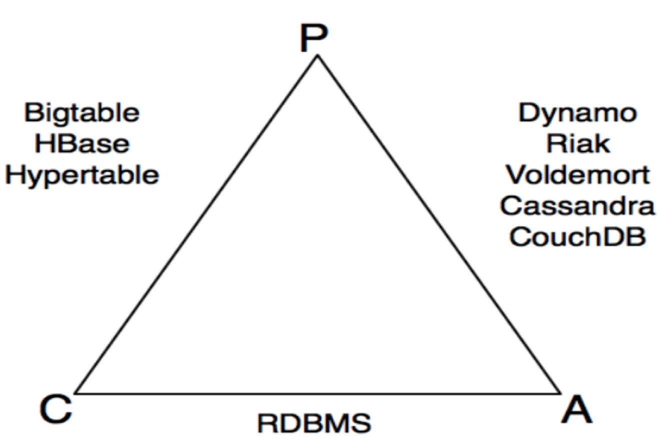




NoSQL

"Work with scientists to find the most common "**20 queries**" and make them fast." How to deal with Big Data Advice From Jim Gray (advice number 3)

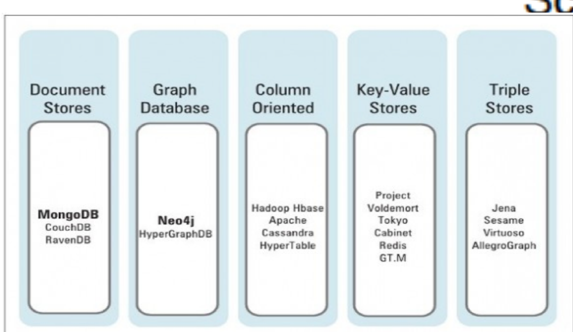
- Use distributed database
  - Hbase, Cassandra, MongoDB
- Pros
  - Scale on write/read
  - Elastic
- Cons
  - Read latency
  - Consistency tradeoffs are hard
  - Maturity – fairly young technology



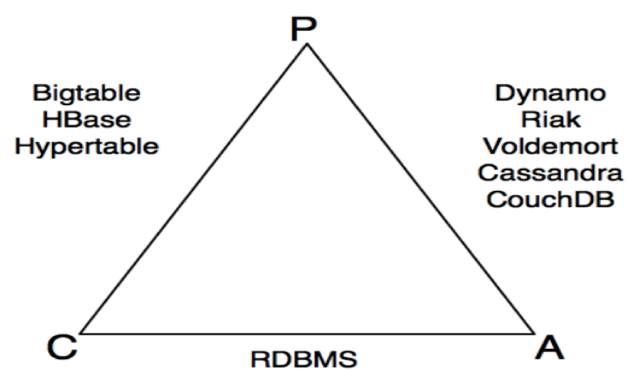
NoSQL

Year	System/ Paper	Scale to 1000s	Primary Index	Secondary Indexes	Transactions	Joins/ Analytics	Integrity Constraints	Views	Language/ Algebra	Data model	my label
1971	RDBMS	O	✔	✔	✔	✔	✔	✔	✔	tables	sql-like
2003	memcached	✔	✔	O	O	O	O	O	O	key-val	nosql
2004	MapReduce	✔	O	O	O	✔	O	O	O	key-val	batch
2005	CouchDB	✔	✔	✔	record	MR	O	✔	O	document	nosql
2006	BigTable (Hbase)	✔	✔	✔	record	compat. w/MR	/	O	O	ext. record	nosql
2007	MongoDB	✔	✔	✔	EC, record	O	O	O	O	document	nosql
2007	Dynamo	✔	✔	O	O	O	O	O	O	ext. record	nosql
2008	Pig	✔	O	O	O	✔	/	O	✔	tables	sql-like
2008	HIVE	✔	O	O	O	✔	✔	O	✔	tables	sql-like
2008	Cassandra	✔	✔	✔	EC, record	O	✔	✔	O	key-val	nosql
2009	Voldemort	✔	✔	O	EC, record	O	O	O	O	key-val	nosql
2009	Riak	✔	✔	✔	EC, record	MR	O			key-val	nosql
2010	Dremel	✔	O	O	O	/	✔	O	✔	tables	sql-like
2011	Megastore	✔	✔	✔	entity groups	O	/	O	/	tables	nosql
2011	Tenzing	✔	O	O	O	O	✔	✔	✔	tables	sql-like
2011	Spark/Shark	✔	O	O	O	✔	✔	O	✔	tables	sql-like
2012	Spanner	✔	✔	✔	✔	?	✔	✔	✔	tables	sql-like
2012	Accumulo	✔	✔	✔	record	compat. w/MR	/	O	O	ext. record	nosql
2013	Impala	✔	O	O	O	✔	✔	O	✔	tables	sql-like

Scale was the primary motivation!



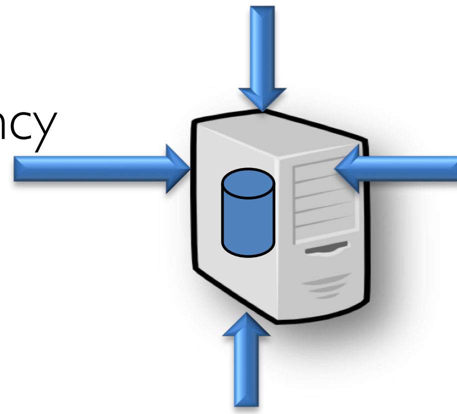
Bill Howe, UW

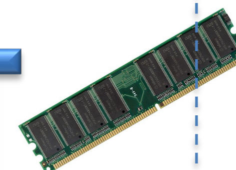


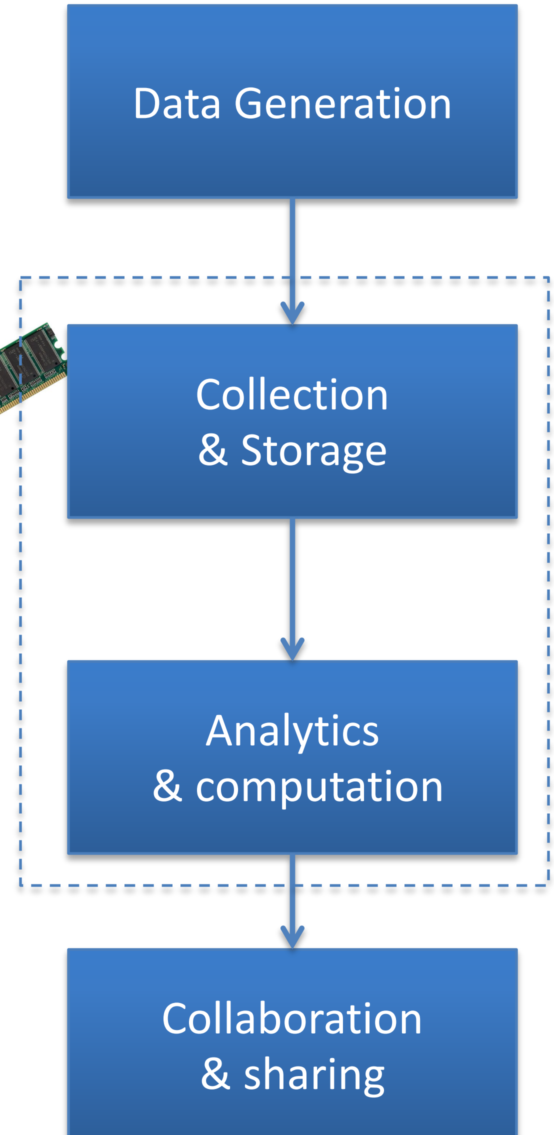
CEP – Complex Event Processing

- Process the data as it comes
- Maintain a window of the data in-memory

- Pros:
  - Extremely low-latency
  - Relatively low-cost



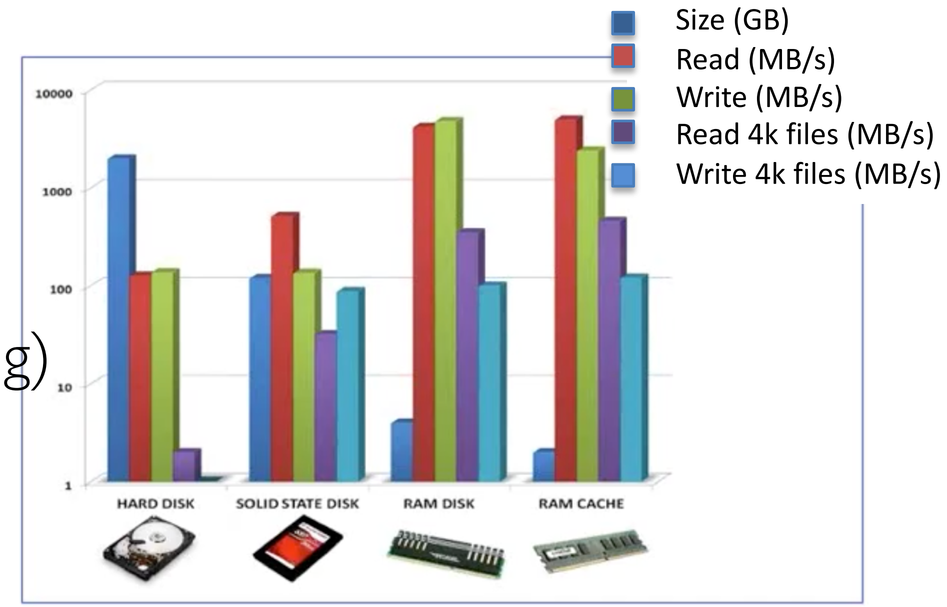




- Cons
  - Hard to scale (Mostly limited to **scale-up**)
  - Not agile - Queries must be pre-generated
  - Fairly complex

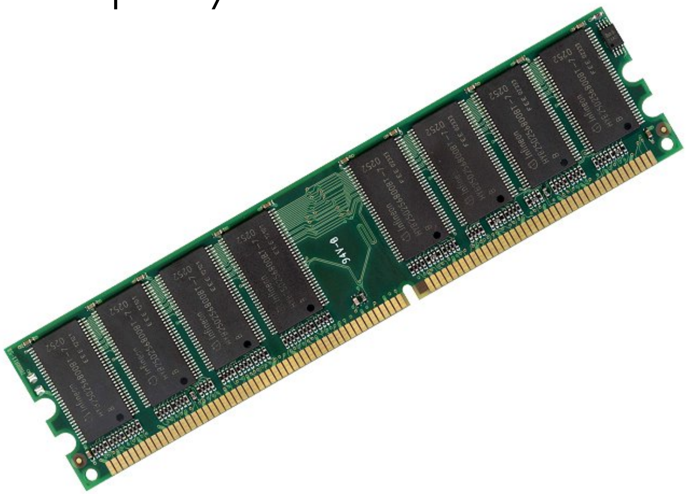
In Memory Data Grid

- **Distributed** in-memory database
  - Scale out (Horizontal scaling)



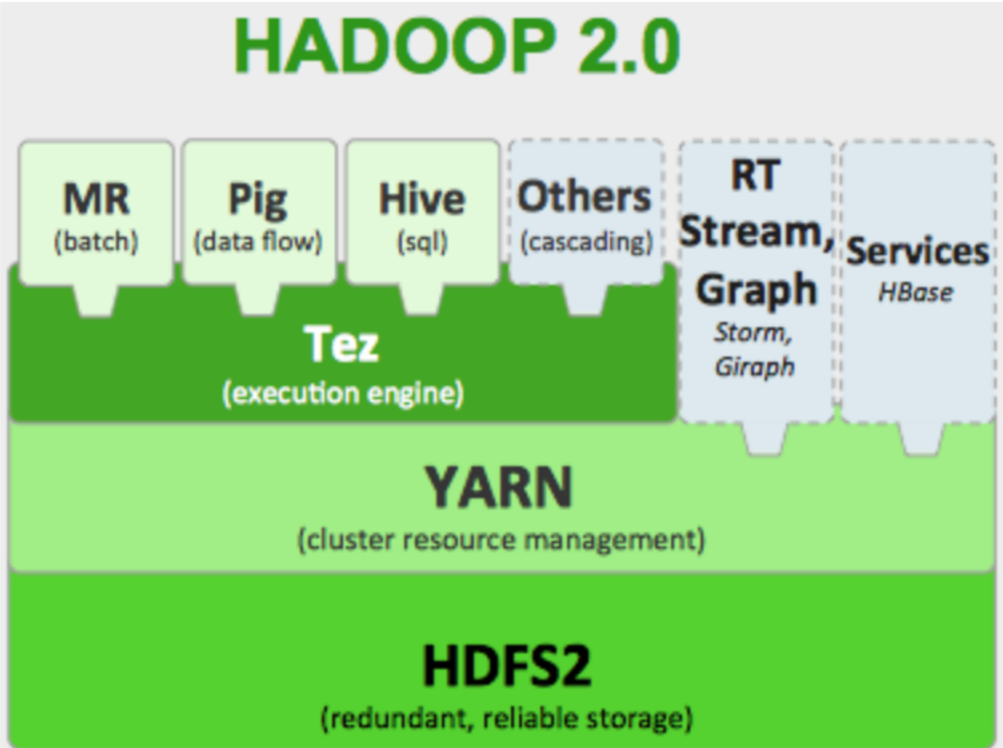
- Pros
  - Scale on write/read
  - Fits to event driven (CEP style) , ad-hoc query model

- Cons
  - **Cost** of memory vs disk
  - Memory **capacity** is limited

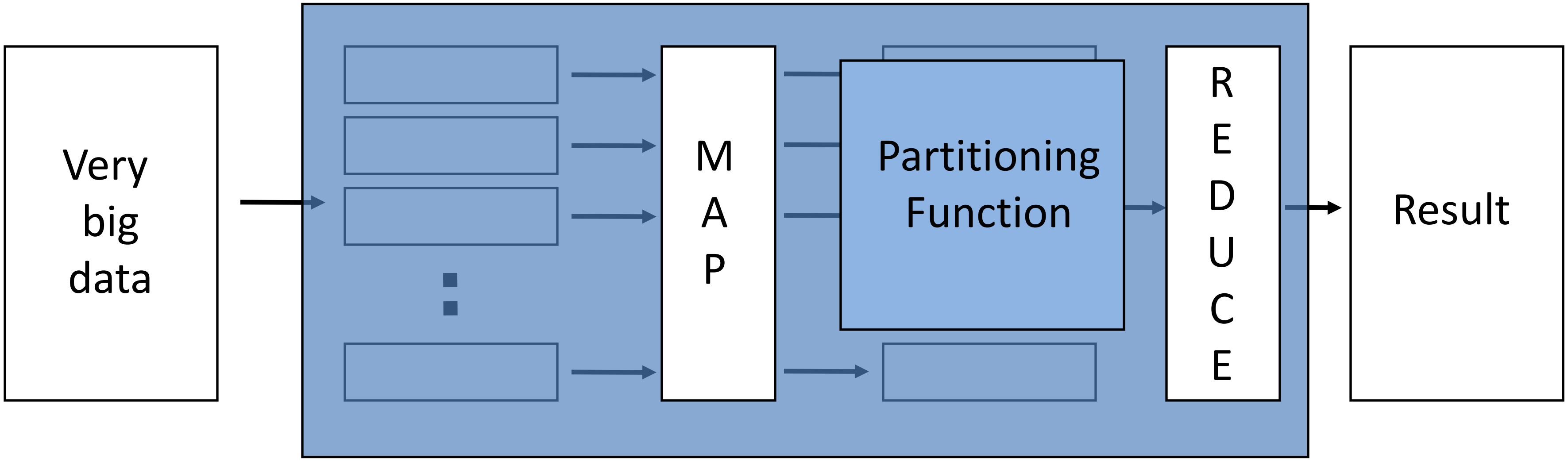


Hadoop MapReudce

- Distributed **batch** processing
- Pros
  - Designed to process massive amount of data
  - Mature
  - Low cost
- Cons
  - **Not** real-time



Map Reduce



- Map:
  - Accepts
    - *input* key/value pair
  - Emits
    - *intermediate* key/value pair
- Reduce :
  - Accepts
    - *intermediate* key/value* pair
  - Emits
    - *output* key/value pair

Sorting 1 TB of DATA

- Estimate:
  - read 100MB/s, write 100MB/s
  - no disk seeks, **instant sort**
  - 341 minutes → 5.6 hours
- The terabyte benchmark winner (2008):
  - 209 seconds (3.48 minutes)
  - 910 nodes x (4 dual-core processors, 4 disks, 8 GB memory)
- October 2012
  - 55 seconds [1]

[1] http://www.youtube.com/watch?v=XbUPlbYxT8g&feature=youtu.be

Hadoop Map/Reduce – Reality check..

"With the paths that go through Hadoop [at Yahoo!], **the latency is about fifteen minutes.** … [I]t will never be true real-time.." (**Yahoo** CTO Raymie Stata)

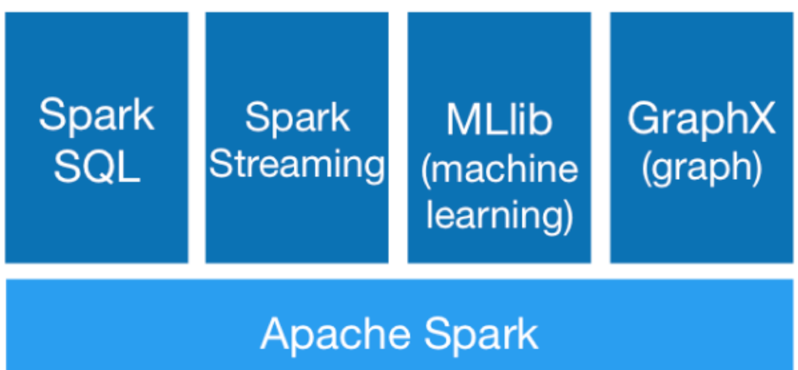
Hadoop/Hive..Not realtime. Many dependencies. Lots of points of failure. Complicated system. Not dependable enough to hit realtime goals (Alex Himel, Engineering Manager at **Facebook**.)

"MapReduce and other batch-processing systems **cannot process small updates individually as they rely on creating large batches for efficiency,**" (**Google** senior director of engineering Eisar Lipkovitz)

Apache Spark

Lightning-fast cluster computing (in-memory)

- Generality
  - Combine SQL, streaming, complex analytics.
- Runs Everywhere
  - Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources (HDFS, Cassandra, HBase, and S3)
- Ease of Use
  - Write applications quickly in Java, Scala, Python, R.



Spark

Developer(s)	Apache Software Foundation, UC Berkeley AMPLab, Databricks
Initial release	May 30, 2014; 18 months ago
Stable release	v1.5.2 / November 9, 2015; 51 days ago
Development status	Active
Written in	Scala, Java, Python, R
Operating system	Linux, Mac OS, Windows
Type	data analytics, machine learning algorithms
License	Apache License 2.0
Website	spark.apache.org

Apache Spark

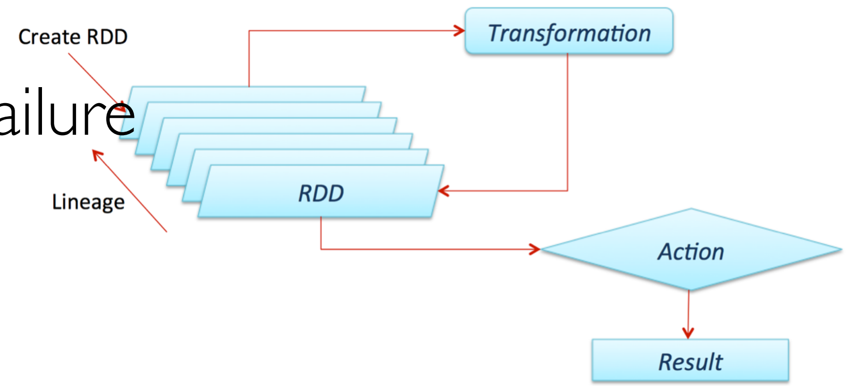
Lightning-fast cluster computing

Resilient Distributed Datasets (RDD)

– **Immutable**, partitioned **collections** of records
– can only be built through **coarse-grained** deterministic transformations (map, filter, join...)

Efficient fault-tolerance using lineage

– Log coarse-grained operations instead of fine-grained data updates
– An RDD has enough information about how it's derived from other dataset
– Recompute lost partitions on failure

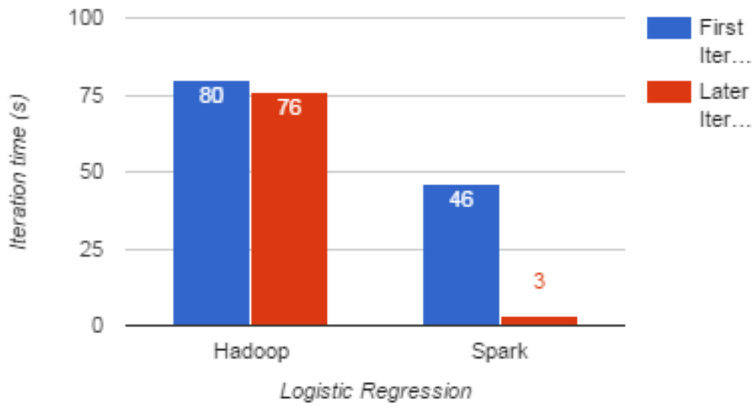


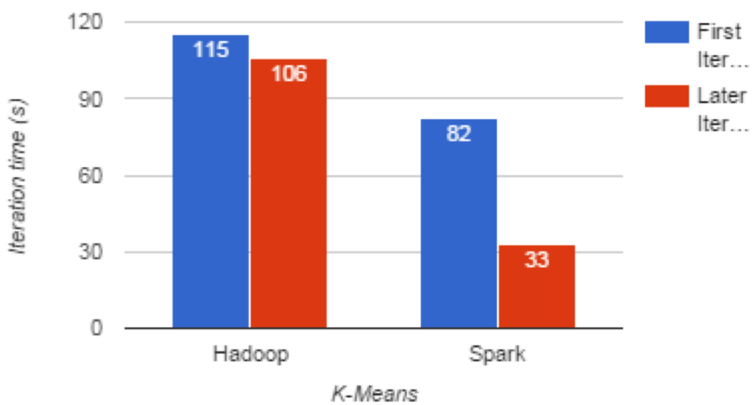
https://dzone.com/refcardz/apache-spark

Apache Spark

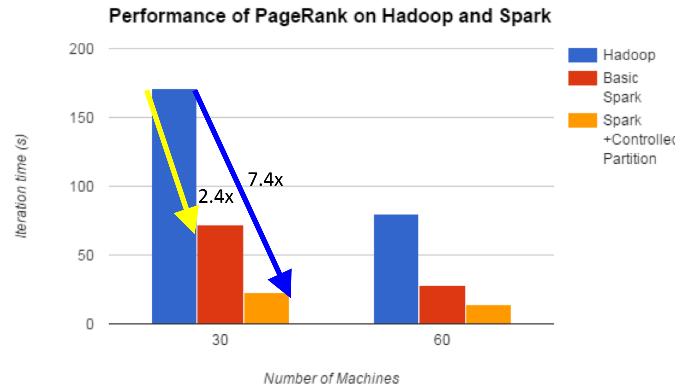
Lightning-fast cluster computing

- 10 iterations on 100GB data using 25-100 machines

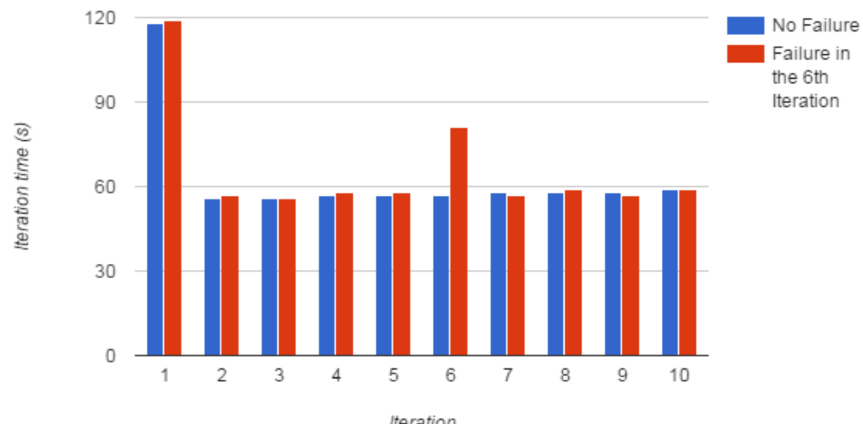




- 10 iterations on 54GB data with approximately 4M articles



- 10 iterations of k-means on 75 nodes, each iteration contains 400 tasks on 100GB data



Matei Zaharia, Mosharaf Chowdhury, Resilient Distributed Datasets A Fault-Tolerant Abstraction for In-Memory Cluster Computing NSDI'12 presentation

Apache Storm

By Nathan Marz

- Storm is a distributed real-time computation system that solves typical
  - downsides of queues & workers systems.
  - Built with Big Data in mind (the "Hadoop of realtime").
- Storm Trident (high level abstraction over Storm core)
  - Micro-batching (~ streaming)

STORM

Distributed and fault-tolerant realtime computation

Developer(s)	Backtype, Twitter
Stable release	1.0.5 / 15 September 2017
Written in	Clojure & Java
Operating system	Cross-platform
Type	Distributed stream processing
License	Apache License 2.0
Website	storm.apache.org

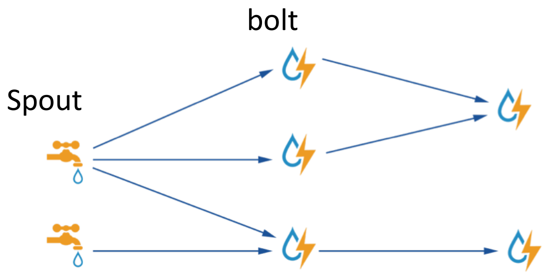
Apache Storm

Core concepts

- Topologies
- Spouts and bolts
- Data model
- Groupings

What storm does

- Distributes code and configurations
- Manage processes (robust)
- Monitors topologies & reassigns failed tasks
- Provides reliability by tracking tuples
- Routing and partitioning of Streams
- Serialization
- Fine-Grained performance stats of topologies



topology

Tuple = datum containing 1+ fields

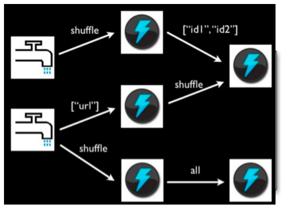
(1.1.1.1, "foo.com")

Values can be of any type such as Java primitive types, String, byte[].
Custom objects should provide their own Kryo serializer though.

Stream = unbounded sequence of tuples

```
       ...
(1.1.1.1, "foo.com")
(2.2.2.2, "bar.net")
(3.3.3.3, "foo.com")
       ...
```

http://storm.incubator.apache.org/documentation/Concepts.html



Grouping: shuffle, Fields, All, Global,

Performance

- Apache Storm has many use cases:
  - realtime analytics,
  - online machine learning,
  - distributed RPC, ETL.

- A benchmark clocked it at over a million tuples processed per second per node. It is scalable, fault-tolerant, guarantees your data will be processed, and is easy to set up and operate.

- Apache Storm integrates with the existing queueing and database technologies.

Apache Kafka

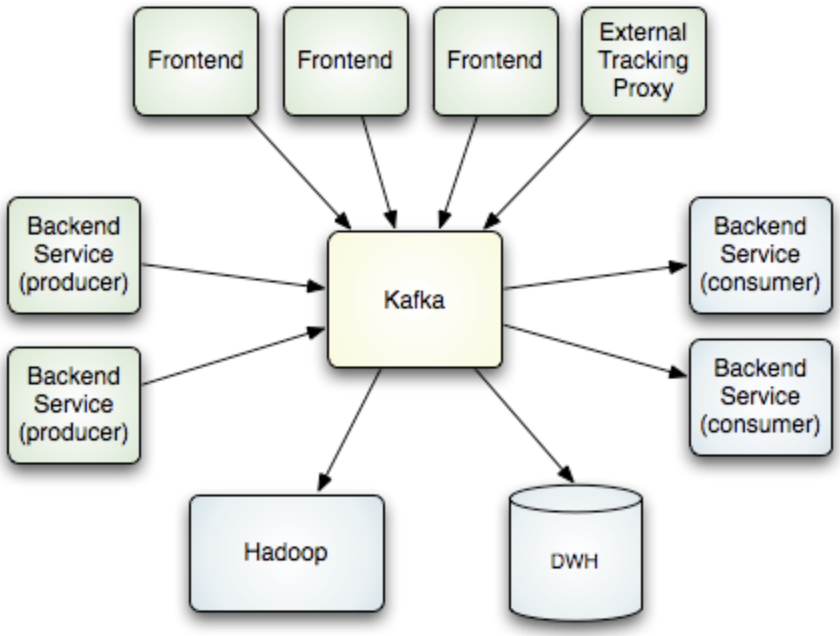
A high-throughput distributed messaging system

- Apache Kafka is publish-subscribe messaging rethought as a distributed commit log.
- Kafka maintains feeds of messages in categories called *topics*.
  - Processes can <u>publish</u> messages to a Kafka (topic *producers*).
  - processes can <u>subscribe</u> to topics and process the feed of published messages *consumers*.
- Kafka is run as a cluster comprised of one or more servers each of which is called a *broker*.

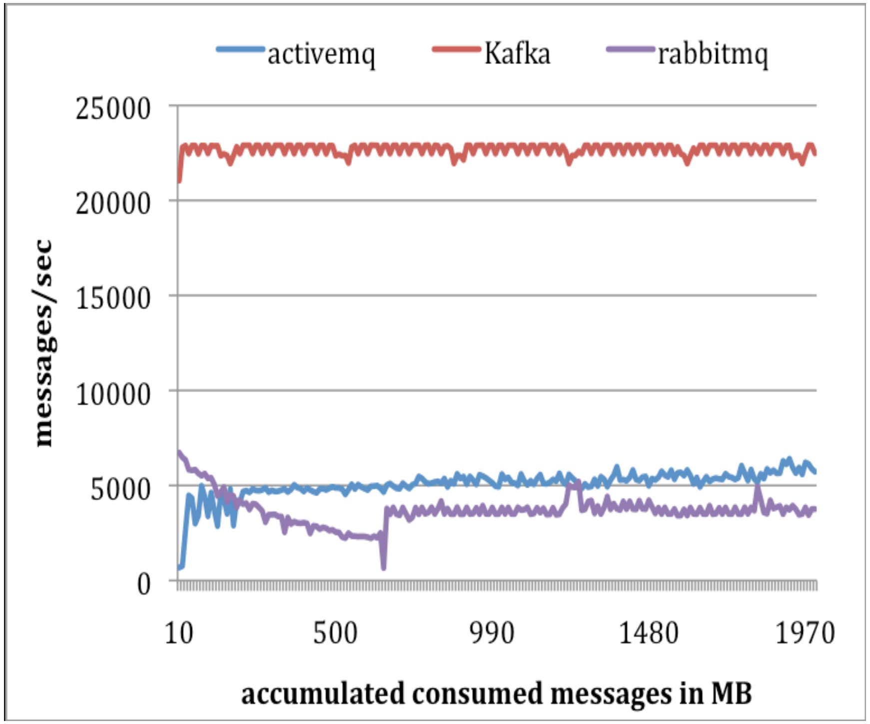
Developer(s)	Apache Software Foundation
Stable release	0.9 / November 2015; 1 month ago
Development status	Active
Written in	Scala
Operating system	Cross-platform
Type	Message broker
License	Apache License 2.0
Website	kafka.apache.org

Apache Kafka

A high-throughput distributed messaging system



Credit : http://kafka.apache.org/design.html

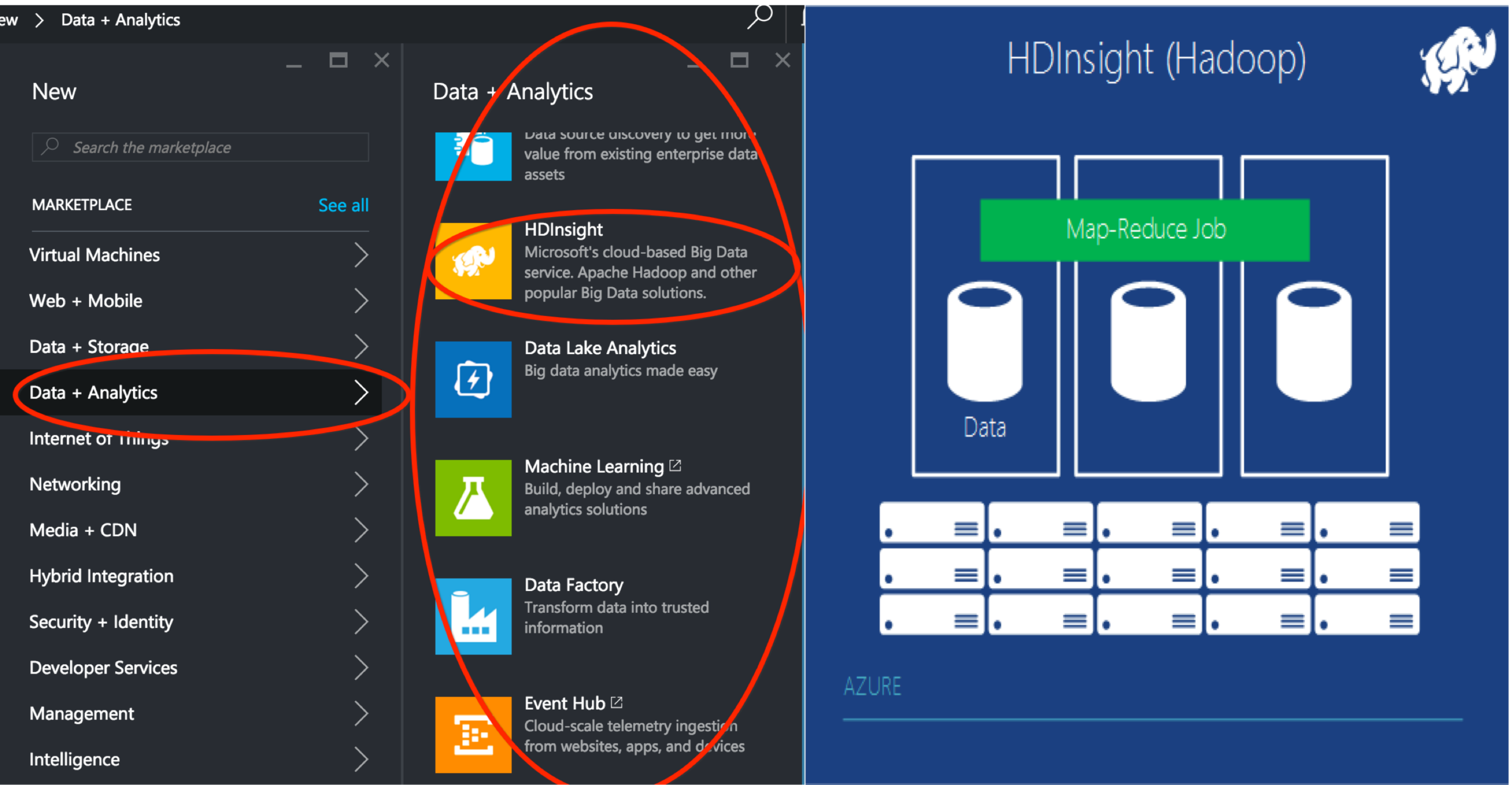


Consumer Performance

Credit : http://research.microsoft.com/en-us/UM/people/srikanth/netdb11/netdb11papers/netdb11-final12.pdf

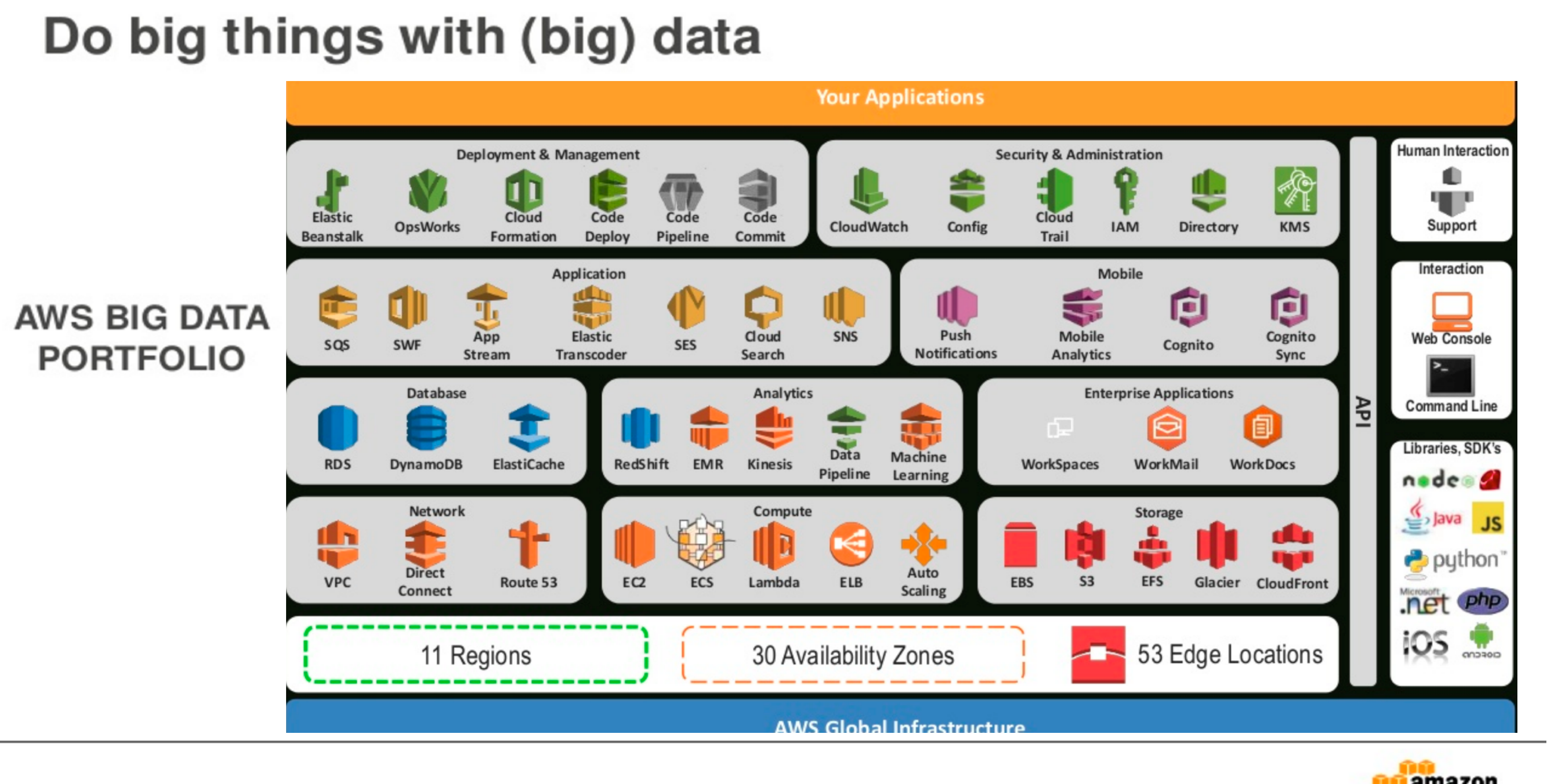
Big data Analytics in Microsoft Azure

- HDInsight
- Map reduce type job
- Other types of data analytics



Big data Analytics in AWS Cloud

- Redshift
- EMR
- Kinesis
- Data Pipeline
- Machine learning
- …



IBM BigInsights

- BigInsights analytical platform for persistent "big data" Based
  - on open sources platforms: Apache Spark and Apache Hadoop
  - IBM technologies: value-add services include Big SQL, Text Analytics, BigSheets, and Big R

https://www.ibm.com/support/knowledgecenter/en/SSPT3X/SSPT3X_welcome.html

Google BigQuery

A fast, economical and fully managed data warehouse for large-scale data analytics

- Google BigQuery[1] is a Restful web service that lets you do interactive analysis of massive datasets
  - up to billions of rows.
  - more features [2]

[1] https://cloud.google.com/bigquery
[2] https://cloud.google.com/bigquery#all-features

content

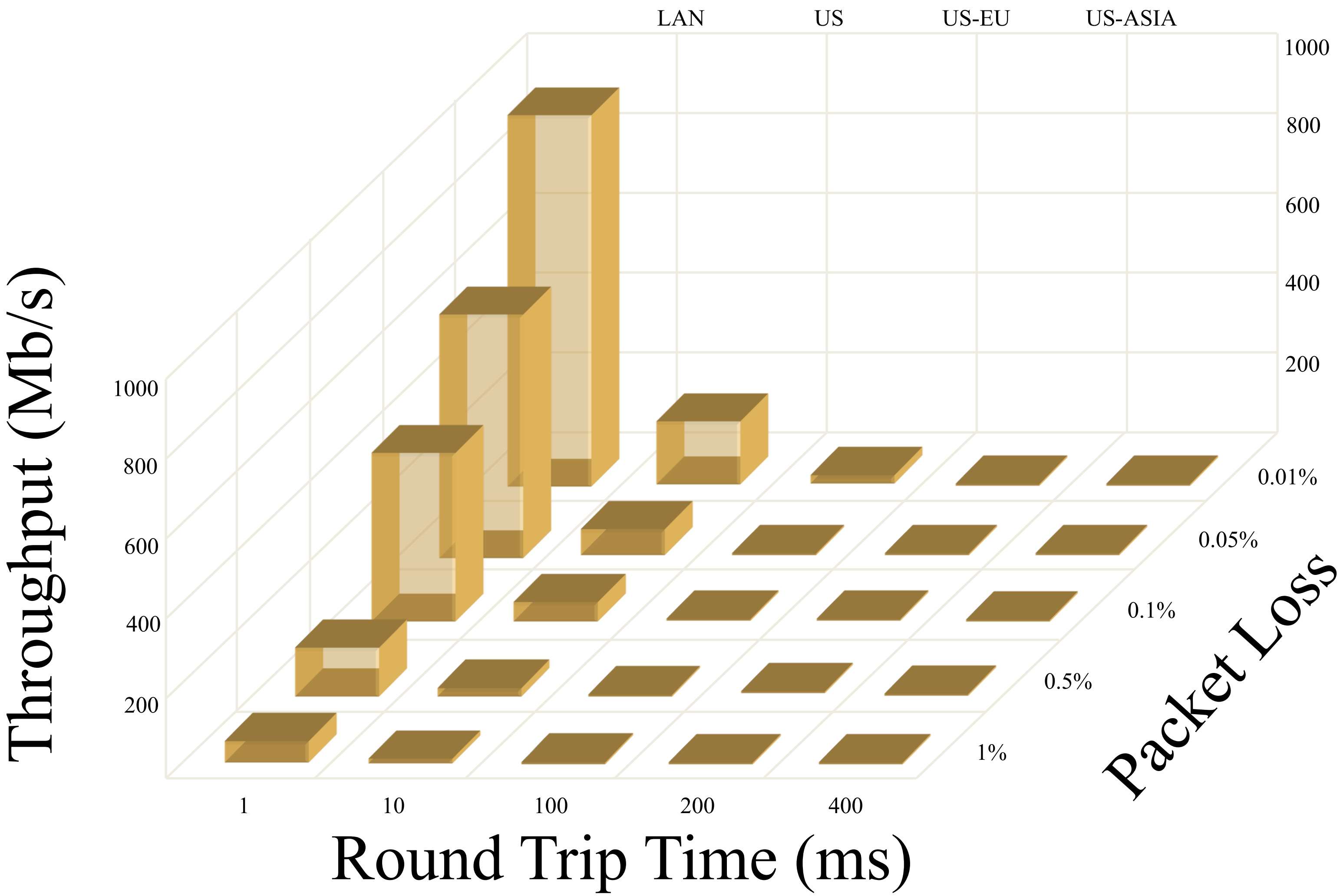
- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

The problem

- TCP Was never designed to move large datasets over wide area high Performance Networks.

- For loading a webpage, TCP is great.
- For sustained data transfer, it is far from ideal.
  - Most of the time even though the connection itself is good (let say 45Mbps), transfers are much slower.
  - There are two reason for a slow transfer over fast connections:
    - Latency
    - and packet loss bring TCP-based file transfer to a crawl.

TCP Throughput vs RTT and Packet Loss



Source: Yunhong Gu, 2007, experiments over wide area 1G.

The solutions

- Use parallel TCP streams
  - GridFTP
- Use specialized network protocols
  - UDT, FAST, etc.
- Use RAID to stripe data across disks to improve throughput when reading

These techniques are well understood in HEP, astronomy, but not yet in biology

Robert Grossman University of Chicago Open Data Group, November 14, 2011, https://www.slideshare.net/rgrossman/managing-big-data-sc-11-tutorial

Moving 113GB of Bio-mirror Data

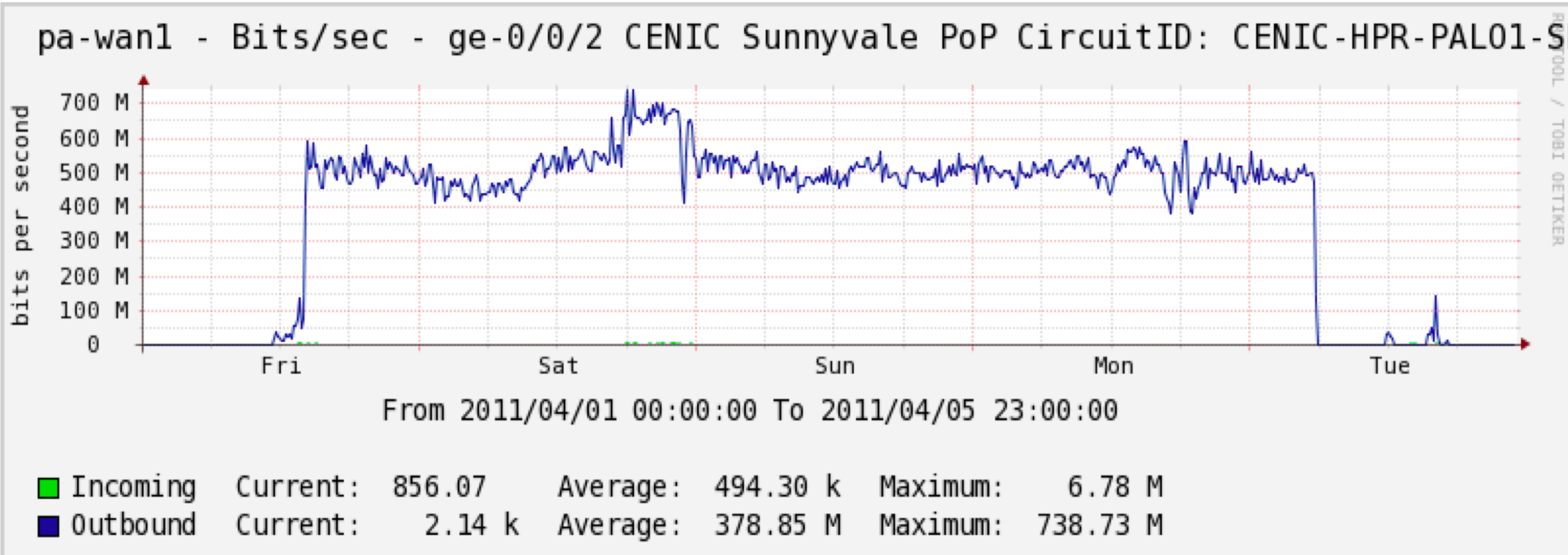
-

Site	RTT	TCP	UDT	TCP/UDT	Km
NCSA	10	**139**	139	1	**200**
Purdue	17	125	125	1	500
ORNL	25	361	120	3	1,200
TACC	37	616	120	55	2,000
SDSC	65	750	475	1.6	3,300
CSTNET	274	**3722**	304	12	**12,000**

- GridFTP TCP and UDT transfer times for 113 GB from gridip.bio--mirror.net/biomirror/ blast/ (Indiana USA).
  - All TCP and UDT times in minutes.
  - Source: http://gridip.bio-mirror.net/biomirror/

Robert Grossman University of Chicago Open Data Group, November 14, 2011, https://www.slideshare.net/rgrossman/managing-big-data-sc-11-tutorial

Case study: CGI 60 genomes



- Trace by Complete Genomics showing performance of moving 60 complete human genomes from Mountain View to Chicago using the open source Sector/UDT.
- Approximately 18 TB at about 0.5 Mbs on 1G link.

How FedEx Has More Bandwidth Than the Internet—and When That'll Change

- If you're looking to transfer hundreds of gigabytes of data, it's still—weirdly—faster to ship hard drives via FedEx than it is to transfer the files over the internet.
- "Cisco estimates that total internet traffic currently averages **167 terabits per second**. FedEx has a fleet of 654 aircraft with a lift capacity of 26.5 million pounds daily. A solid-state laptop drive weighs about 78 grams and can hold up to a terabyte. That means FedEx is capable of transferring 150 exabytes of data per day, or **14 petabits per second—almost a hundred times the current throughput of the internet**.

http://gizmodo.com/5981713/how-fedex-has-more-bandwidth-than-the-internetand-when-thatll-change

Migrate or transport exabyte-scale data sets into and out of AWS

content

- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

When to Consider a Big Data Solution
User point of view

- You're limited by your current platform or environment because you can't process the amount of data that you want to process

- You want to involve new sources of data in the analytics, but you can't, because it doesn't fit into schema-defined rows and columns without sacrificing fidelity or the richness of the data

When to Consider a Big Data Solution

- You need to ingest data as quickly as possible and need to work with a schema-on-demand
  - You're forced into a schema-on-write approach (the schema must be created before data is loaded),
  - but you need to ingest data quickly, or perhaps in a discovery process, and want the cost benefits of a schema-on-read approach (data is simply copied to the file store, and no special transformation is needed) until you know that you've got something that's ready for analysis?

When to Consider a Big Data Solution

- You want to analyse not just raw structured data, but also semi-structured and unstructured data from a wide variety of sources
- you're not satisfied with the effectiveness of your algorithms or models
  - when all, or most, of the data needs to be analysed
  - or when a sampling of the data isn't going to work

When to Consider a Big Data Solution

- you aren't completely sure where the investigation will take you, and you want elasticity of compute, storage, and the types of analytics that will be pursued—all of these became useful as we added more sources and new methods

If your answers to any of these questions are "yes," you need to consider a Big Data solution.

content

- General Introduction
- Definitions
- Data Analytics
- Solutions for Big Data Analytics
- The Network (Internet)
- When to consider BigData solution
- Scientific e-infrastructure – some challenges to overcome

Scientific e-infrastructure – some challenges to overcome

- Collection
  - How can we make sure that data are collected together with the **information** necessary **to re- use them**?
- Trust
  - How can we make informed judgements about whether certain data are **authentic** and can be **trusted**?
  - How can we judge which repositories we can trust? How can appropriate access and use of resources be granted or controlled

Riding the wave, How Europe can gain from the rising tide of scientific data

Scientific e-infrastructure – some challenges to overcome

- Usability
  - How can we move to a situation where non-specialists can overcome the barriers and be able to start sensible work on unfamiliar data

- Interoperability
  - How can we implement interoperability within disciplines and move to an overarching multi-disciplinary way of understanding and using data?
  - How can we find unfamiliar but relevant data resources beyond simple keyword searches, but involving a deeper probing into the data
  - How can automated tools find the information needed to tackle data

Riding the wave, How Europe can gain from the rising tide of scientific data

Scientific e-infrastructure – some challenges to overcome

- Diversity
  - How do we overcome the problems of diversity – heterogeneity of data, but also of backgrounds and data-sharing cultures in the scientific community?
  - How do we deal with the diversity of data repositories and access rules – within or between disciplines, and within or across national borders?
- Security
  - How can we guarantee data integrity?
  - How can we avoid data poisoning by individuals or groups intending to bias them in their interest?

Riding the wave, How Europe can gain from the rising tide of scientific data

References

1. T. Hey, S. Tansley, and K. Tolle, The Fourth Paradigm: Data-Intensive Scientific Discovery, T. Hey, S. Tansley, and K. Tolle, Eds. Microsoft, 2009.
   – Available: http://research.microsoft.com/en-us/collaboration/fourthparadigm/
2. Enabling knowledge creation in data-driven science
   – https://sciencenode.org/feature/enabling-knowledge-creation-data-driven-science.php
3. Science as an open enterprise: open data for open science
   – http://royalsociety.org/uploadedFiles/Royal_Society_Content/policy/projects/sape/2012-06-20-SAOE.pdf
4. Realtime Analytics for Big Data: A Facebook Case Study
   http://www.youtube.com/watch?v=viPRny0nq3o