SHORT HISTORY of BIG DATA & STREAMING PROGRAMMING TECHNOLOGY

Jeffrey Ricker

- 1991BS Mechanical Engineering (Robotics) Tulane University
- 1996US DOD High Performance Computing Modernization Program
- 1997 DARPA Shaolin Project
- 1998 Founded XMLSolutions Corp
- 2004 Founded Distributed Instruments LLC
- 2013 Amazon Big Data
- 2015 Founded The Ricker Lyman Robotic Company

Objective

Provide historical context of technology evolution leading up to streaming big data

Agenda

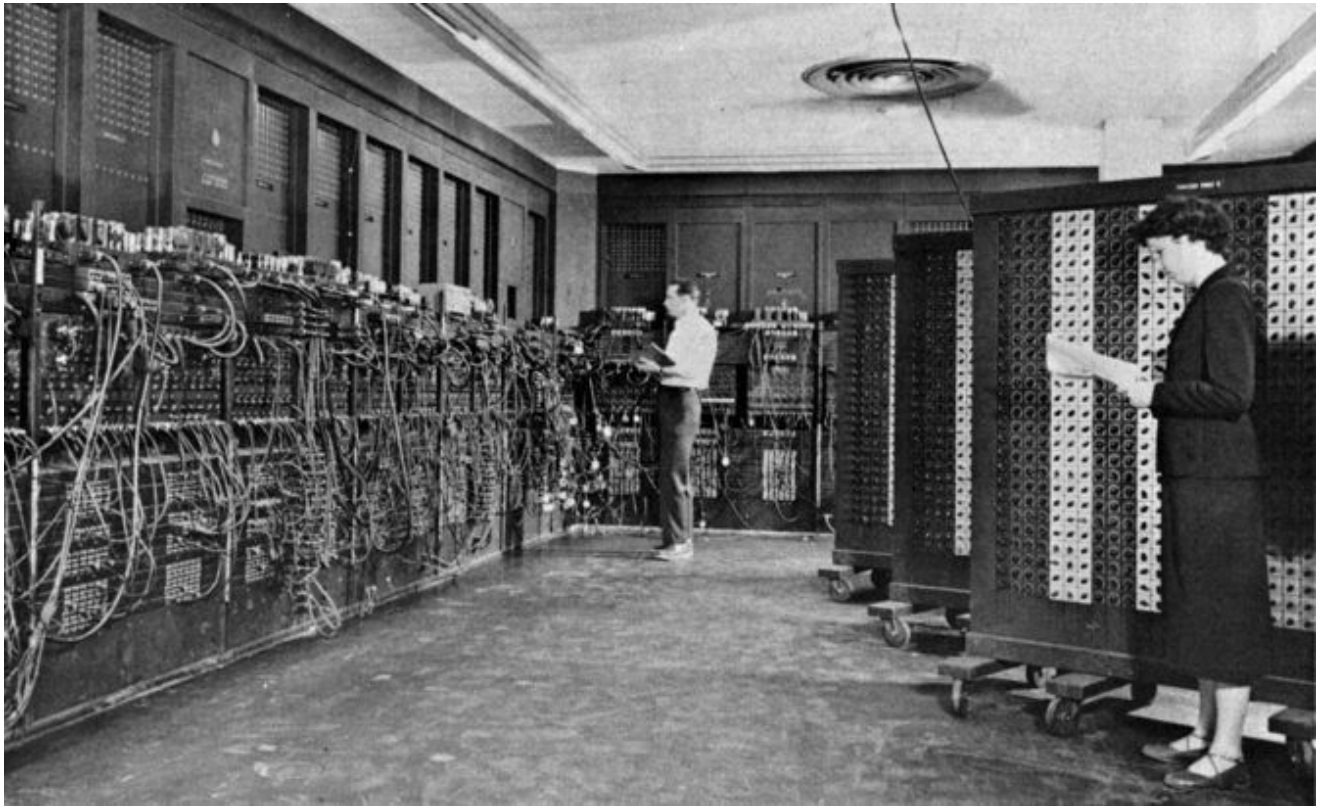
---

1. High performance computing
2. Open source
3. Hadoop (big data)
4. Functional programming
5. Streaming programming
6. Why history?

High Performance Computing

Size matters

1946 ENIAC



1964 CDC 6600

1976 Cray 1

1991 CM-5

1994 Beowulf cluster

2018 Summit

Summit has 4,356 nodes, each one equipped with two 22-core **Power9** CPUs, and six **NVIDIA Tesla V100 GPUs**. The nodes are linked together with a Mellanox dual-rail EDR InfiniBand network.

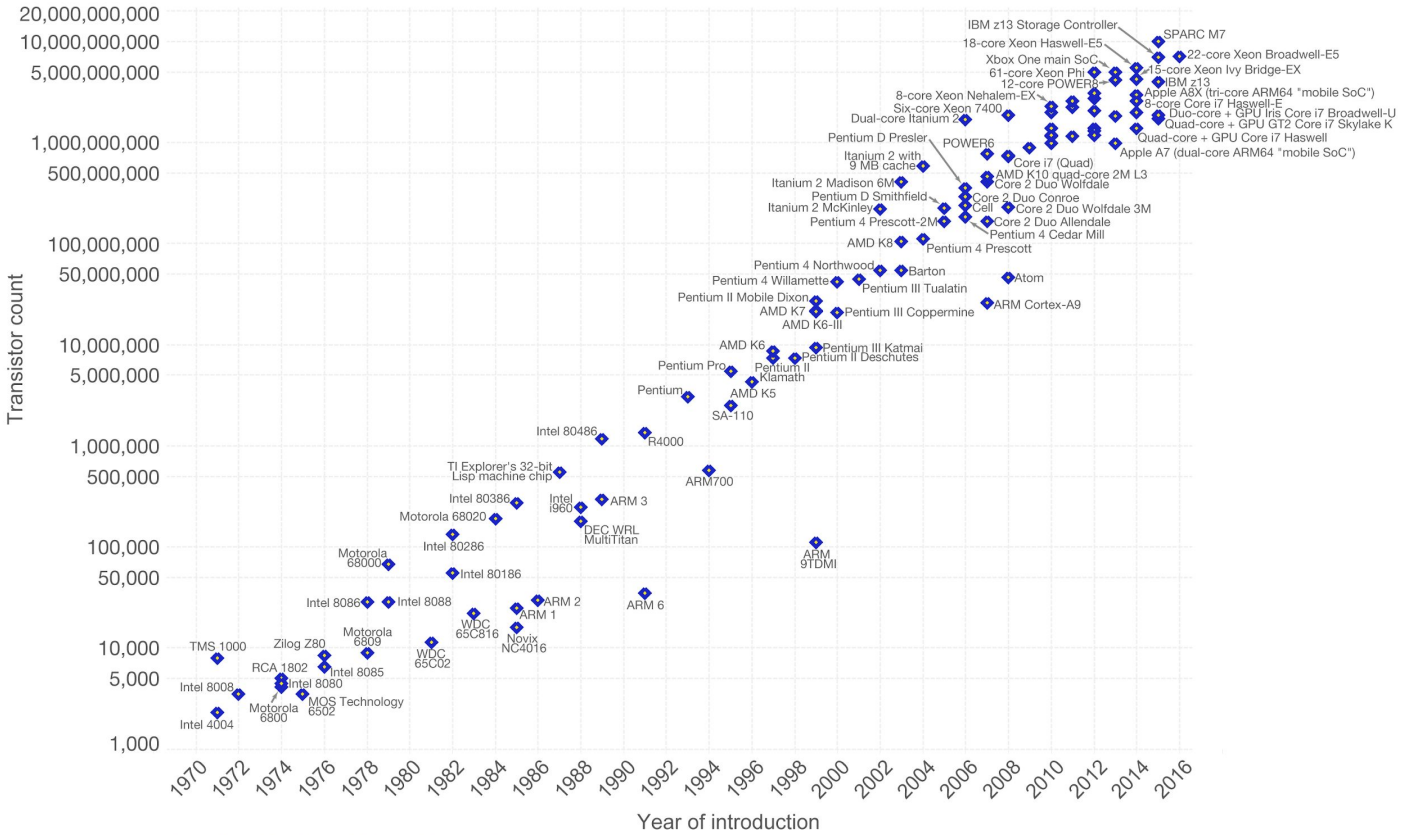
High Performance Computing

```
1940-1970:  the first supercomputers

1975-1990:  the Cray era

1990-2010:  the cluster era

2000-2020:  the GPU and hybrid era

2020-:      ???
```

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

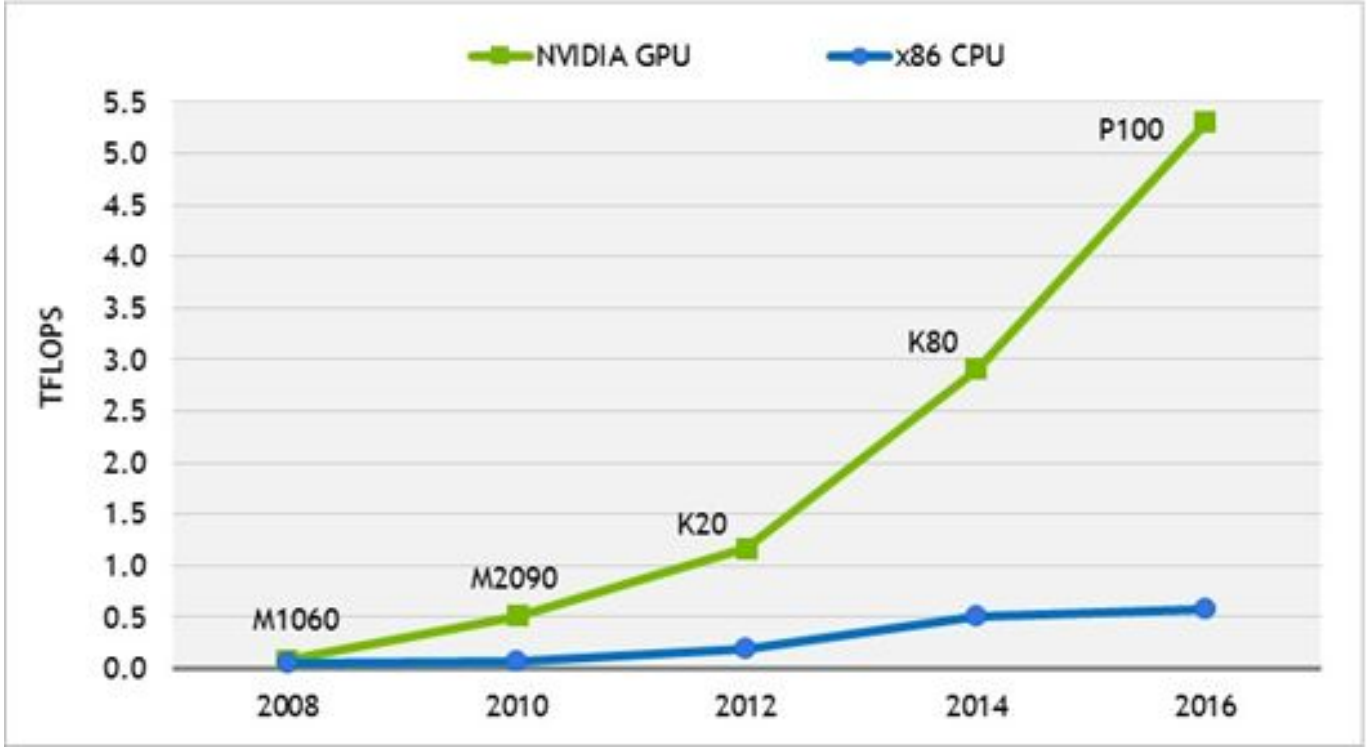
Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

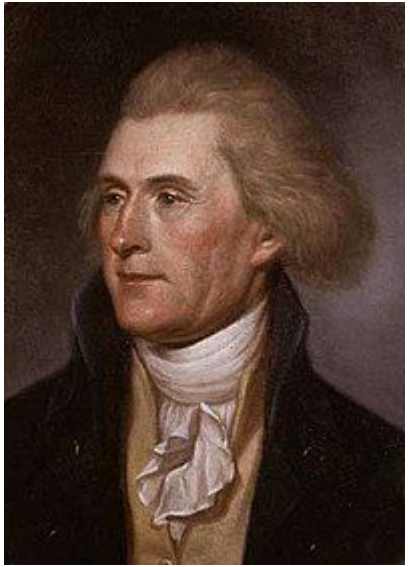
GPU evolution



Open Source

A business model of innovation

1790



X000001
July 31, 1790

The United States.

To all to whom these Presents shall come, Greeting.

Whereas Samuel Hopkins of the City of Philadelphia and State of Pennsylvania hath discovered an Improvement, not known or used before such Discovery, in the making of Pot ash and Pearl ash, by a new Apparatus and Process, that is to say, in the making of Pearl ash 1st by burning the raw Ashes in a Furnace, 2d by dissolving and boiling them when so burnt in Water, 3d by drawing off and settling the ley, and 4th by boiling the ley into Salts which then are the true Pearl ash; and also in the making of Pot ash, by fluxing the Pearl ash so made as aforesaid; which Operation of burning the raw Ashes in a Furnace, preparatory to their Dissolution and boiling in Water, is new, leaves little Residuum; and produces a much greater Quantity of Salt: These are therefore in pursuance of the Act, entitled "An Act to promote the Progress of useful Arts", to grant to the said Samuel Hopkins, his Heirs, Administrators and Assigns, for the Term of fourteen Years, the sole and exclusive Right and Liberty of using and vending to others the said Discovery, of burning the raw Ashes previous to their being dissolved and boiled in Water, according to the true Intent and Meaning of the Act aforesaid. In Testimony whereof I have caused these Letters to be made patent, and the Seal of the United States to be hereunto affixed. Given under my Hand at the City of New York this thirty first Day of July in the Year of our Lord one thousand seven hundred & Ninety.

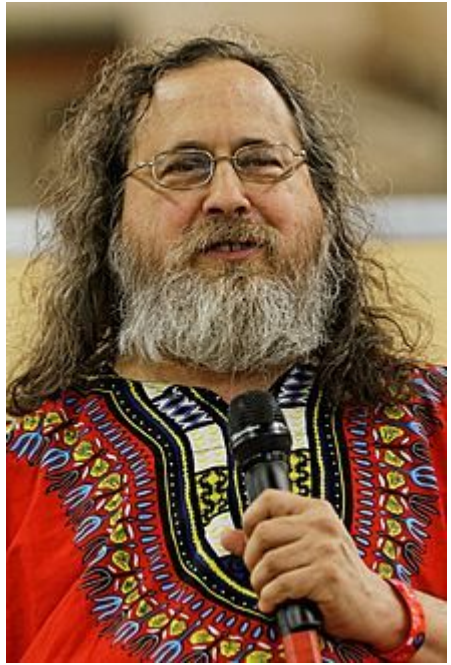
G Washington

City of New York July 31st 1790. –

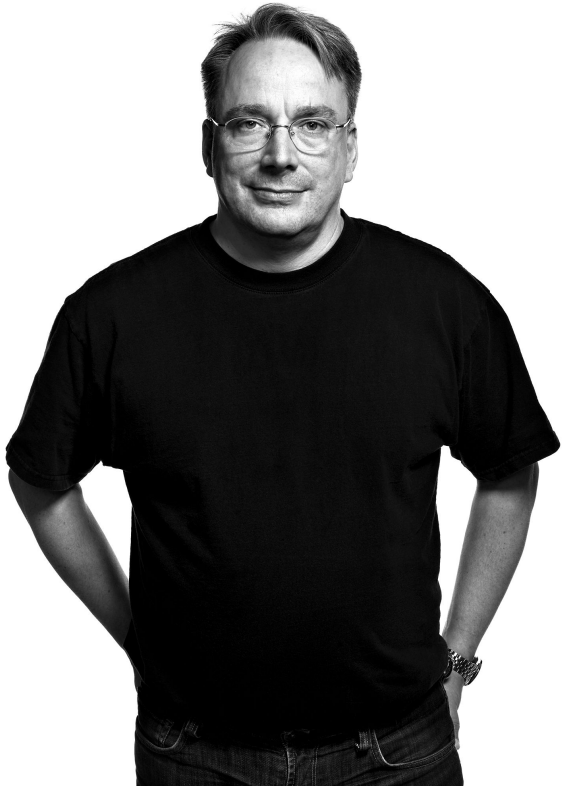
I do hereby certify that the foregoing Letters patent were delivered to me in pursuance of the Act, entitled "An Act to promote the Progress of useful Arts", that I have examined the same and find them conformable to the said Act.

Edm: Randolph Attorney General for the United States.

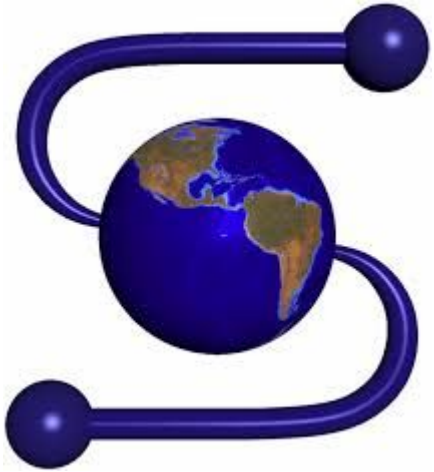
1984



1991



1993



1999

Key events in open source

1984 Richard Stallman (MIT) starts GNU project

1989 GPL

1991 Linus Torvalds releases Linux

1993 Mosaic browser
Red Hat founded

1994 Netscape
MySQL launched

1996 Apache launched

1997 Eric Raymond "The Cathedral and the Bazaar"

1998 Netscape open sources
Mozilla Firefox

1999 Apache Foundation
IBM announces $1 billion investment in Linux

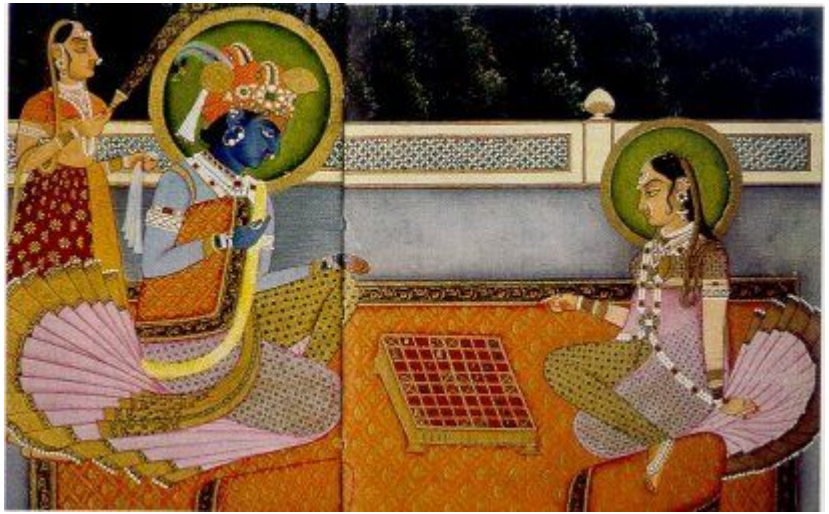
2006 Hadoop incubator

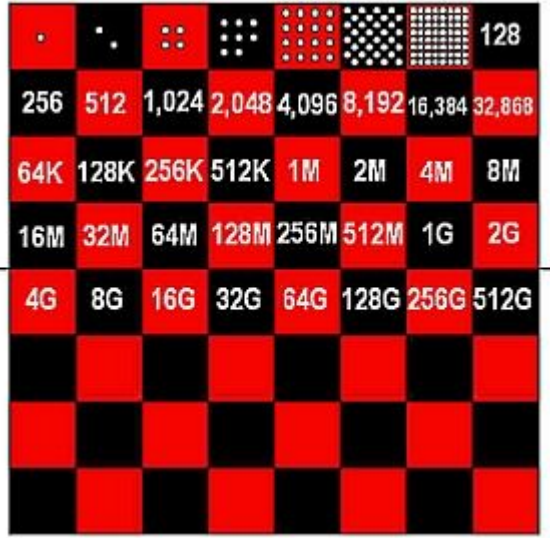
Hadoop

How distributed computing went mainstream

Original search

Legend of Paal Paysam





1997 Lucene

Doug Cutting builds full text search library

Analyze ordinary text with the purpose of building an index.

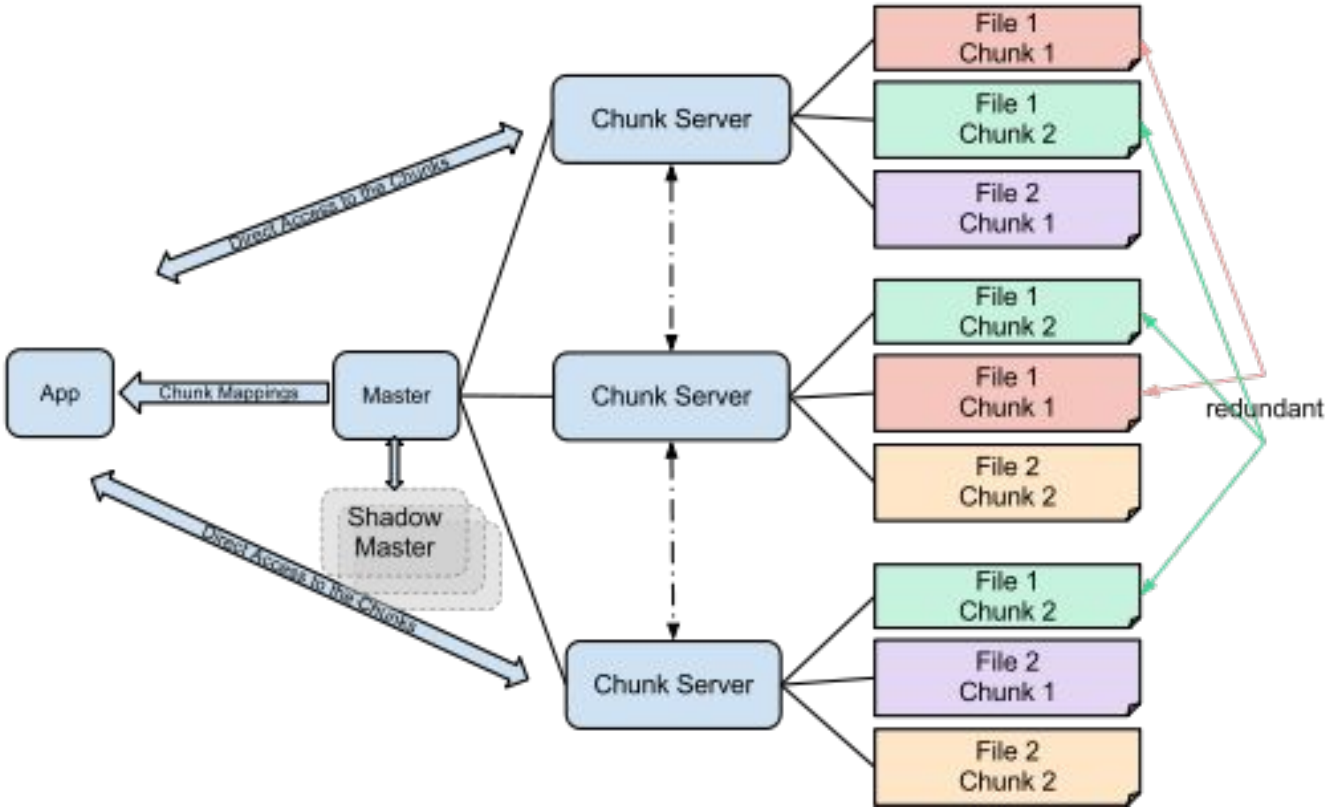
Index is a data structure that maps each term to its location in text, so that when you search for a term, it immediately knows all the places where that term occurs.

Add Nutch: a web crawler

2003 Google File System

- **schemaless** with no predefined structure, i.e. no rigid schema with tables and columns (and column types and sizes)
- **durable** once data is written it should never be lost
- capable of handling **component failure** without human intervention (e.g. CPU, disk, memory, network, power supply, MB)
- **automatically rebalanced** to even out disk space consumption throughout cluster

Google file system



2004 Google MapReduce

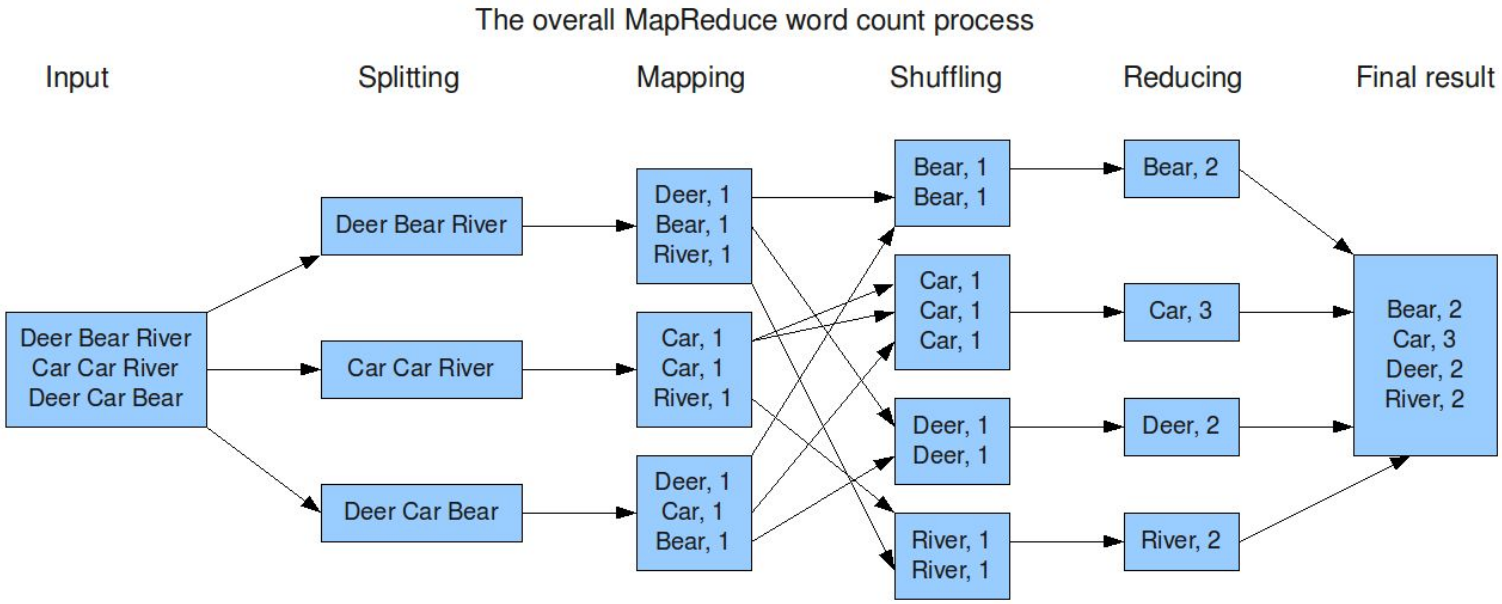
The three main problems that the MapReduce paper solved are:

1. Parallelization — how to parallelize the computation
2. Distribution — how to distribute the data
3. Fault-tolerance — how to handle component failure

A program is sent to where the data resides.

There’s simply too much data to be moved around.

MapReduce


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2

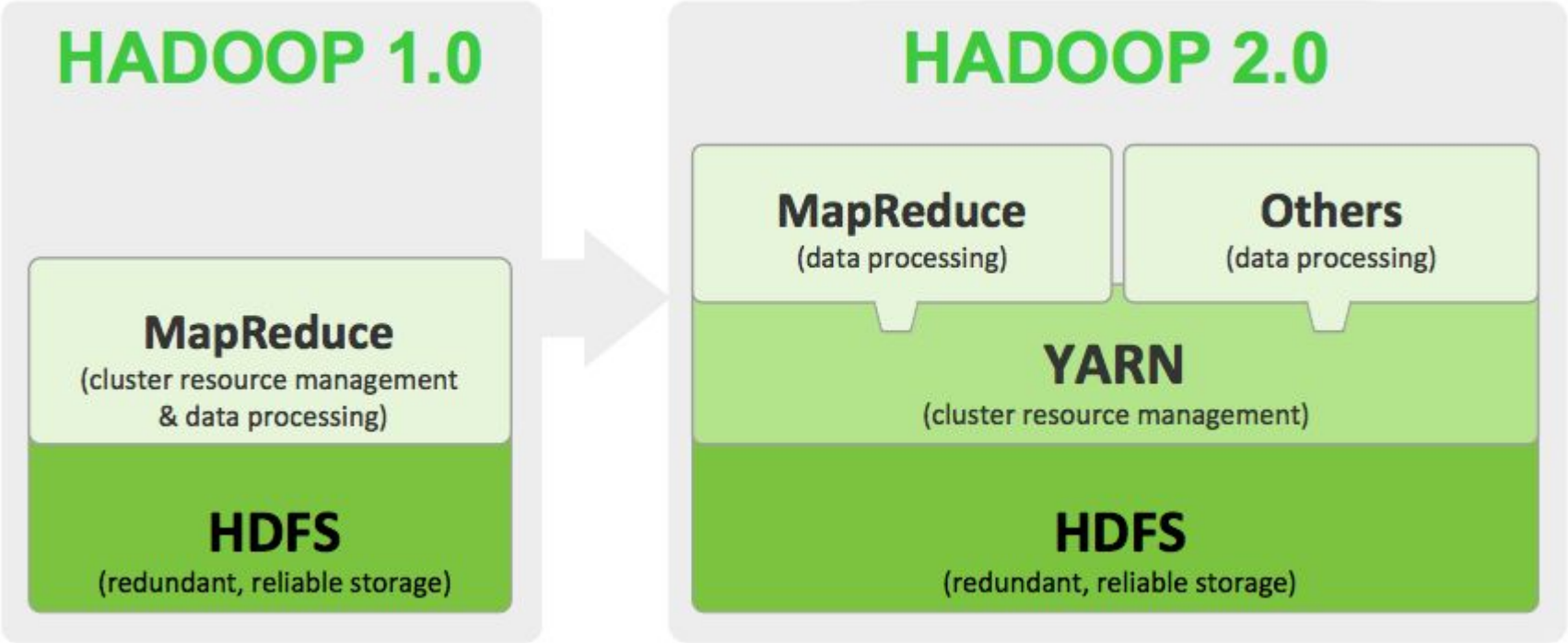
Pre Hadoop

1997 Lucene started by Doug Cutting
2000 Lucene released to Source Forge
2001 Lucene becomes Apache project http://lucene.apache.org/
2001 Cutting and Mike Cafarella start Apache Nutch
2003.10 Google File System published
2004 Nutch Distributed File System
2004.01 Scala released
2004.12 Map Reduce published https://ai.google/research/pubs/pub62
2005.06 MapReduce integrated into Nutch

Post Hadoop

2006.02 Hadoop incubator released
2006 Bigtable published https://ai.google/research/pubs/pub27898
2007.02 Yahoo reports 1000 node Hadoop cluster
2008.01 Hadoop becomes a top level Apache project
2008 HBase joins Hadoop
2008.05 ZooKeeper
2008.10 Pig (from Yahoo) and Hive (from Facebook)
2008 Cloudera founded
2012 Yahoo Hadoop cluster reaches 42,000 nodes
2012.08 YARN becomes subproject

2012 YARN


HADOOP 1.0
MapReduce
(cluster resource management
& data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)

Functional programming

What is old is new again

Programming paradigms

- Imperative
- Structured
- Procedural
- Object-oriented
- Event-driven
- Declarative
- Functional
- Reactive

Gottfried von Leibniz

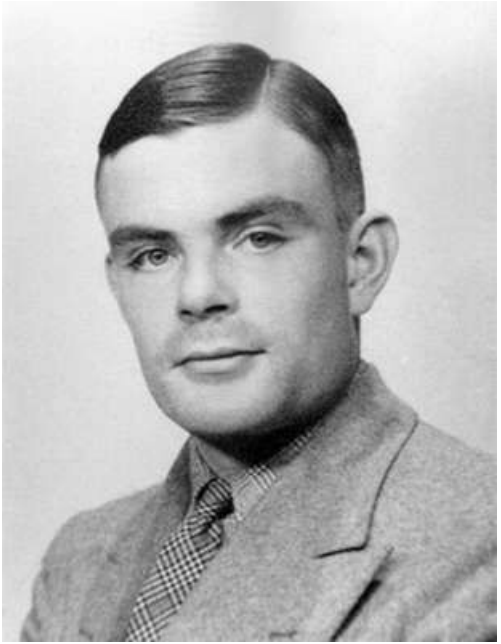
---

(1) Create a ‘universal language’ in which all possible problems can be stated.

(2) Find a decision method to solve all the problems stated in the universal language.

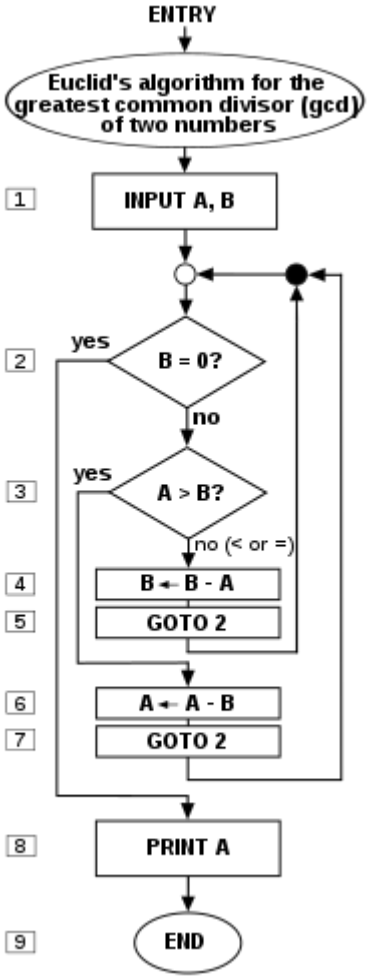
Entscheidungsproblem

1936 Church & Turing



Turning machine

Algorithms


ENTRY
Euclid's algorithm for the greatest common divisor (gcd) of two numbers
1
INPUT A, B
2
yes
B = 0?
no
3
yes
A > B?
no (< or =)
4
B ← B - A
5
GOTO 2
6
A ← A - B
7
GOTO 2
8
PRINT A
9
END

Lambda calculus

- A formal system in mathematical logic for expressing computation based on function abstraction and application using variable binding and substitution.
- The smallest universal programming language of the world. The λ calculus consists of a single transformation rule (variable substitution) and a single function definition scheme.

Lambda examples

```
first of (sort (append (‘dog’, ‘rabbit’) (sort ((‘mouse’, ‘cat’)))))
→ first of (sort (append (‘dog’, ‘rabbit’) (‘cat’, ‘mouse’)))
→ first of (sort (‘dog’, ‘rabbit’, ‘cat’, ‘mouse’))
→ first of (‘cat’, ‘dog’, ‘mouse’, ‘rabbit’)
→ ‘cat’.

(7 + 4) * (8 + 5 * 3)
→ 11 * (8 + 5 * 3)
→ 11 * (8 + 15)
→ 11 * 23
→ 253.
```

Lisp

```
(defun factorial (n)
  (if (= n 0) 1
      (* n (factorial (- n 1)))))
```

Functional programming languages

1932 Lambda calculus -- Alonzo Church

1958 LISP -- John McCarthy

1970 Scheme

1986 Erlang

1990 Haskell

1995 JavaScript

2004 Scala

2005 F#

2007 Clojure

Stream Programming

What, when, why, how?

A rose by any other name...

- Stream programming
- Stream processing
- Real-time analytics
- Streaming analytics
- Complex event processing (CEP)
- Real-time streaming analytics
- Event processing

Earlier CEP frameworks

- 2002Aurora
- 2005Borealis
- 2005 Apama
- 2007 Cayuga
- 2008 Esper
- 2011 Apache S4 (Yahoo)

What is CEP?

Trivial

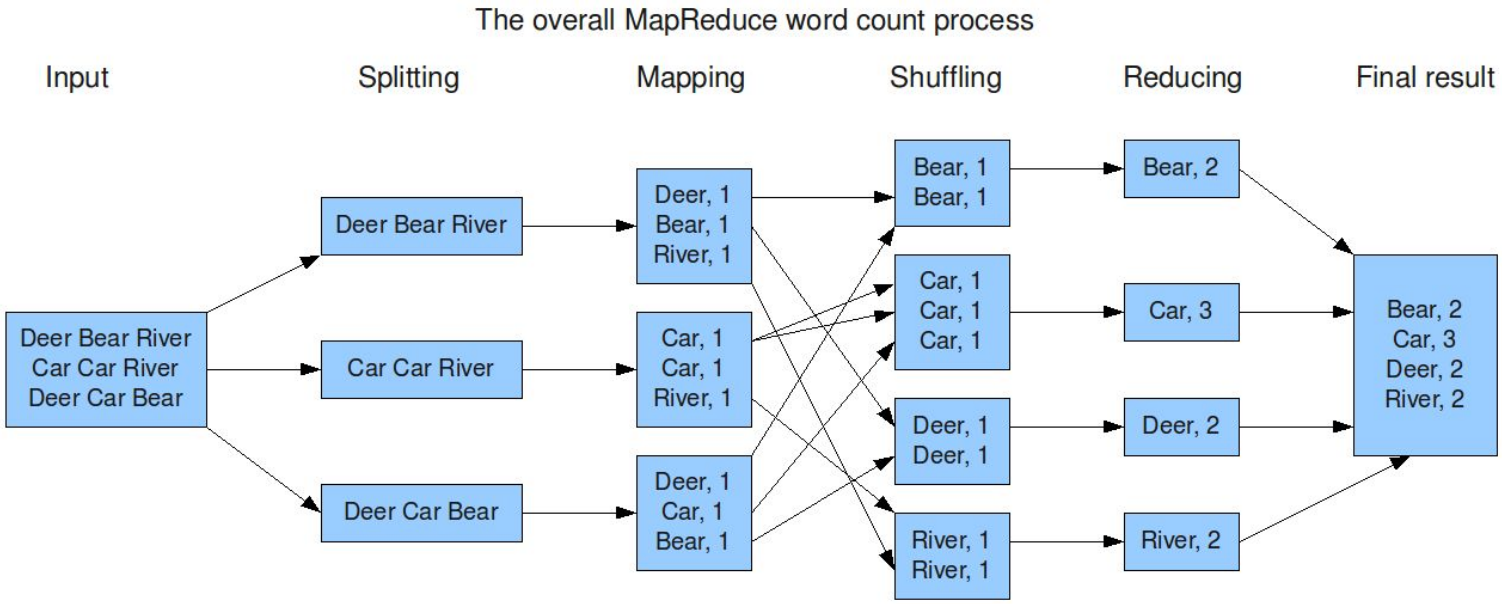
React to a button pushed

Complex

React to a button pushed 3 times in 10 seconds

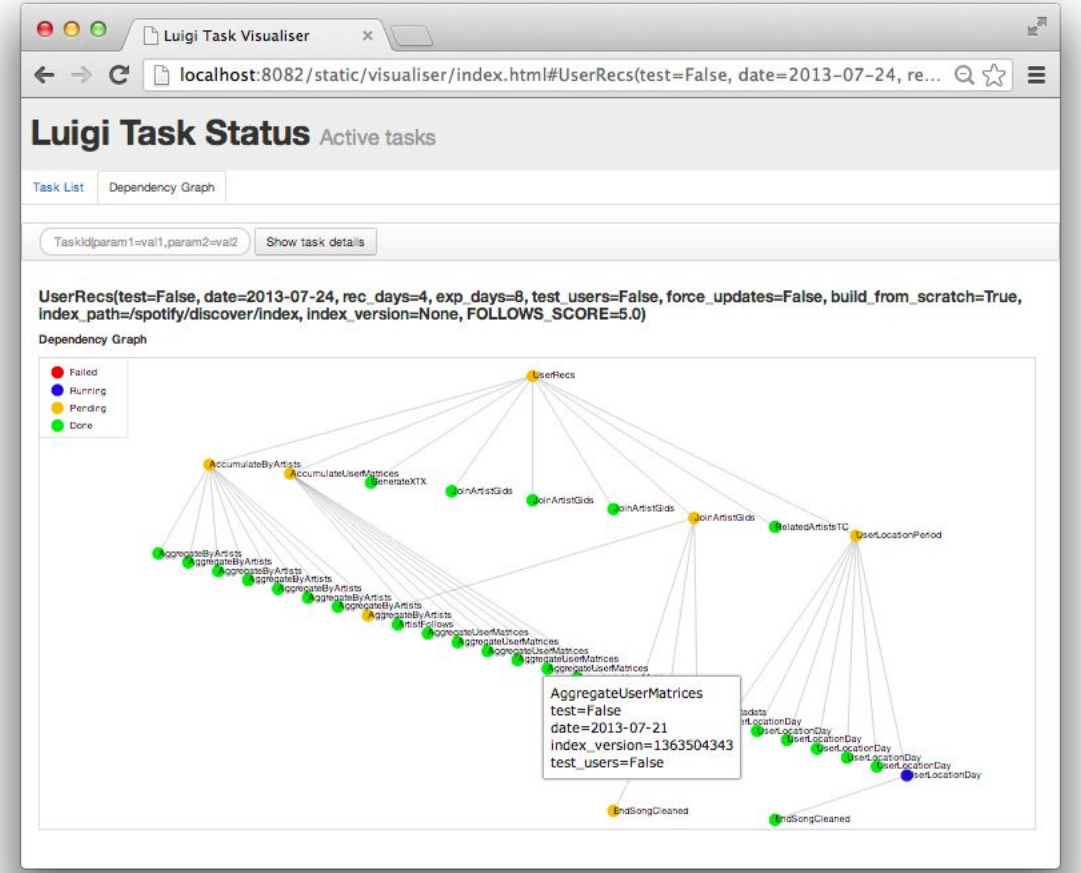
0001	Push	
0005	Push	
0012	Push	
0014	Push	ALERT
0021	Push	alert again?
0033	Push	

MapReduce

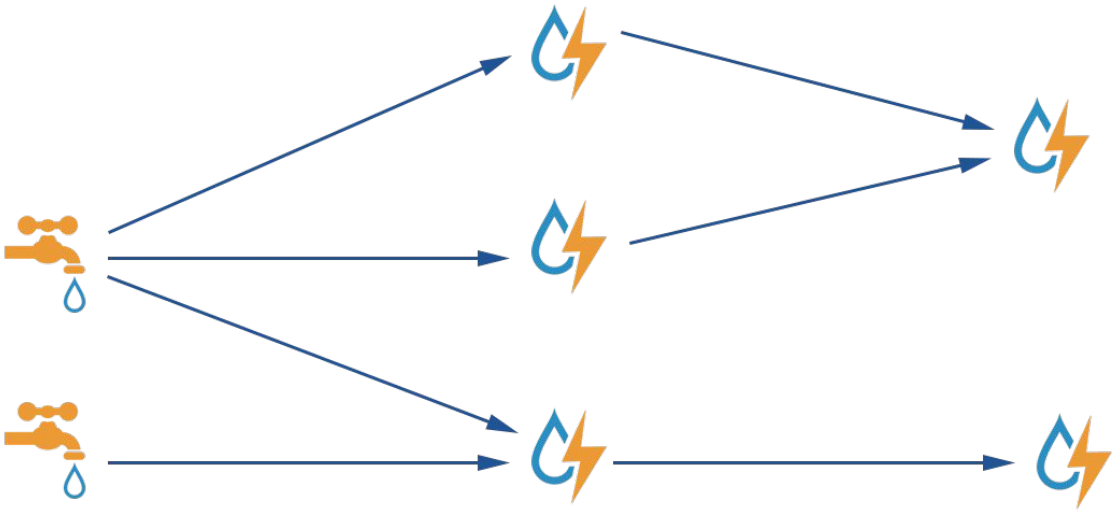

The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2

Workflows

Oozie
Luigi
Azkaban
Airflow
Pinball
Cascading
Taskflow



2011 Apache Storm



Streaming frameworks

2011Apache Storm

2014**Apache Spark**

Apache Samza

2015Apache Flink

Apache Nifi

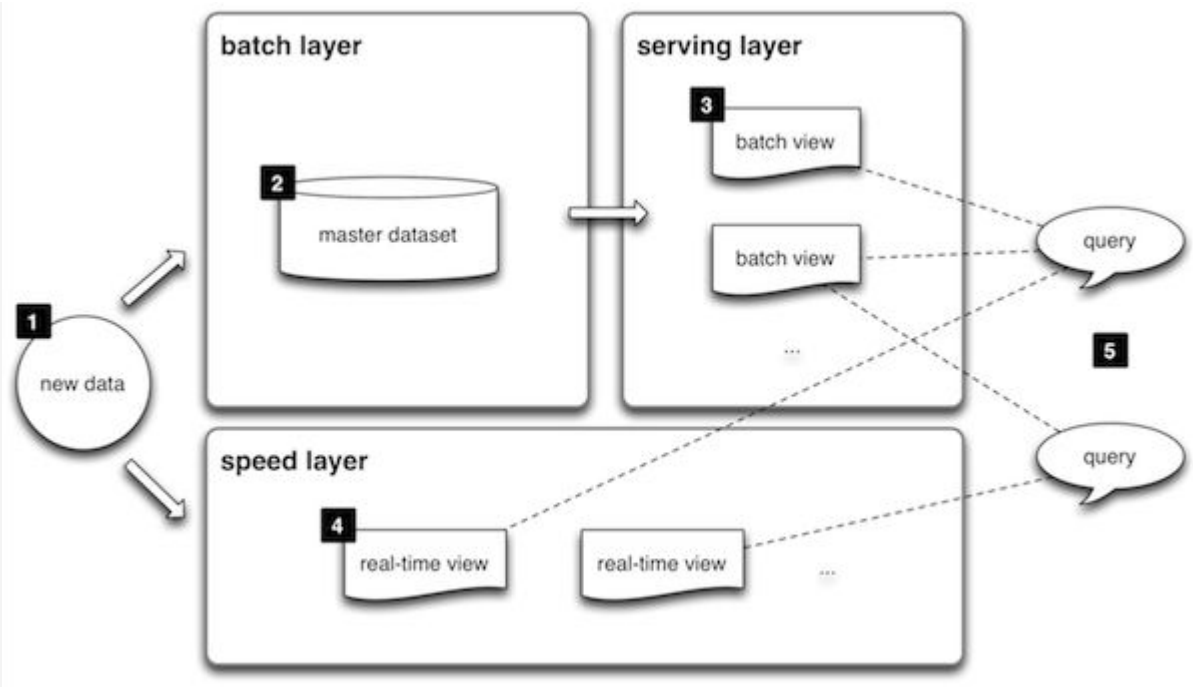
2016Apache Gearpump

Apache Apex

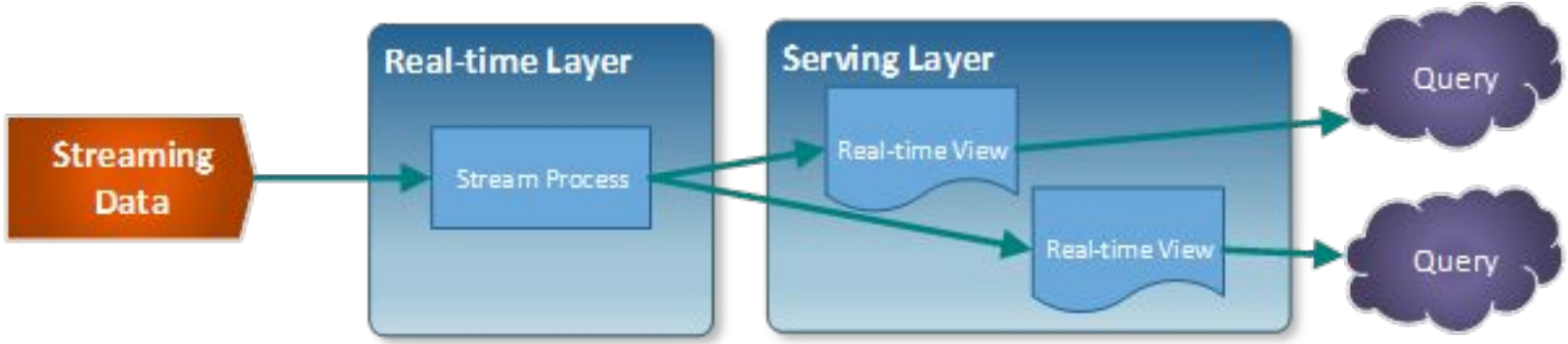
Kafka Streams

Akka Streams

2013 Lambda Architecture



2015 Kappa architecture

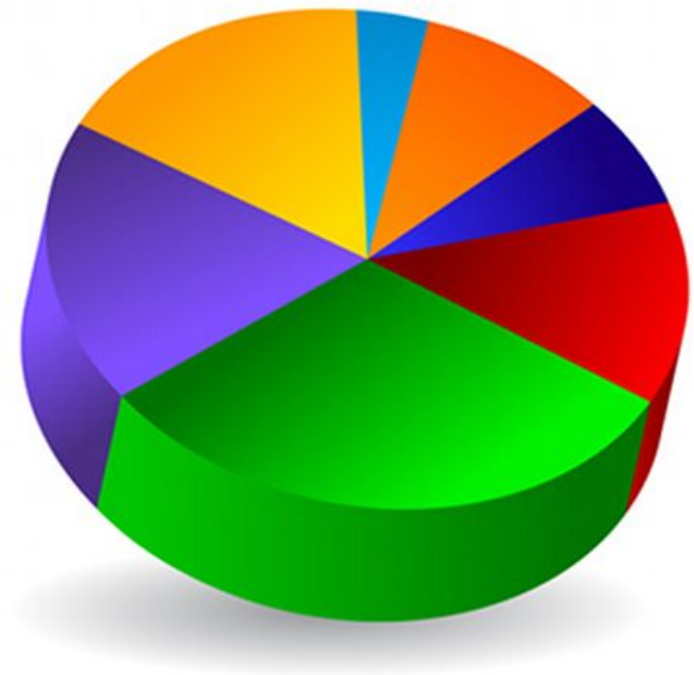


Why history?

The only constant is change

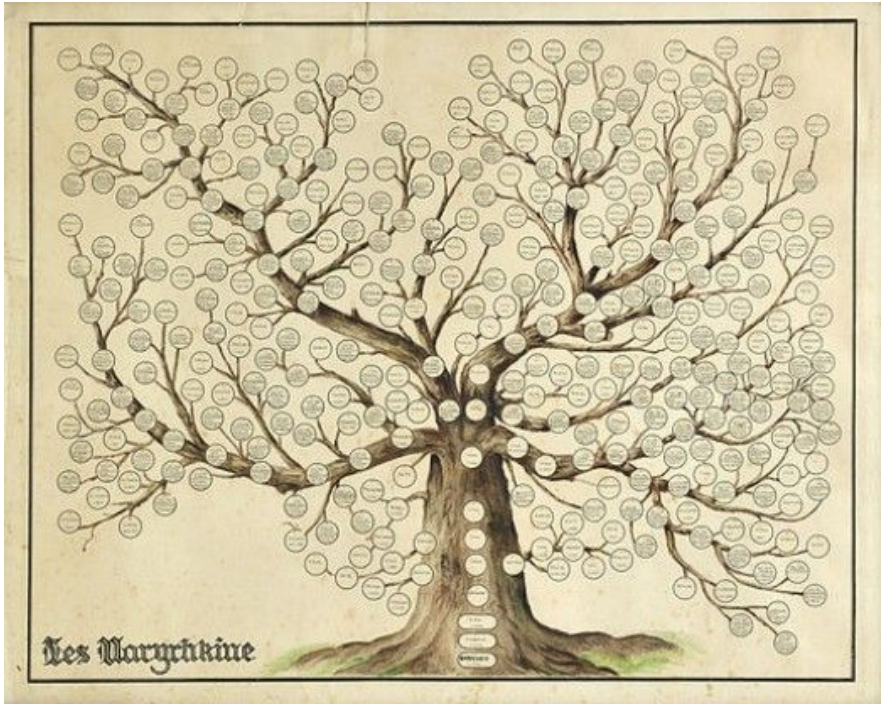
Pie metaphor

- Different technologies fill different roles, like slices of a pie
- Where slices meet neither solution is a perfect match
- The pie is always expanding
- The boundary between slices become large gaps
- New technologies arise to fill the gaps



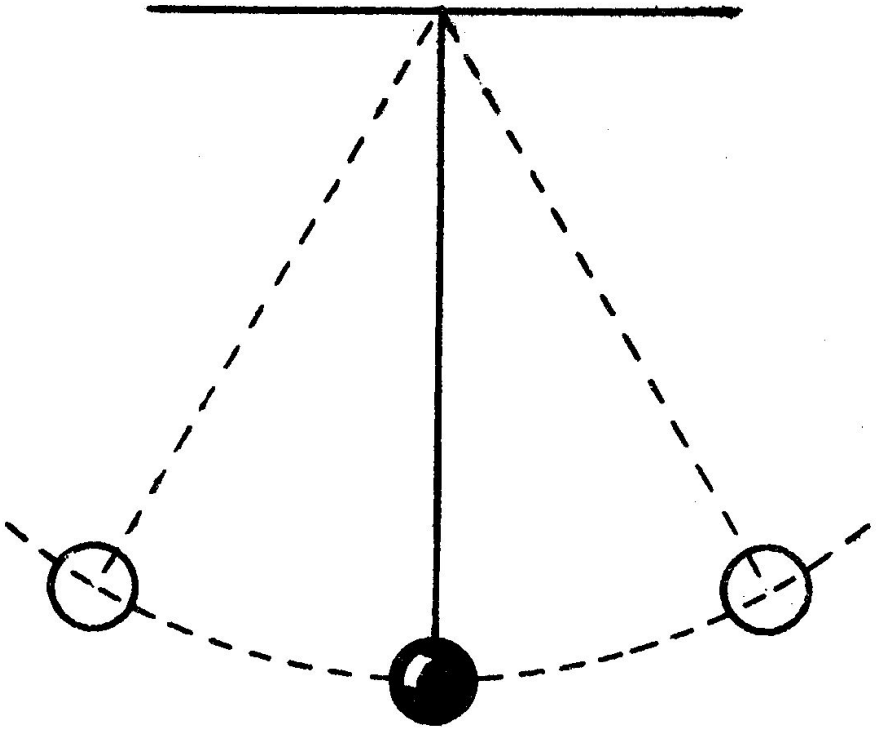
Tree metaphor

- There are many, many instances of the same thing
- Focus on root or branch technology before dealing with leaves
- What is old is new again

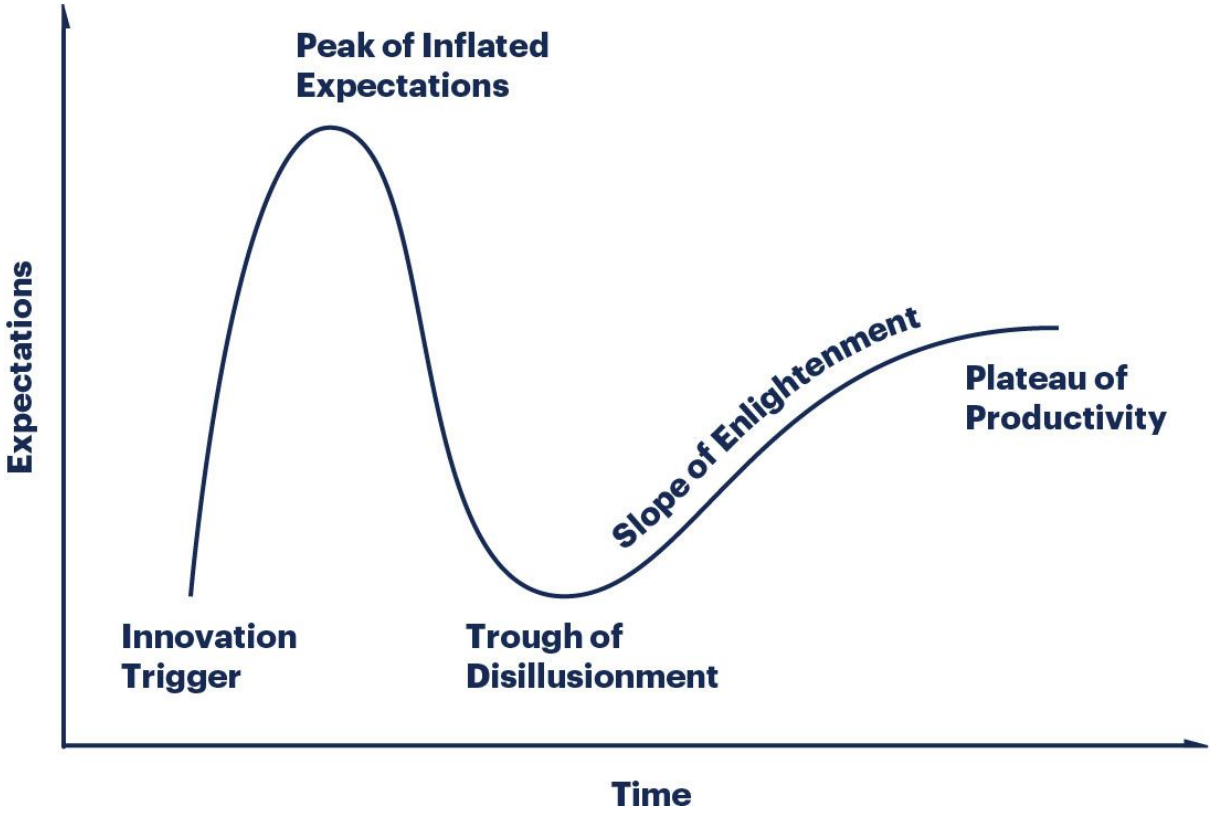


Pendulum metaphor

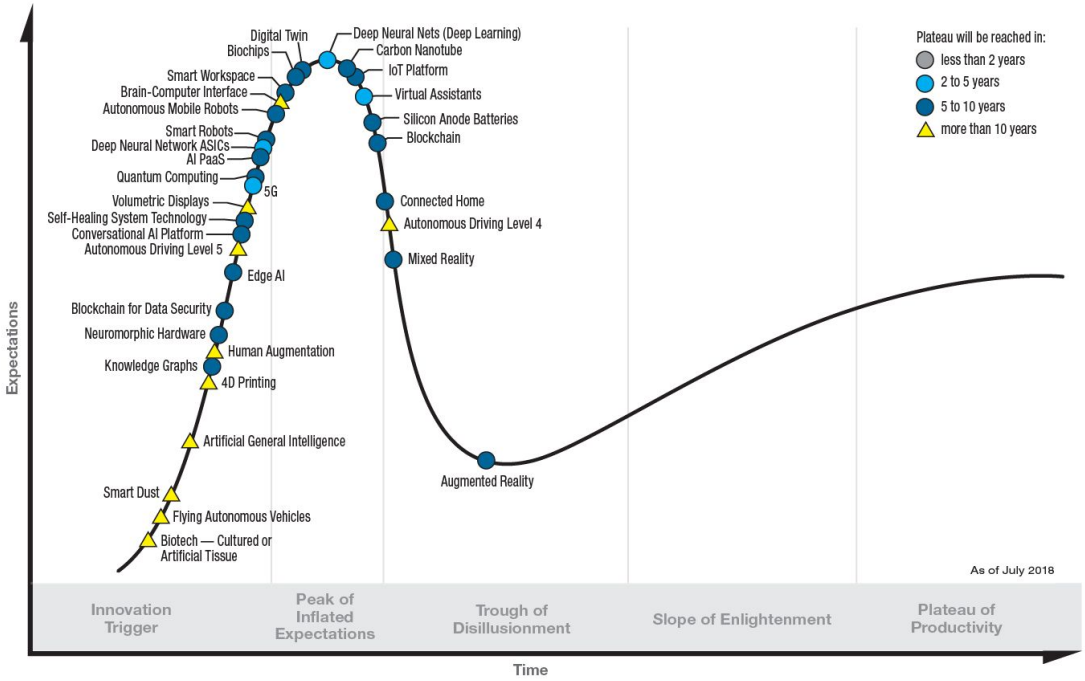
- Markets oscillate
- Centralized to decentralized
- Structured to unstructured
- Consolidated to fractured



The hype cycle



Hype Cycle for Emerging Technologies, 2018

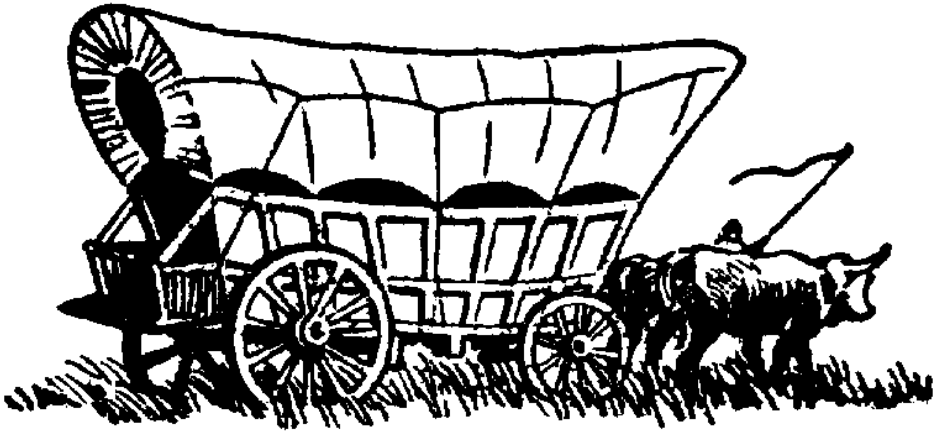


gartner.com/SmarterWithGartner

Source: Gartner (August 2018)

Gartner

Build wagons, not cabins



THANK YOU!