

CMPT 354: Database System I

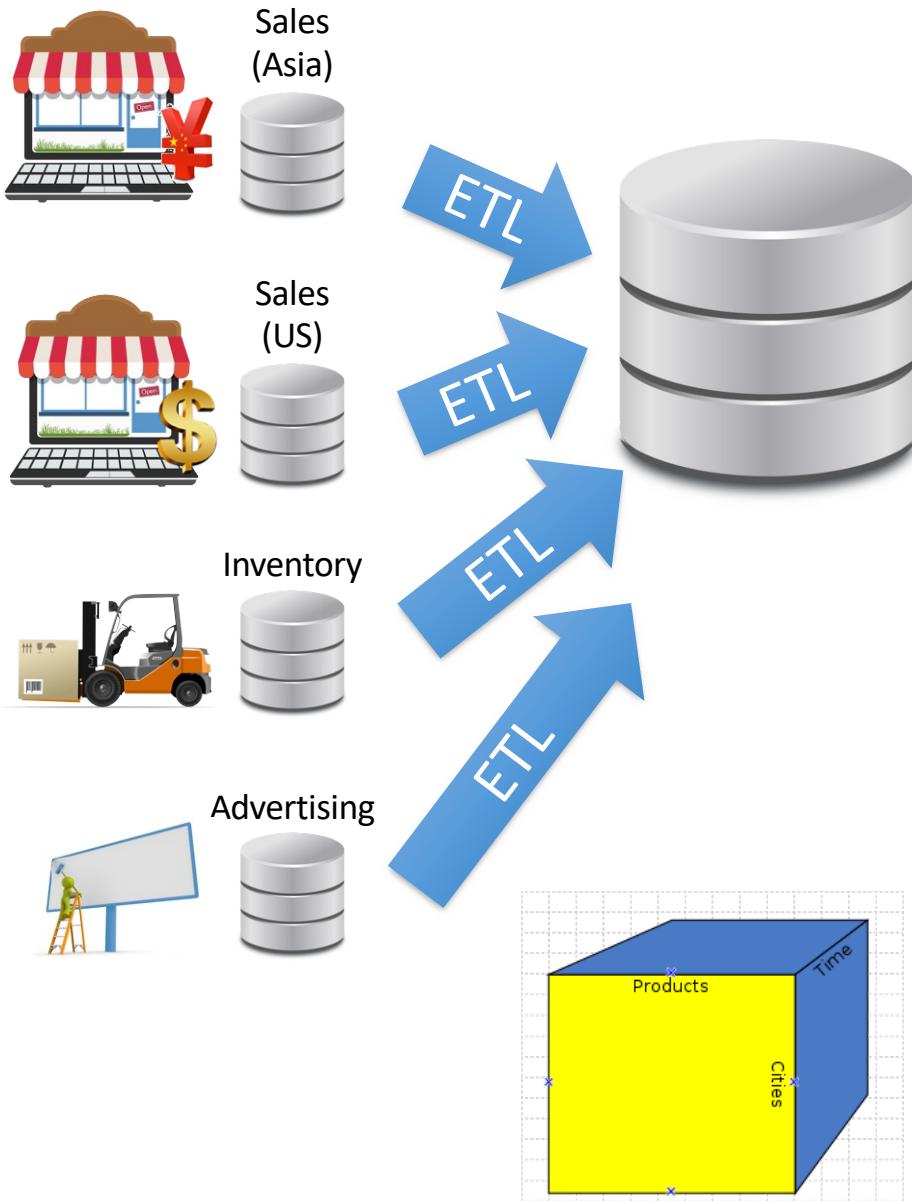
Lecture 12. Where to go from here

Where are we now

- **How to query a database**
 - SQL (Lec 3-4)
- **How to design a database**
 - ER Model (Lec 8)
 - Design Theory (Lec 9)
 - Relational Model (Lec 2)
- **How to develop a database application**
 - Database Application (Lec 10)
 - Transactions (Lec 12)
- **How to process a SQL query**
 - Relational Algebra (Lec 5)
 - Query Processing (Lec 6)
 - Query Optimization (Lec 12)

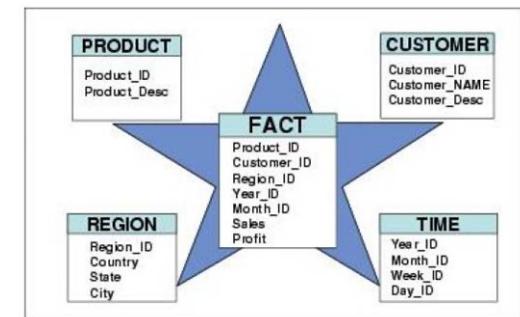
Where to go from here

- Data Warehouse
- Parallel Databases
- Big Data Processing
- Cloud Databases
- Data Science



Data Warehouse

Collects and organizes historical data from multiple sources



So far ...

- Star Schemas
- Data cubes



Data Warehouse

Collects and organizes historical data from multiple sources

- How do we deal with semi-structured and unstructured data?
- Do we really want to force a schema on load?

Data Warehouse

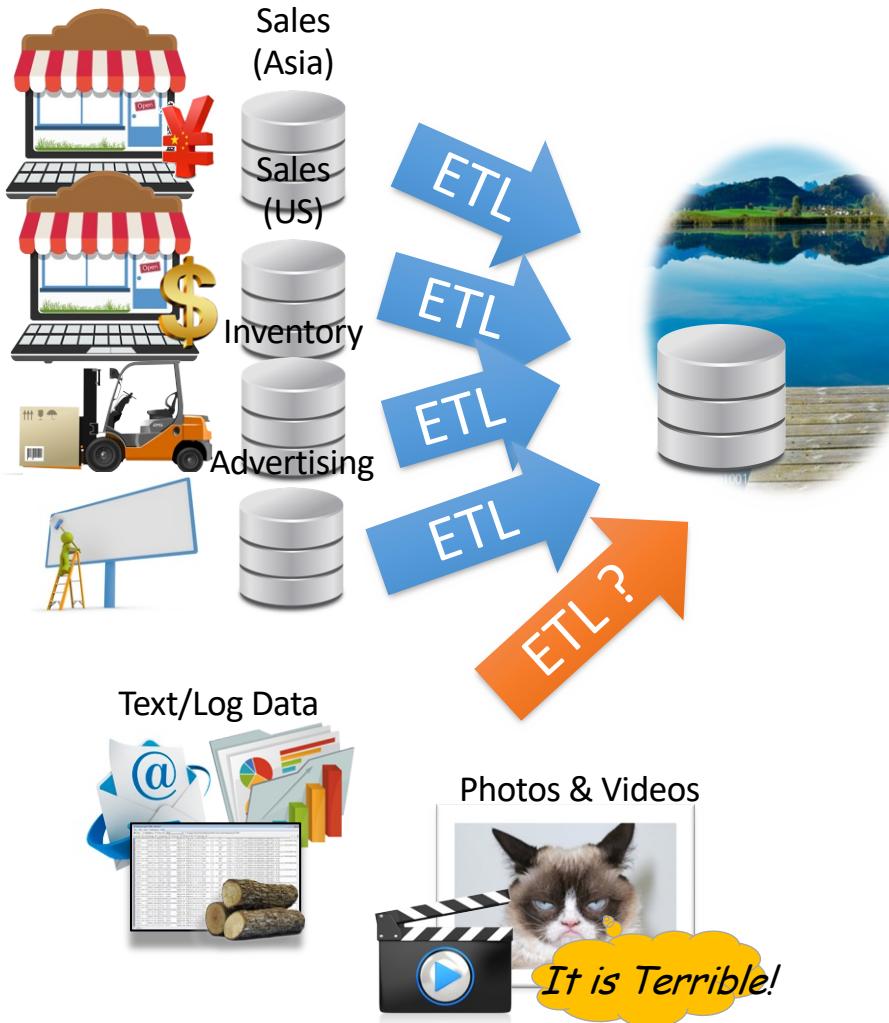


Collects and organizes historical data from multiple sources

How do we **load** and **process** this data in a relational system?

Do we read on load?

Depends on use ...
Can be difficult ...
Requires thought ...



Data Lake *

Store a copy of all the data

- in one place
- in its original “natural” form

Enable data consumers to choose how to transform and use data.

- *Schema on Read*

Enabled by new Tools:
Map-Reduce & Distributed Filesystems

What could go wrong?

* Still being defined...[Buzzword Disclaimer]

The Dark Side of Data Lakes



- Cultural shift: *Curate* → *Save Everything!*
 - Noise begins to dominate signal
- Limited data governance and planning
 - **Example:**
hdfs://important/joseph_big_file3.csv_with_json
 - **What** does it contain?
 - **When** and **who** created it?
 - No cleaning and verification → lots of dirty data
 - New tools are more complex and old tools no longer work

Enter the data engineer

A Brighter Future for Data Lakes

Enter the data engineer

- Data engineers bring new skills
 - Distributed data processing and cleaning
 - DevOps and MLOps
- Technologies are improving
 - SQL over large files
 - Self describing file formats & catalog managers
- Organizations are evolving
 - Tracking data usage and file permissions
 - New job title: data engineers



Where to go from here

- Data Warehouse
- Parallel Databases
- Big Data Processing
- Cloud Databases
- Data Science

Why compute in parallel?

- Multi-cores:
 - Most processors have multiple cores
 - This trend will likely increase in the future
- Big data: too large to fit in main memory
 - Distributed query processing on 100x-1000x servers
 - Widely available now using cloud services

Parallel DBMSs

- How to evaluate a parallel DBMS?
- How to architect a parallel DBMS?
- How to partition data in a parallel DBMS?

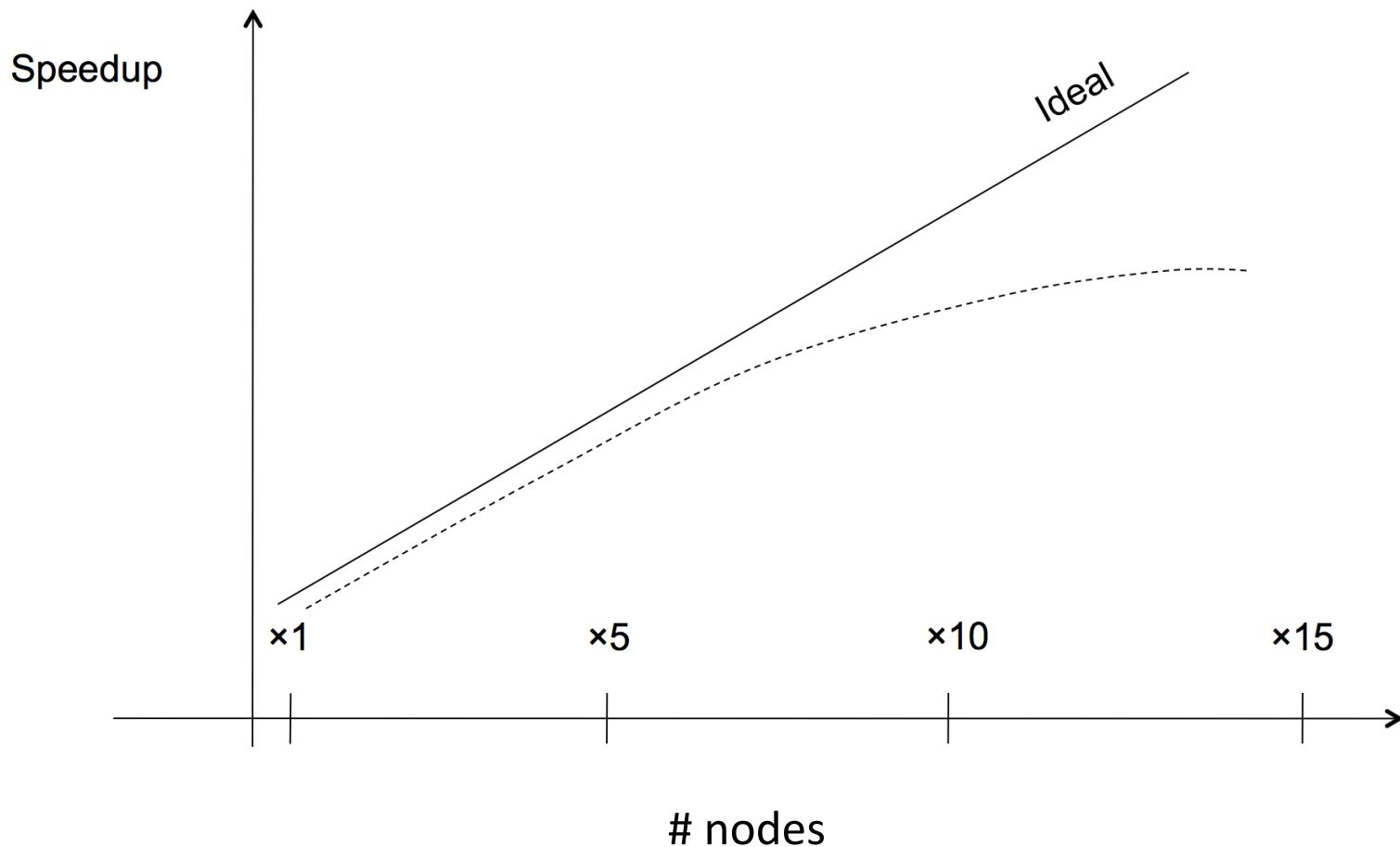
Parallel DBMSs

- **How to evaluate a parallel DBMS?**
- How to architect a parallel DBMS?
- How to partition data in a parallel DBMS?

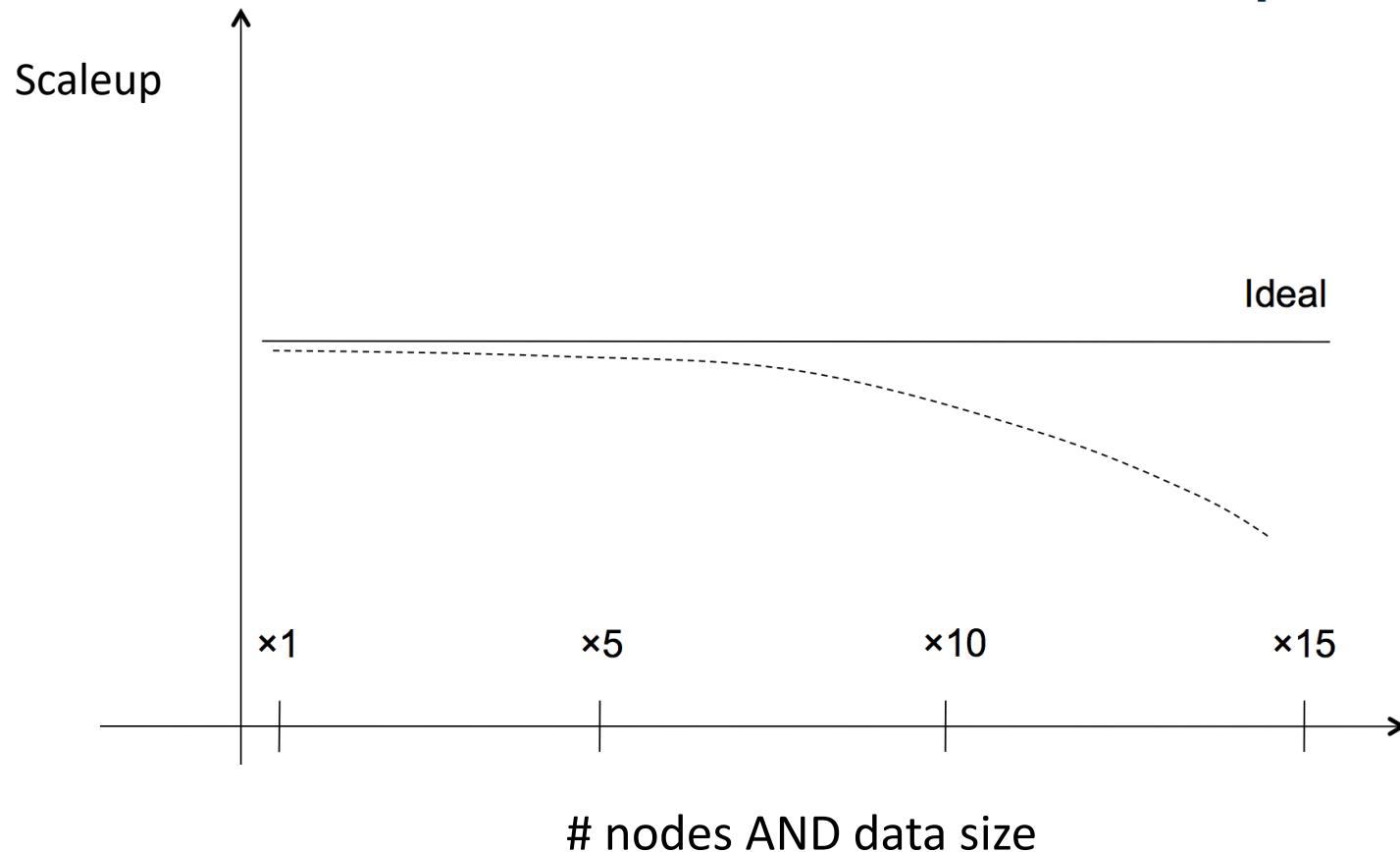
Performance Metrics for Parallel DBMSs

- Nodes = processors, compute
- **Speedup**
 - More nodes, same data → Higher speed
- **Scaleup**
 - More nodes, more data → same speed

Linear v.s. Non-linear Speedup



Linear v.s. Non-linear Scaleup



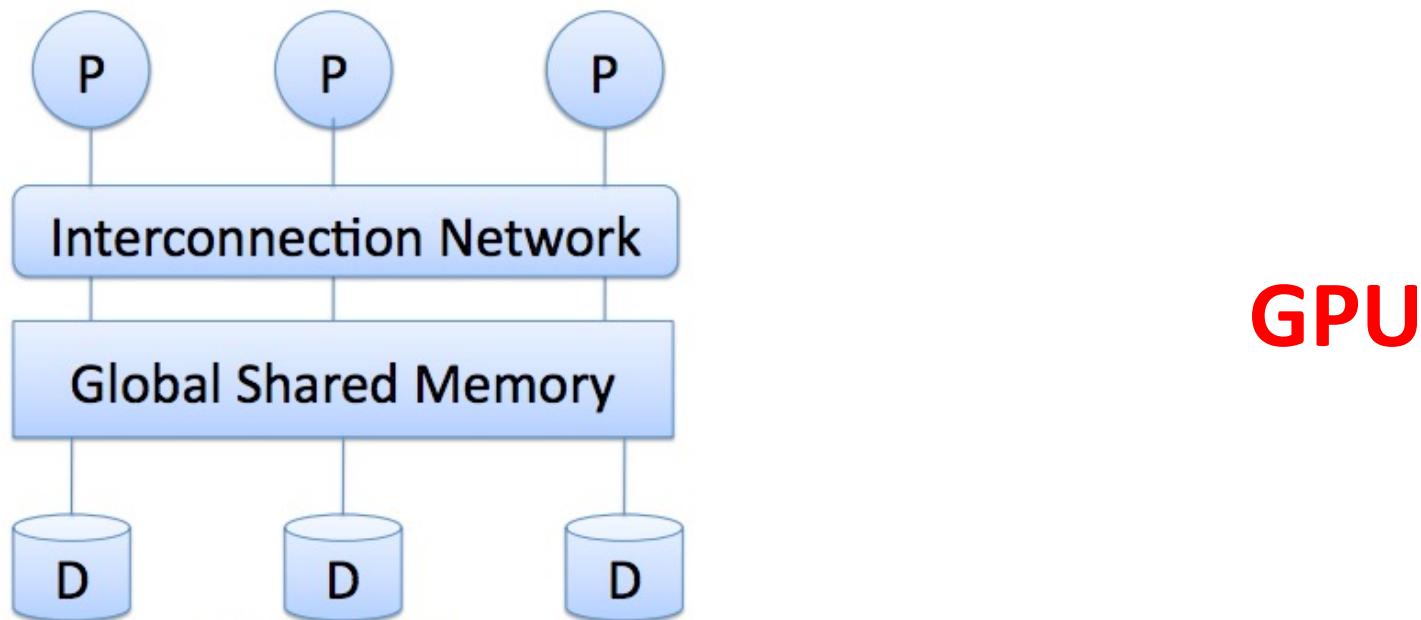
Parallel DBMSs

- How to evaluate a parallel DBMS?
- How to architect a parallel DBMS?
- How to partition data in a parallel DBMS?

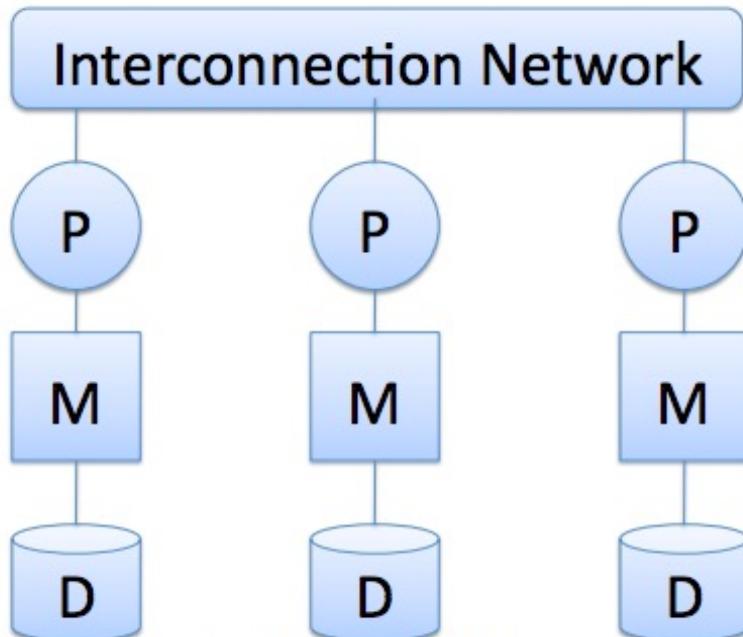
Three Architectures

- **Shared Memory**
- **Shared Nothing**
- **Shared Disk**

Shared Memory

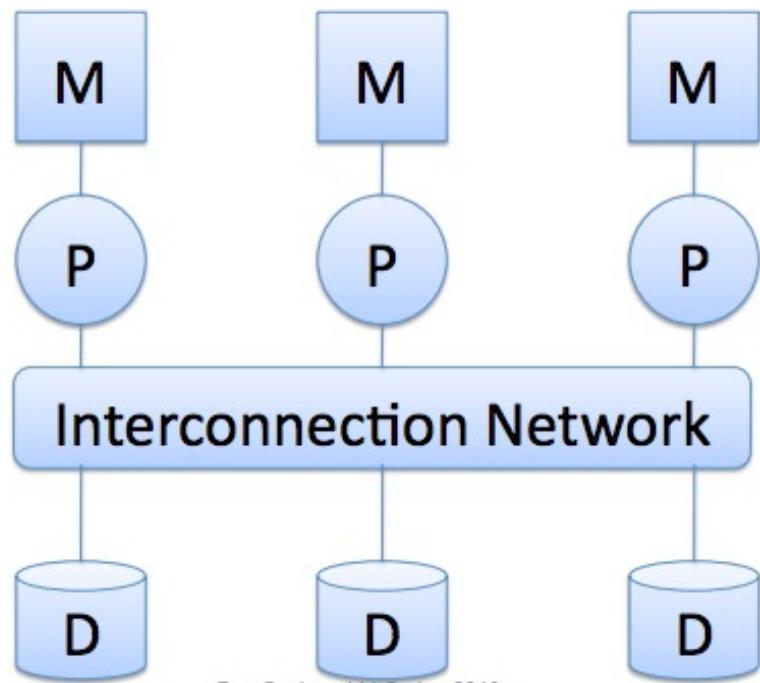


Shared Nothing



Parallel DBMSs, MapReduce,
Spark

Shared Disk



Azure Synapse Analytics

Three Architectures

- **Shared Memory**



Computation vs. Communication Trade-offs

- **Shared Nothing**



Economic Consideration

- **Shared Disk**

Parallel DBMSs

- How to evaluate a parallel DBMS?
- How to architect a parallel DBMS?
- How to partition data in a parallel DBMS?

Horizontal Data Partitioning

- **Round Robin**

- ☺ Load Balancing
- ☹ Bad Query Performance

- **Range Partitioning**

- ☺ Good for range/point queries
- ☹ Data Skew (i.e., Bad Load balancing)

- **Hash Partitioning**

- ☺ Load Balancing, Good for point queries
- ☹ Hard to answer range queries

Where to go from here

- Data Warehouse
- Parallel Databases
- **Big Data Processing**
- Cloud Databases
- Data Science

3 Vs of Big Data

- Volume: data size
- Velocity: rate of data coming in
- Variety: data sources, formats, workloads

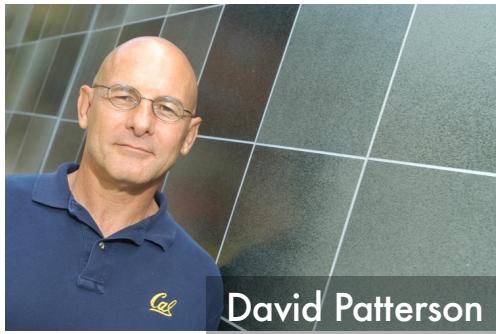
Big Data Systems



Shared Nothing Architecture

How was Spark created?

UC Berkeley's Research Centers



David Patterson

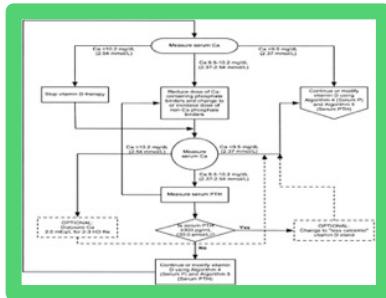
Requirements

- A common vision
- About 5 years
- At least three faculty
- A dozen students

Years	Title	Profs: Director, Co-PIs	Students
1977–1981	X-Tree: Tree Multiprocessor	Despain, Patterson, Sequin	12
1980–1984	RISC: Reduced Instructions	Patterson, Ousterhout, Sequin	17
1983–1986	SOAR: Smalltalk On A RISC	Patterson, Ousterhout	22
1985–1989	SPUR: Symbolic Processing Using RISCs	Patterson, Fateman, Hiltzinger, Hodges, Katz, Ousterhout	21
1988–1992	RAID: Redundant Array of Inexpensive Disks	Katz, Ousterhout, Patterson, Stonebraker	16
1993–1998	NOW: Network of Workstations	Culler, Anderson, Brewer, Patterson	25
1997–2002	IRAM: Intelligent RAM	Patterson, Kubiatowicz, Wawrzynek, Yelick	12
2001–2005	ROC: Recovery Oriented Computing Systems	Patterson, Fox	11
2005–2011	RAD Lab: Reliable Adaptive Distributed Computing Lab	Patterson, Fox, Jordan, Joseph, Katz, Shenker, Stoica	30
2007–2013	Par Lab: Parallel Computing Lab	Patterson, Asanovic, Demmel, Fox, Keutzer, Kubiatowicz, Sen, Yelick	40
2011–2017	AMP Lab: Algorithms, Machines, and People	Franklin, Jordan, Joseph, Katz, Patterson, Recht, Shenker, Stoica	40
2013–2018	ASPIRE Lab	Asanovic, Alon, Bachrach, Demmel, Fox, Keutzer, Nikolic, Patterson, Sen, Wawrzynek	40

AMPLab's Vision

Make sense of BIG DATA by tightly integrating
algorithms, machines, and people



+

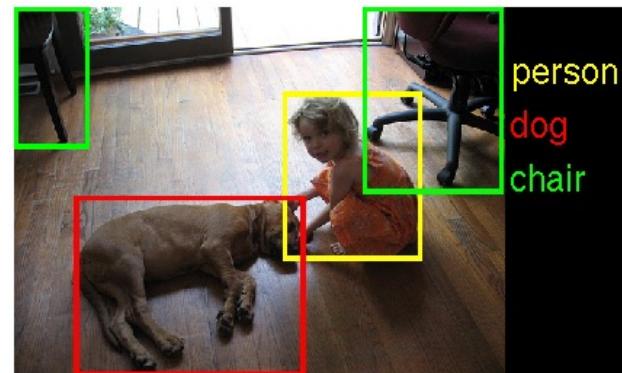


+



Example: Extract Value From Image Data

What are in the image?



How to solve the problem?

Deep Learning (Algorithms)
GPU Cluster (Machines)
ImageNet (People)

Spark's Initial Idea

- Algorithms + Machines
 - Run ML Algorithms on Hadoop
- Why is it slow?
 1. The algorithms are iterative (i.e., multiple scans of data)
 2. MapReduce writes/reads data to/from **disk** at each iteration
- Solution
 - Keep data in **memory**



Matei Zaharia

How About Fault Tolerance?

Resilient Distributed Datasets (RDD)



Main Idea: Logging the transformations (used to build an RDD) rather than the RDD itself

Why Spark?

Fast

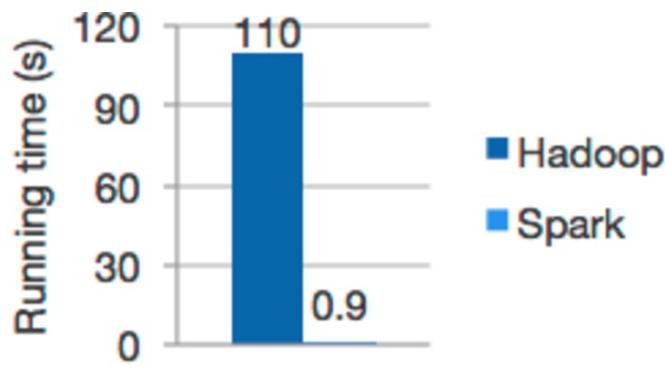


Easy to Use



What Makes Spark *Fast*?

- In-memory Computation



Logistic regression in Hadoop and Spark

What you save?

- Serialization/Deserialization
- Compression/Decompression
- I/O cost



“CPU (and not I/O) is often the bottleneck”

Ousterhout et al. Making Sense of Performance in Data Analytics Frameworks. NSDI 2015: 293-307

Hardware Trends

	2010	2016	
Storage	50+MB/s (HDD)	500+MB/s (SSD)	10X
Network	1Gbps	10Gbps	10X
CPU	~3GHz	~3GHz	

What Makes Spark *Fast*?

Project Tungsten: Bringing Apache Spark Closer to Bare Metal



by Reynold Xin and Josh Rosen

Posted in **ENGINEERING BLOG** | April 28, 2015

1. Memory Management and Binary Processing
2. Cache-aware computation
3. Code generation

Why Spark?

Fast



Easy to Use



What Makes Spark Easy-to-Use?

- Over 80 High-level Operators

WordCount (Mapreduce)

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WordCount extends Configured implements Tool {

    public static class TokenizerMapper
        extends Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        @Override
        public void map(LongWritable key, Text value, Context context
            ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class IntSumReducer
        extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        @Override
        public void reduce(Text key, Iterable<IntWritable> values,
                           Context context
                           ) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }

        public static void main(String[] args) throws Exception {
            int res = ToolRunner.run(new Configuration(), new WordCount(), args);
            System.exit(res);
        }

        @Override
        public int run(String[] args) throws Exception {
            Configuration conf = this.getConf();
            Job job = Job.getInstance(conf, "word count");
            job.setJarByClass(WordCount.class);
            job.setInputFormatClass(TextInputFormat.class);
            job.setMapperClass(TokenizerMapper.class);
            job.setCombinerClass(IntSumReducer.class);
            job.setReducerClass(IntSumReducer.class);

            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(IntWritable.class);
            job.setOutputFormatClass(TextOutputFormat.class);
            TextInputFormat.addInputPath(job, new Path(args[0]));
            TextOutputFormat.setOutputPath(job, new Path(args[1]));

            return job.waitForCompletion(true) ? 0 : 1;
        }
    }
}
```

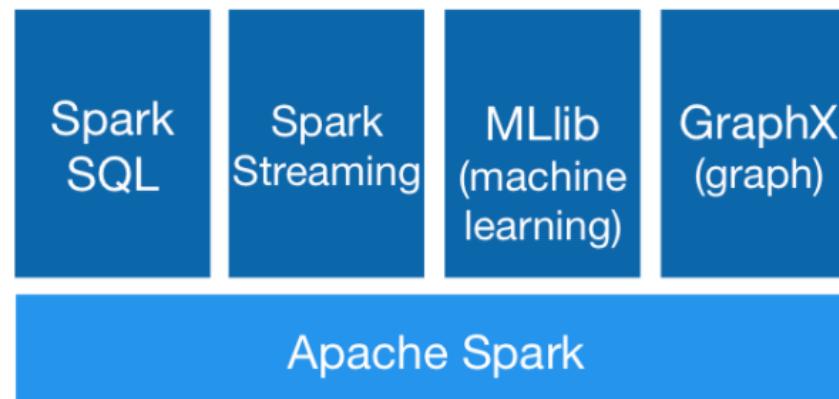
WordCount (Spark)

```
text_file = spark.textFile("hdfs://...")

text_file.flatMap(lambda line: line.split())
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a+b)
```

What Makes Spark *Easy-to-Use*?

- **Unified Engine**



Easy to manage, learn, and combine functionality

Analogy



Specialized Devices



Unified Device

What Makes Spark *Easy-to-Use*?

- Integrate Broadly

Languages:



The Big Data world is diversified



The Apache Spark logo features the word 'Spark' in a large, grey, sans-serif font. Above the letter 'S', there is a small orange star-like spark icon. Below the word 'Spark', there is a small trademark symbol (TM).

Data Sources:



Environments:



Where to go from here

- Data Warehouse
- Parallel Databases
- Big Data Processing
- Cloud Databases
- Data Science

Amazon

- From Wikipedia 2006

Article | Talk | Head

Amazon.com

From Wikipedia, the free encyclopedia

This is an old revision of this page
05:10, 20 March 2006 ([→Customer permanent link](#) to this revision, wh revision.

(diff) ← Previous revision | Latest revisi

Amazon.com

(NASDAQ: AMZN) is an American electronic commerce company based in Seattle, Washington. It was one of the

- From Wikipedia 2016

Amazon.com

From Wikipedia, the free encyclopedia

Further information: [Timeline of Amazon](#)

Amazon.com, Inc. (/əməzən/ or /əməzən/), often referred to as simply **Amazon**, is an American electronic commerce and cloud computing company with headquarters in Seattle, Washington. It is the

What is Cloud Computing?

- The buzz word before “Big Data”
 - Larry Ellison’s response in 2009 (<https://youtu.be/UOEFXaWHppE?t=7s>)
 - Berkeley RADLab’s paper in 2009 (<https://youtu.be/IJCxqoh5ep4>)
- A technical point of view
 - Internet-based computing (i.e., computers attached to network)
- A business-model point of view
 - Pay-as-you-go (i.e., rental)



Three Types of Cloud Computing

CourSys

Application + Cloud = **SaaS** (Software as a service)

Database

Platform + Cloud = **PaaS** (Platform as a service)

Servers

Infrastructure + Cloud = **IaaS** (Infrastructure as a service)

Cloud Databases

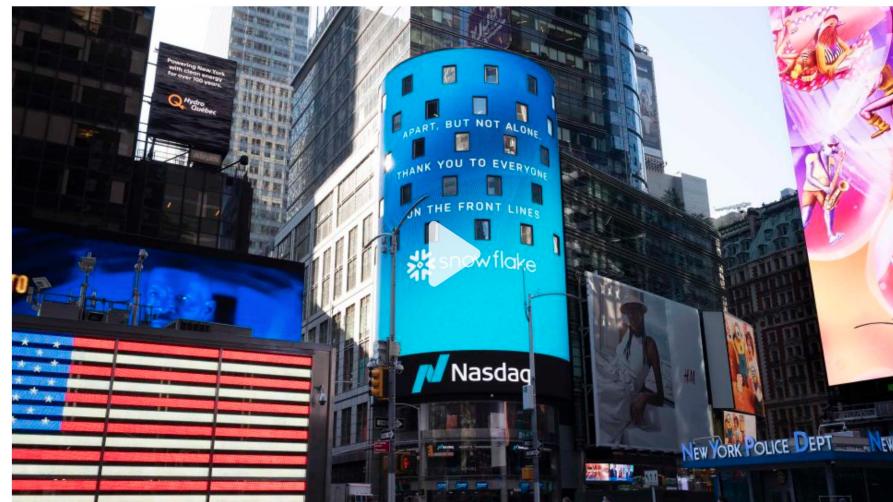
- Amazon
 - Aurora
 - Redshift
- Microsoft
 - Azure SQL Databases
 - Azure Synapse Analytics
- Google
 - Spanner
 - BigQuery

Snowflake shares more than double. It's the biggest software IPO ever



By [Paul R. La Monica](#), CNN Business

Updated 3:37 AM EDT, Thu September 17, 2020



Where to go from here

- Data Warehouse
- Parallel Databases
- Big Data Processing
- Cloud Databases
- **Data Science**

Computer Science vs. Data Science

What	When	Who	Goal
Computer Science	1950-	Software Engineer	Write software to make computers work

Plan → Design → Develop → Test → Deploy → Maintain

What	When	Who	Goal
Data Science	2010-	Data Scientist	Extract insights from data to answer questions

Collect → Clean → Integrate → Analyze → Visualize → Communicate

Data Preparation (The Last Decade)

2013



Big Data Borat
@BigDataBorat

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

6:47 PM · Feb 26, 2013 · Twitter Web Client

543 Retweets 25 Quote Tweets 397 Likes

...

2014

The New York Times

For Big-Data Scientists, ‘Janitor Work’ Is Key Hurdle to Insights

Yet far too much handcrafted work — what data scientists call “data wrangling,” “data munging” and “data janitor work” — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

Data Preparation Is Still the Bottleneck!!!

2020



The State of Data Science 2020
Moving from hype toward maturity

We were disappointed, if not surprised, to see that data wrangling still takes the lion's share of time in a typical data professional's day. Our respondents reported that almost half of their time is spent on the combined tasks of data loading and cleansing. Data

2021

“If 80 percent of our work is data preparation, then ensuring data quality is the important work of a machine learning team.”

Andrew Ng



Three Questions

1. What makes data preparation hard?
2. Why has this problem not been solved?
3. How to solve it in the next decade?

Three Questions

1. What makes data preparation hard?
2. Why has this problem not been solved?
3. How to solve it in the next decade?

What Makes Data Preparation Hard?



Collection



Cleaning



Integration



Analysis

How much time is spent on preparation?

1. **Too many tasks** (e.g., standardize date, dedup address, etc)
2. Humans have **different levels of expertise** (in data science and programming)
3. **Domain specific** (finance, social science, healthcare, economics, etc.)

Three Questions

1. What makes data preparation hard?
2. Why has this problem not been solved?
 - Academia
 - Industry
3. How to solve it in the next decade?

Three Questions

1. What makes data preparation hard?
2. Why has this problem not been solved?
 - Academia
 - Industry
3. How to solve it in the next decade?

Decades of Research

23 years ago

[BOOK] [Data preparation for data mining](#)

D Pyle - 1999 - books.google.com

Data Preparation for Data Mining addresses an issue unfortunately ignored by most authorities on data mining: **data preparation**. Thanks largely to its perceived difficulty, **data preparation** has traditionally taken a backseat to the more alluring question of how best to ...

☆ 99 Cited by 2274 Related articles All 5 versions

22 years ago

[PDF] [Data cleaning: Problems and current approaches](#)

E Rahm, HH Do - IEEE Data Eng. Bull., 2000 - cs.brown.edu

We classify data quality problems that are addressed by **data cleaning** and provide an overview of the main solution approaches. **Data cleaning** is especially required when integrating heterogeneous data sources and should be addressed together with schema ...

☆ 99 Cited by 2402 Related articles All 24 versions

20 years ago

[Data integration: A theoretical perspective](#)

M Lenzerini - Proceedings of the twenty-first ACM SIGMOD-SIGACT ..., 2002 - dl.acm.org

Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data. The problem of designing **data integration** systems is important in current real world applications, and is characterized by

☆ 99 Cited by 3528 Related articles All 25 versions

Why Has Not Been Solved By Academia?

RESEARCH-ARTICLE

The Seattle Report on Database Research

[Twitter](#) [LinkedIn](#) [Google Scholar](#) [Facebook](#) [Email](#)

Authors: [Daniel Abadi](#), [Anastasia Ailamaki](#), [David Andersen](#), [Peter Bailis](#), [Magdalena Balazinska](#), [Philip Bernstein](#), [Peter Boncz](#), [Surajit Chaudhuri](#), [Alvin Cheung](#), [AnHai Doan](#), [Luna Dong](#), [Michael J. Franklin](#), [Juliana Freire](#), [Alon Halevy](#), [Joseph M. Hellerstein](#), [Stratos Idreos](#), [Donald Kossmann](#), [Tim Kraska](#), [Sailesh Krishnamurthy](#), [Volker Markl](#), [Sergey Melnik](#), [Tova Milo](#), [C. Mohan](#), [Thomas Neumann](#), [Beng Chin Ooi](#), [Fatma Ozcan](#), [Jignesh Patel](#), [Andrew Pavlo](#), [Raluca Popa](#), [Raghu Ramakrishnan](#), [Christopher Ré](#), [Michael Stonebraker](#), [Dan Suciu](#) ([Less](#))
[Authors Info & Claims](#)

ACM SIGMOD Record, Volume 48, Issue 4 • December 2019 • pp 44–53 • <https://doi.org/10.1145/3385658.3385668>

Published: 25 February 2020

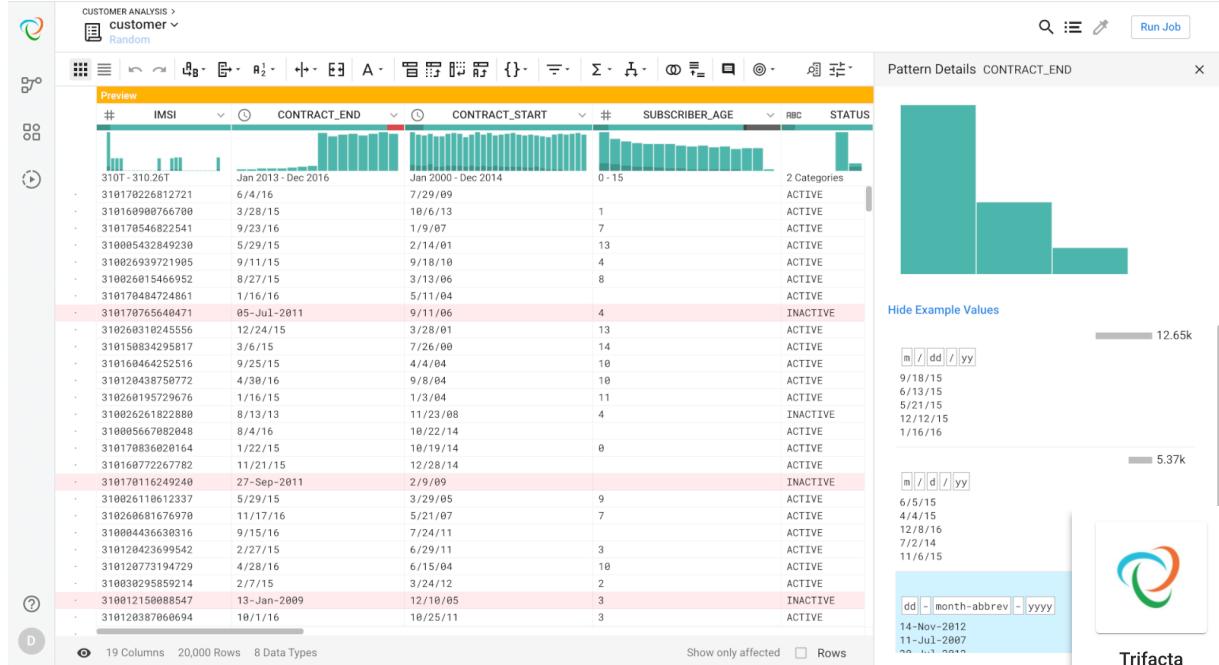
Data integration and wrangling: Data scientists repeatedly say that data integration and data wrangling is 80-90% of their challenge. These are problems the database community has worked on for decades. Thus, it can bring a solid understanding of the core challenges and known solutions. In the past, we have focused much of our efforts on solving “point problems”, e.g., algorithms for specific challenges such as entity resolution. Instead, we need

Three Questions

1. What makes data preparation hard?
2. Why has this problem not been solved?
 - Academia
 - **Industry**
3. How to solve it in the next decade?

Three Directions

➤ Spreadsheet GUI



➤ Workflow GUI

➤ Notebook GUI

Three Directions

➤ Spreadsheet GUI

➤ Workflow GUI

➤ Notebook GUI

The screenshot shows the Alteryx Data Designer interface. On the left, there's a 'Browse' window displaying a histogram of ZIP codes with a frequency count from 0 to 80. Below it is a 'Data Quality' section for the 'ZIP' field, showing 1,735 records, 255 unique values, and 0 nulls. A 'Results' table at the bottom shows 14 columns of customer data for records 1 through 5.

The central part of the screen displays a data preparation workflow:

- Input Data:** Two sources are connected: 'Customers CRM' (via a book icon) and 'Transactions AWS' (via a book icon).
- Data Preparation:** The data from both sources passes through a 'Crosstab' (binoculars icon), a 'Filter' (blue circle icon), and a 'Formula / Calculate Fields' (beaker icon).
- Data Blend:** The prepared data from the preparation stage is joined (purple circle icon) and then passed through a 'Summarize / Pivot Table' (red sigma icon) and a 'VLOOKUP' (green square icon).
- Visual Analytics:** The final output is visualized using a 'Visualytics' component (blue square icon).

At the bottom right, the Alteryx logo is displayed.

Record #	Customer_ID	Address	City	Customer_Segment	First_Name	Last_Name	Responder	State	Store_Number	Sur
1	5	1234 Main St	Denver	Home Office	INDIA	JOSEPH	No	(Null)	(Null)	(Null)
2	6	1599 Williams St	Denver	Home Office	IR	MACK	No	CO	106	(Null)
3	7	12096 E Lake Cir	Greenwood Village	Home Office	MARISSA	LATTA	No	CO	105	(Null)
4	8	7225 S Gaylord St	Centennial	Home Office	PHYLLIS	WALKER	No	CO	101	(Null)
5	9	4497 Comish Way	Denver	Home Office	VIVIAN	GAULDEN	No	CO	105	(Null)

Three Directions

➤ Spreadsheet GUI



➤ Workflow GUI

➤ Notebook GUI

```
In [1]: import qgrid
import numpy as np
import pandas as pd

df = pd.DataFrame(np.random.rand(3,5))
qg = qgrid.show_grid(df)
qg
```

	0	1	2	3	4
0	0.09835	0.65988	0.24213	0.1324	0.21811
1	0.11348	0.17392	0.18452	0.25759	0.45406
2	0.77328	0.2921	0.37934	0.87462	0.93311

jupyter

Why Has Not Been Solved By Industry?

Spreadsheet/Workflow GUIs
(for non-programmers)



talend



Notebook GUI
(for data scientists)

???



Three Questions

1. What makes data preparation hard?
2. Why has this problem not been solved?
 - Academia
 - Industry
3. How to solve it in the next decade?

Vision

Machine Learning Made Easy



Initial release: 2007; 14 years ago

Big Data Made Easy



Initial release: 2011; 10 years ago

Deep Learning Made Easy



Initial release: 2016; 5 years ago

Data Preparation Made Easy

The next decade



SortingHat

Data Civilizer ...

dataprep

<https://dataprep.ai>



1000+ Stars



300K+ Downloads



30+ Contributors

DataPrep.Connector

Available to use

DataPrep.EDA

Available to use

DataPrep.Clean

Available to use

DataPrep.Feature

Planning

DataPrep.Integrate

Planning

... and more

Task-Centric API Design

1. Declarative
2. Support both coarse-grained and fine-grained tasks
3. Easy to customize

EDA tasks

- plot(df): “I want to see an overview of the dataset”
- plot_missing(df): “I want to understand the missing values of the dataset”
- plot(df, x): “I want to understand the column x”
- plot(df, x, y): “I want to understand the relationship between x and y”
- plot(df, x, config = {hist.bins: 10}): “Set the number of bins of histograms to 10”

Summary

- Data Warehouse
- Parallel Databases
- Big Data Processing
- Cloud Databases
- Data Science

Acknowledge

- Some lecture slides were copied from or inspired by the following course materials
 - “W4111: Introduction to databases” by Eugene Wu at Columbia University
 - “CSE344: Introduction to Data Management” by Dan Suciu at University of Washington
 - “CMPT354: Database System I” by John Edgar at Simon Fraser University
 - “CS186: Introduction to Database Systems” by Joe Hellerstein at UC Berkeley
 - “CS145: Introduction to Databases” by Peter Bailis at Stanford
 - “CS 348: Introduction to Database Management” by Grant Weddell at University of Waterloo
 - UC Berkeley DS100 Fall 2017