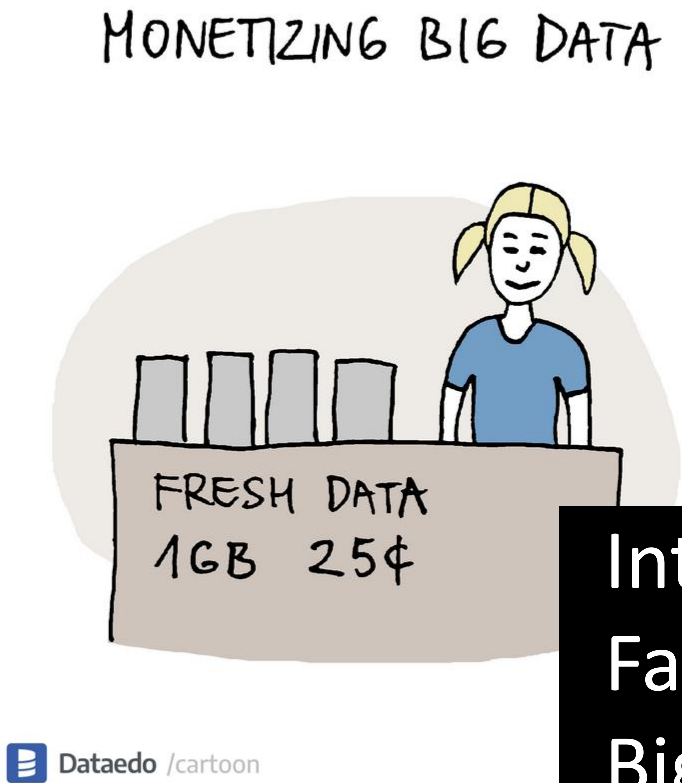


Introduction to Database Systems
Fall 2024, Lecture 12:
Big Data Management

Lecture Outline

Learning Outcome 6: Discuss the pros and cons of different classes of data systems for modern analytics and data science applications

- Characteristics of Big Data
- Classes of Big Data Processing Systems
- Example: MapReduce, Spark

Lecture Outline

Learning Outcome 6: Discuss the pros and cons of different classes of data systems for modern analytics and data science applications

- **Characteristics of Big Data**
- Classes of Big Data Processing Systems
- Example: MapReduce, Spark

Beyond Small Data

Online Transactional Processing (OTLP)

- Build great interactive applications

➔ "Small" data

This class: beyond small data

- Keep track of all the history
- Keep track of all interactions, also low-level
- Keep track of all data: media, user input, logs

➔ "Big Data"

Characteristics of Big Data

Five Vs of Big Data describe key dimensions of big data

Volume 

Velocity 

Variety 

Veracity 

Value 

Volume

Definition: The amount of data generated, stored and processed.

Significance: Data management system must be scale to given data volume.

Volume

Source: https://www.domo.com/learn/infographic/data-never-sleeps-11

Volume

Rivian electric vehicles generate multiple TB of data per day

Volume

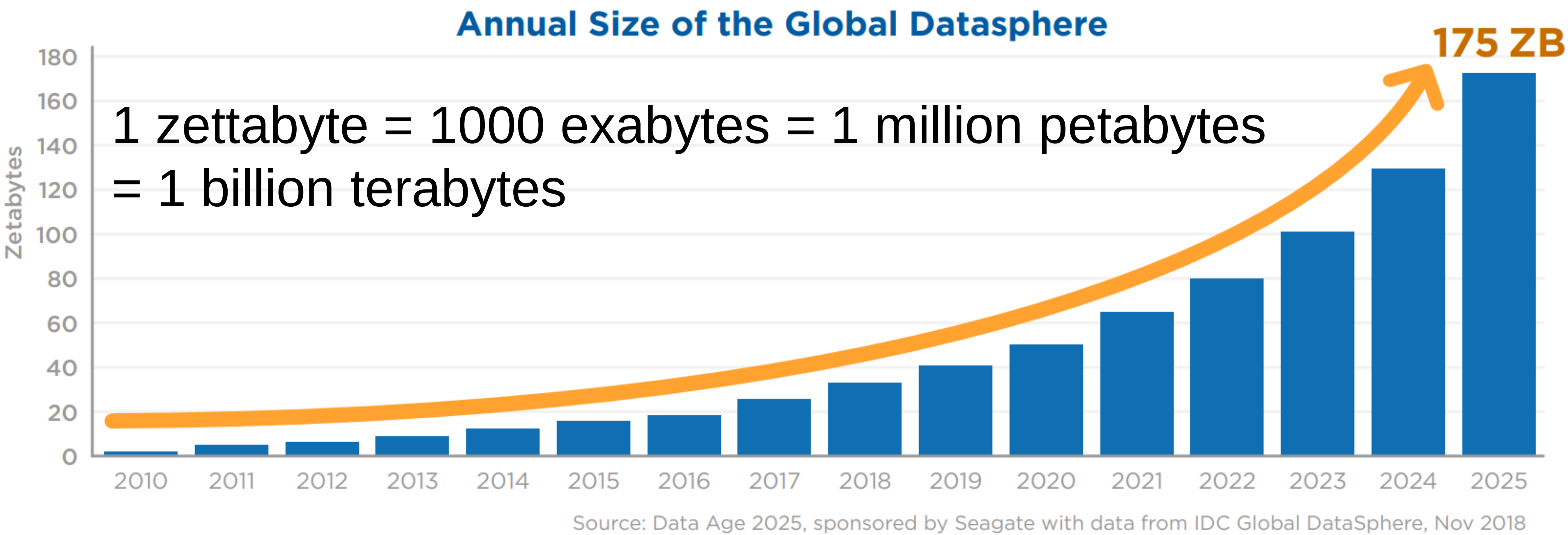
CERN: „In June 2022, we had 424 PB of data on tapes”

Volume

301,722 Datasets Available

Source: Data.gov

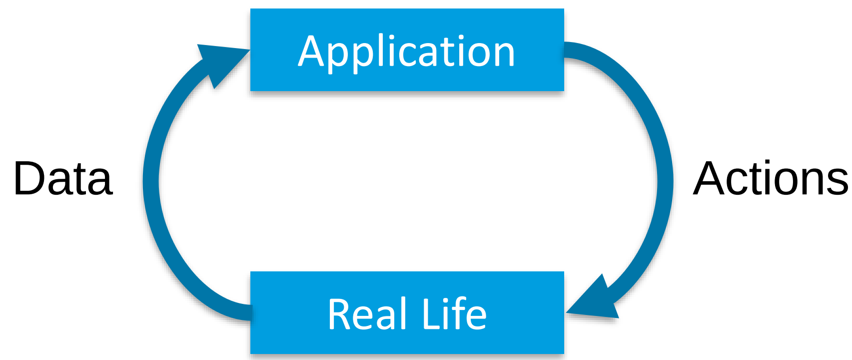
Growth of Data Globally



Source: [Reinsel, Gantz, Rydning; The Digitization of the World from Edge to Core; IDC White Paper 2018]

Velocity

Definition: The speed at which data is generated, collected, and processed.



Significance: Requirement on the system to process data at given speed.

Variety

Definition: The diversity of data types and sources. Data can come in **structured, semi-structured, or unstructured** formats.

Significance: Challenge of managing different types of data. Choose data management system for your use case.

Structured data

- Highly regular structure, repeating patterns
- Example: relational data

ID	Name
1	Pinar
2	Veronika
3	Dovile
4	Zoi
5	Martin
6	Eleni

Semi-structured data

- Some structure, changes over time
- Examples: logs, comment threads, graphs

Common Log Format*

```
127.0.0.1 alice Alice [06/May/2021:11:26:42 +0200] "GET / HTTP/1.1" 200 3477
```

Same log line in JSON

```
{
  "ip_address": "127.0.0.1",
  "user_id": "alice",
  "username": "Alice",
  "timestamp": "06/May/2021:11:26:42 +0200",
  "request_method": "GET",
  "request_url": "/",
  "protocol": "HTTP/1.1",
  "status_code": 200,
  "response_size_bytes": 3477
}
```


*Source: [Common Log Format - Wikipedia]

Variety

PHONES OUT

Advantages of JSON over Common Log Format

https://www.menti.com/al2n6oqco3vf

Common Log Format

```
127.0.0.1 alice Alice [06/May/2021:11:26:42 +0200] "GET / HTTP/1.1" 200 3477
```

Same log line in JSON

```json
{
  "ip_address": "127.0.0.1",
  "user_id": "alice",
  "username": "Alice",
  "timestamp": "06/May/2021:11:26:42 +0200",
  "request_method": "GET",
  "request_url": "/",
  "protocol": "HTTP/1.1",
  "status_code": 200,
  "response_size_bytes": 3477
}
```

Variety

Unstructured data

- Little structure
- Examples: photos, videos, music

Veracity

Definition: The accuracy and trustworthiness of data.

Significance: Data quality varies, and not all collected data can be trusted. It’s essential to filter, clean, and validate data to ensure accurate insights.

Veracity

Only 3% of Companies' Data Meets Basic Quality Standards

by Tadhg Nagle, Thomas C. Redman, and David Sammon

September 11, 2017

Source: Only 3% of Companies' Data Meets Basic Quality Standards, Harvard Business Review

Veracity

A 2016 study by IBM is even more eye-popping. IBM found that poor data quality strips $3.1 trillion from the U.S. economy annually due to lower productivity, system outages and higher maintenance costs, to name only a handful of the bad outcomes that result from poor data quality.

Source: Flying Blind: How Bad Data Undermines Business, Forbes

Veracity

Definition: The accuracy and trustworthiness of data.

Significance: Data quality varies, and not all collected data can be trusted. It’s essential to filter, clean, and validate data to ensure accurate insights.

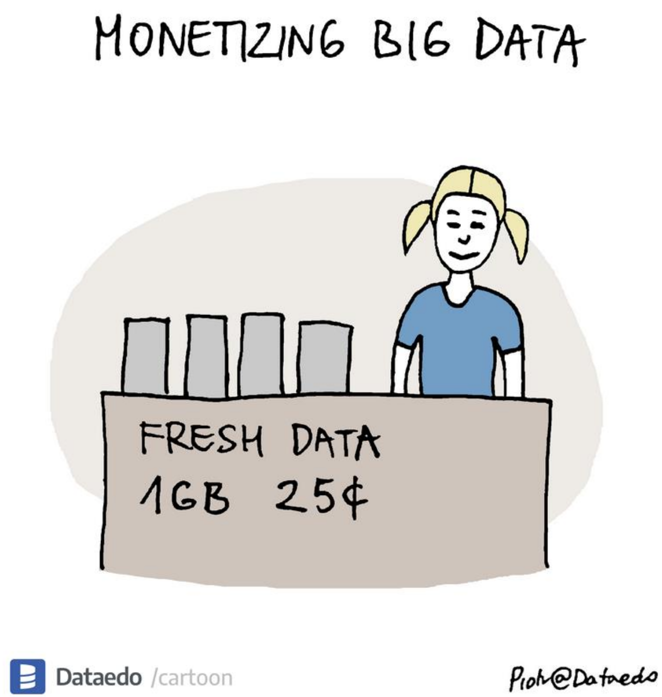
Value

Definition: The ability to turn data into insights for decision making and creating business value.

Significance: Big data is only beneficial if it delivers insights.

- Businesses: more effective operations, stronger customer relationships, $$$
- Science: new discoveries
- Public sector: better serving people

Only half a joke: data marketplaces are a thing!



Snowflake Data Sharing & Marketplace

DATA SHARING[1]

36%

of customers[1] have at ≥1 stable edge[1]

MARKETPLACE LISTINGS[1]

2,946

26% Y/Y Growth

Source: [Snowflake Investor Presentation], October 31, 2024

Characteristics of Big Data

Five Vs of Big Data describe key dimensions of big data

Volume 

Velocity 

Variety 

Veracity 

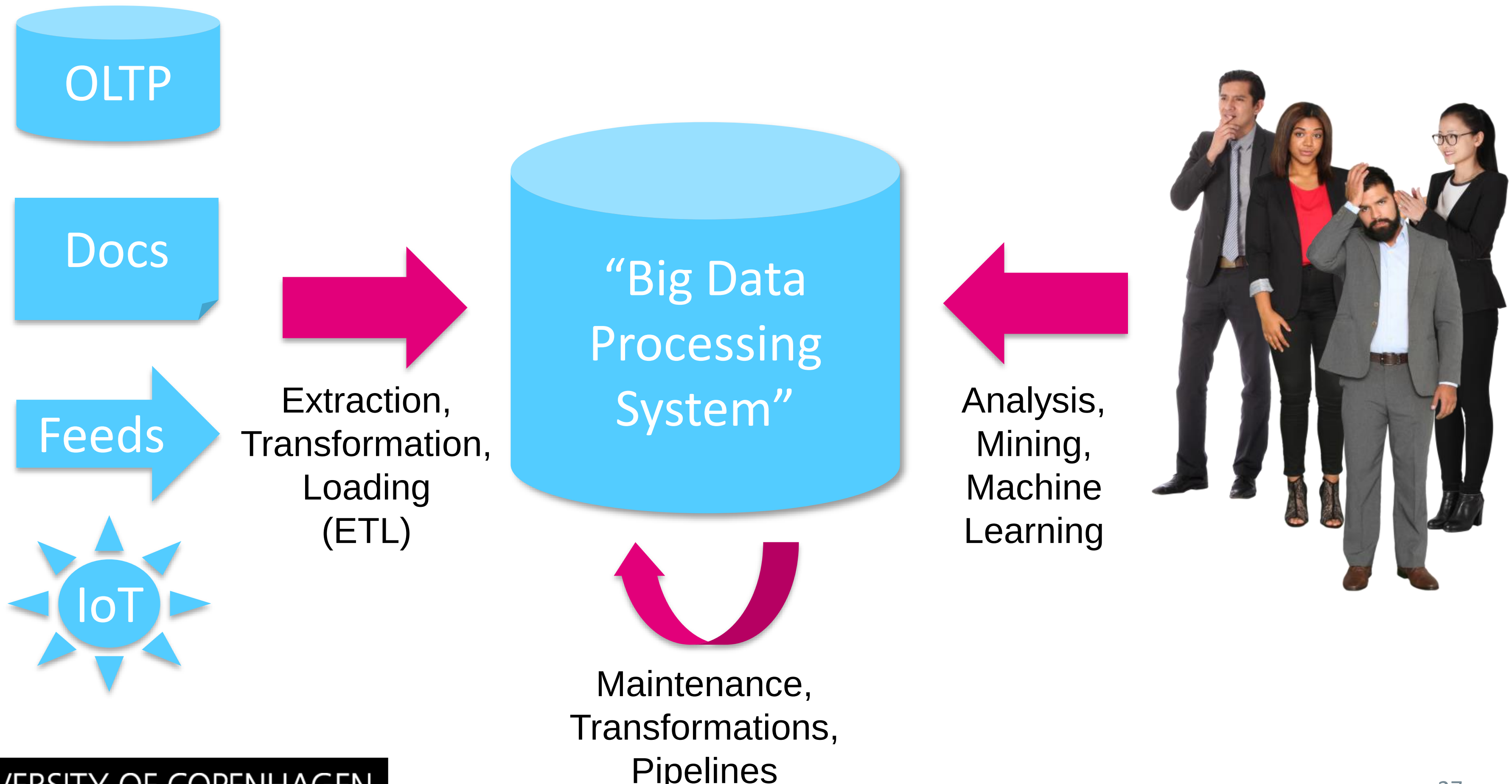
Value 

Lecture Outline

Learning Outcome 6: Discuss the pros and cons of different classes of data systems for modern analytics and data science applications

- Characteristics of Big Data
- **Classes of Big Data Processing Systems**
- Example: MapReduce, Spark

Big Data Processing Overview



Access Patterns to Big Data

Data collections typically consists of many millions of files of data

- Too large for a single server ➔ Distributed storage
- Reading large amounts of data ➔ Sequential scans

Online Analytical Processing (OLAP)

Processing Requirements

Apply filters to reduce data quantity

- Like SQL execution with predicates

Run complex processing pipelines

- Periodic tasks & triggered tasks upon new data
- User-defined functions (UDFs)
- Server-side applications (Python, ML)

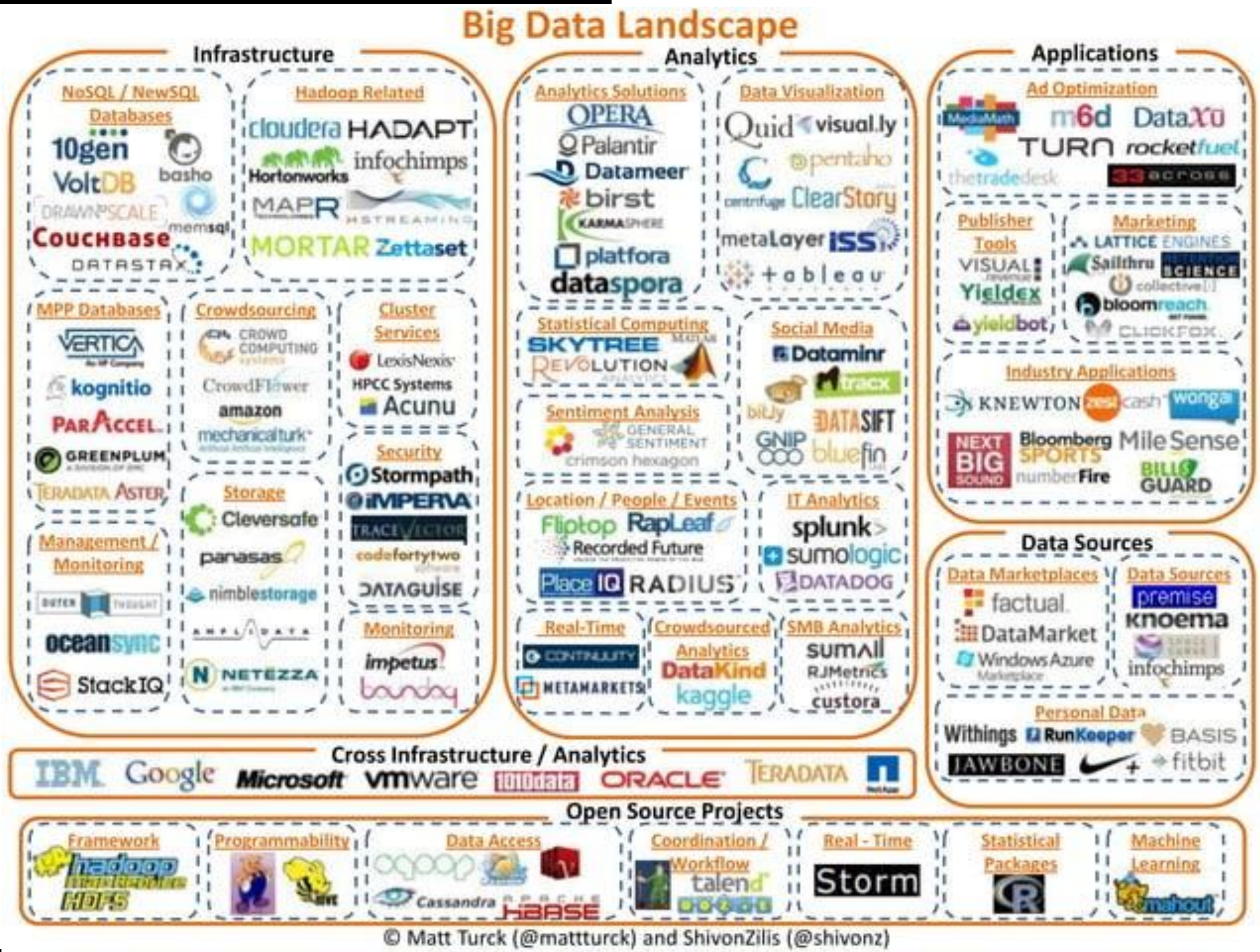
Too large for a single server ➔ Distributed processing

Big Data Processing System

OLAP Database System

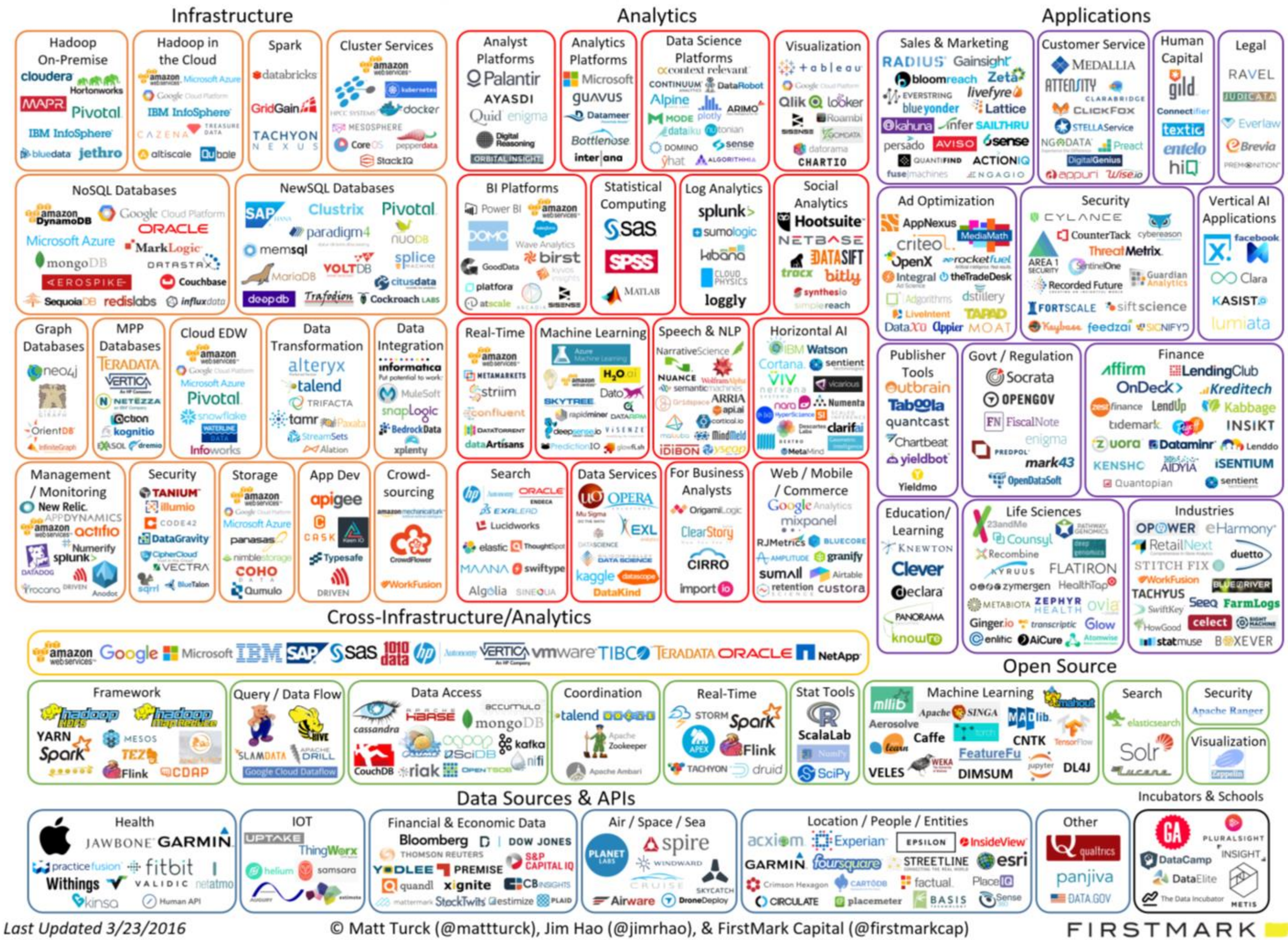
- Distributed storage for exabytes of data
- Distributed compute, typically using large, sequential scans
- Large ecosystem of tools for complex data analysis tasks

Big Data Landscape 2012

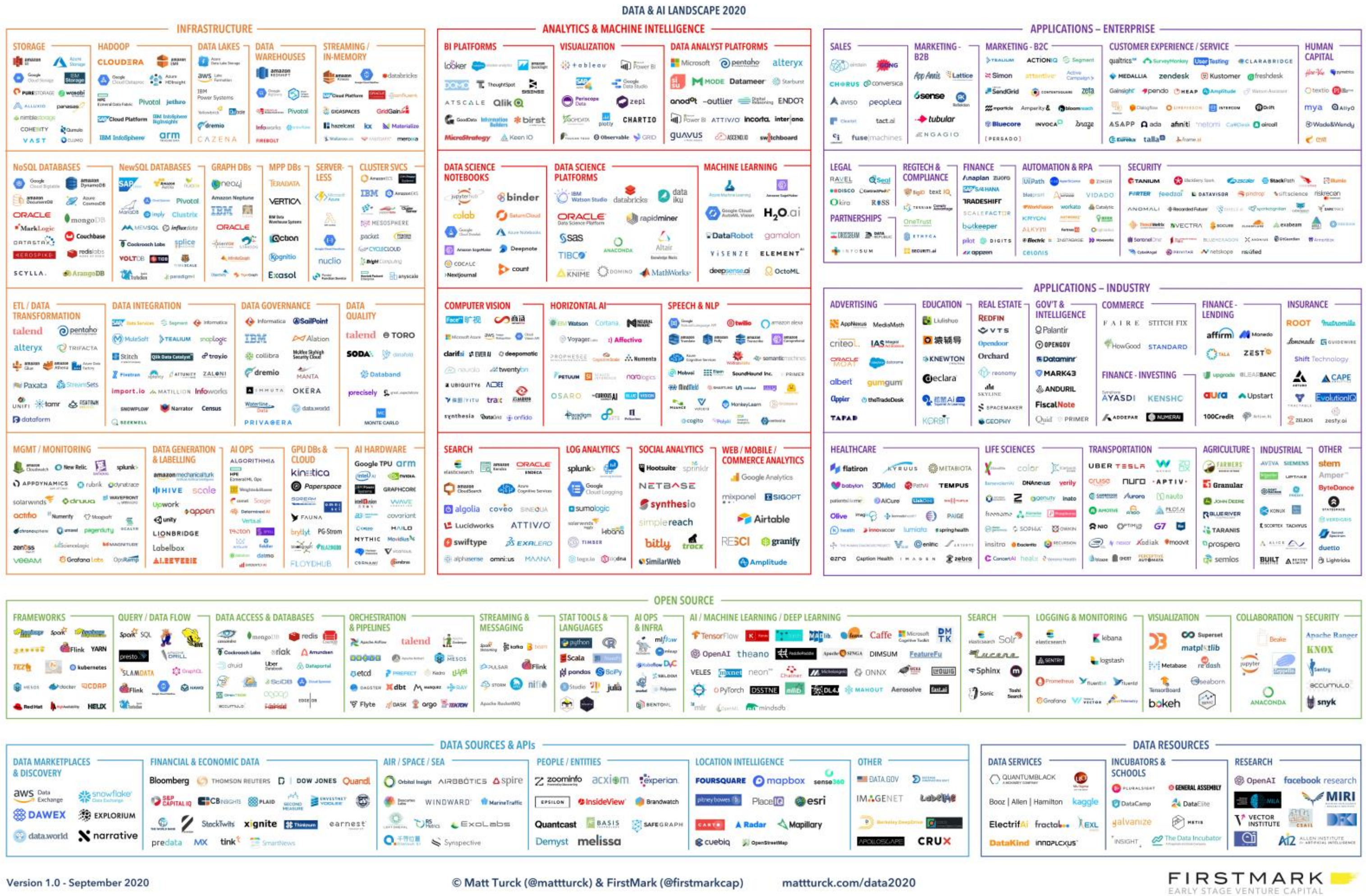


Source: [A chart of the big data ecosystem – Matt Turck]

Big Data Landscape 2016



Big Data Landscape 2020



Source: [Resilience and Vibrancy: The 2020 Data & AI Landscape – Matt Turck]

Big Data Landscape 2023

Conclusions:

- No product does it all
- Typically, you use a set of products
  e.g. ETL → Storage & Processing → UI, Reporting, Serving
- Interesting space for software engineers, sales engineers, sales

Source: [MAD 2023 - Version 1.0]

Classes of Big Data Processing Systems

Relational Database Systems

Strengths

- SQL programming API
- Huge SQL ecosystem
- ACID transactions
- Complex query processing

Weaknesses

- Little support for unstructured data
- Little support for machine learning training and serving

snowflake Amazon Redshift teradata.

Classes of Big Data Processing Systems

MapReduce Systems

Strengths

- Distributed storage
- Great scalability
- Able to process great variety of data sets (structured, semi-structured, unstructured)

Weaknesses

revealed later

Classes of Big Data Processing Systems

Key-value Stores

Strengths

- Distributed storage
- Great scalability
- Short latency for single items
- Quick, “out of the box” way to store data

Weaknesses

- Limited storage model (only key, value pairs)
- Limited query interface (no SQL)
- No means to scan and filter data

Amazon DynamoDB Redis cassandra

Document Databases

Strengths

- Store objects / XML / JSON in hierarchical form
- Good integration with object-oriented languages and JavaScript

Weaknesses

- Limited query interface (no SQL)
- No ACID guarantees
- Not designed for scans (very related to key-value stores)

MongoDB Couchbase

Classes of Big Data Processing Systems

Graph Databases

Strengths

- Capture graph relationships, e.g. knowledge graphs, social networks
- Fast at traversing edge chains, no joins needed

Weaknesses

- Specific to graph applications
- Not many such use cases
- Relational databases outperform graph databases these days

Classes of Big Data Processing Systems

Data Lakehouses

Strengths

- Data storage in open data formats (Apache Parquet + Apache Iceberg) in the cloud
- Great ML support, e.g. training, but also Python-based notebooks

Weaknesses

- Typical API is Apache Spark, less versatile and supported than SQL
- However: Lakehouses are moving to SQL

databricks

Lecture Outline

Learning Outcome 6: Discuss the pros and cons of different classes of data systems for modern analytics and data science applications

- Characteristics of Big Data
- Classes of Big Data Processing Systems
- **Example: MapReduce**, Spark

MapReduce Overview

Early tools to deal with Big Data

- MapReduce by Google in 2003
- Open-source Apache Hadoop based on MapReduce

Interface

- Map (k1, v1) → list (k2, v2)
- Reduce (k2, list(v2)) → (k2, v3)

MapReduce Framework

Framework

- Read lots of data (e.g. text documents)
- **Map:** process a data item
- Sort and shuffle
- **Reduce:** aggregate
- Write results

Map and Reduce functions are user-specified

MapReduce Phases

Map Phase

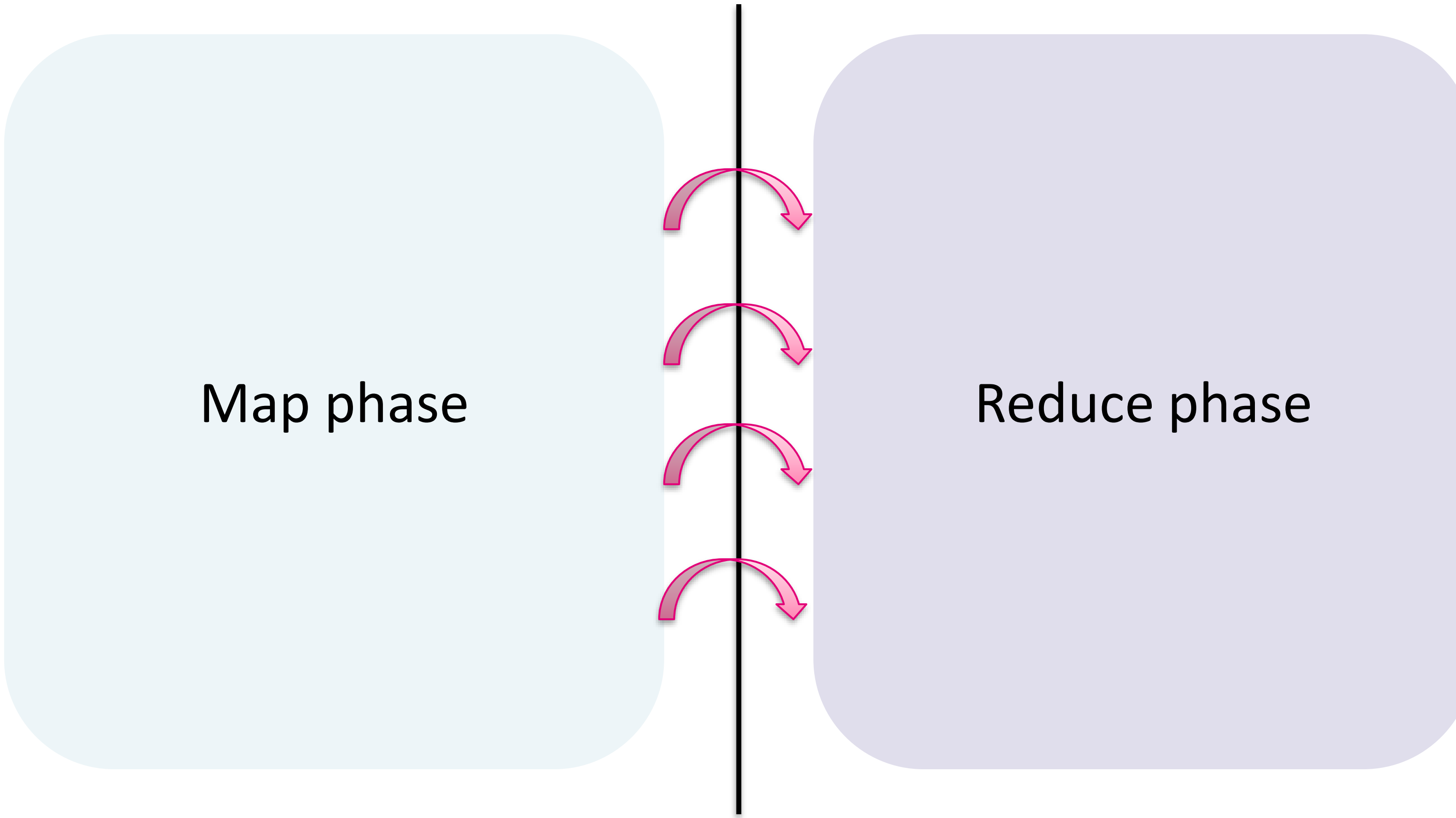
- Input: (k1, v1)
- Output: list (k2, v2)
- Independent for every key-value pair
  → mapping processes run in parallel

Shuffle Phase: intermediate results are shuffled through the network

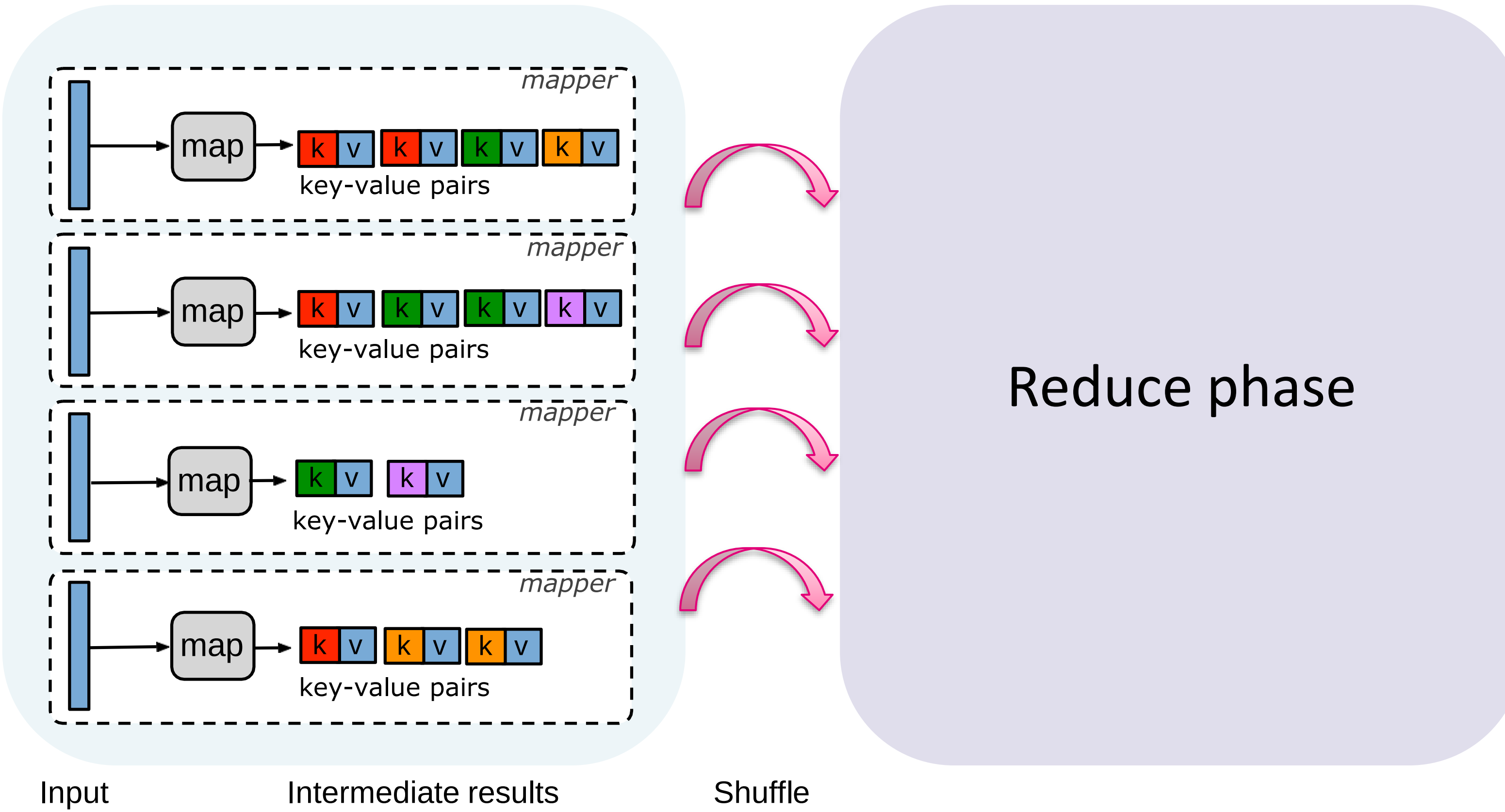
Reduce Phase

- Input: (k2, list(v2))
- Output: (k2, v3)
- Independent per group
  → reducer processes can run in parallel (per group)

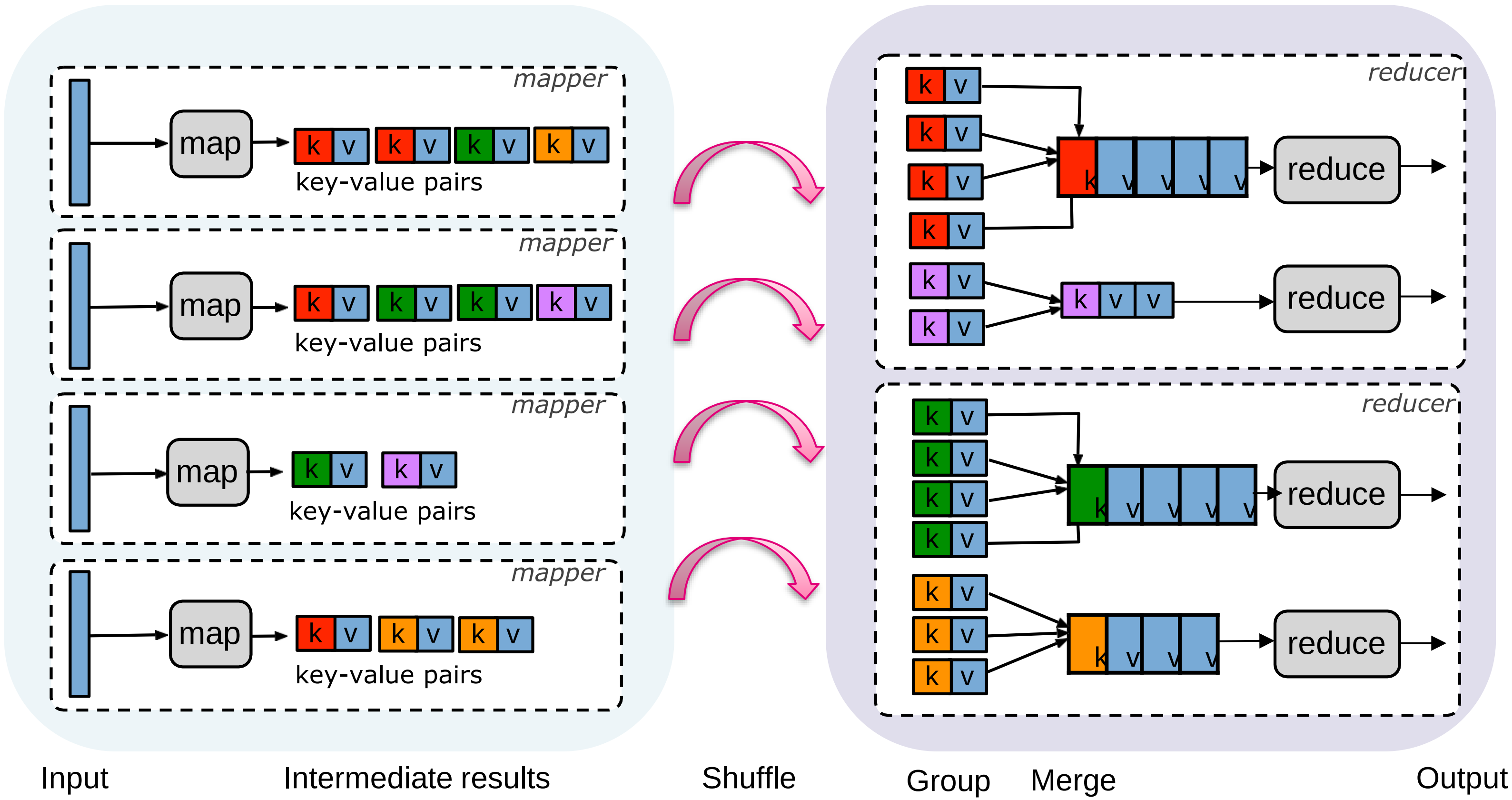
MapReduce Illustration



MapReduce Illustration



MapReduce Illustration



MapReduce Wordcount Example

Wordcount Example

- Given a text document, output the count for each word.

MapReduce interface

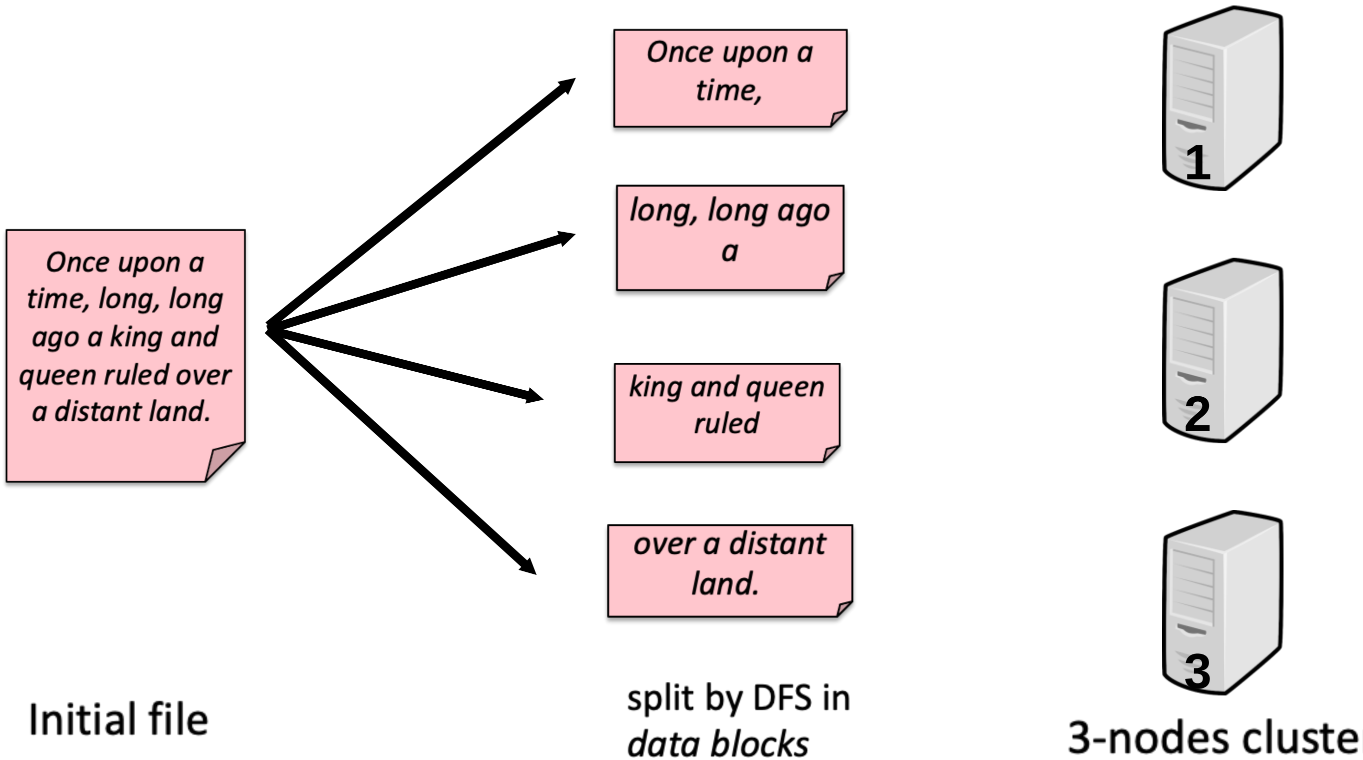
- Map (k1, v1) → list (k2, v2)
- Reduce (k2, list(v2)) → (k2, v3)

MapReduce Wordcount Example

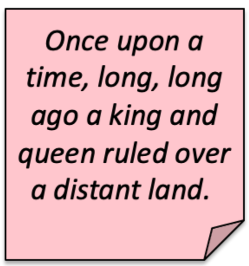
MapReduce implementation

- Map (filename, text) → {word, 1}
- Reduce (word, [1,1,…]) → (word, count)

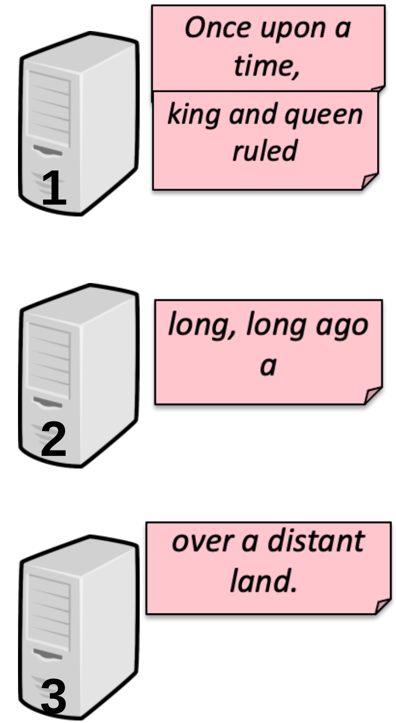
MapReduce Wordcount Example



MapReduce Wordcount Example

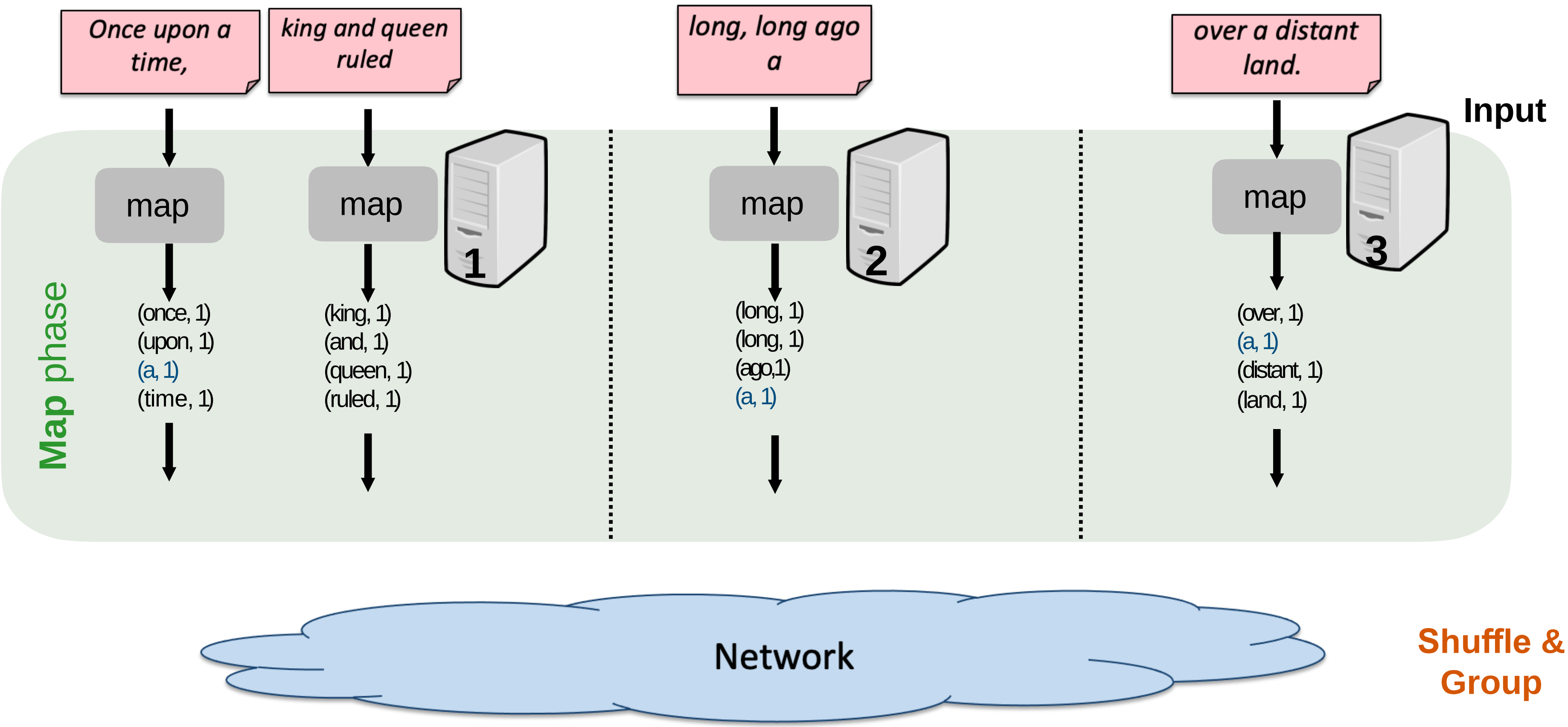


Initial file

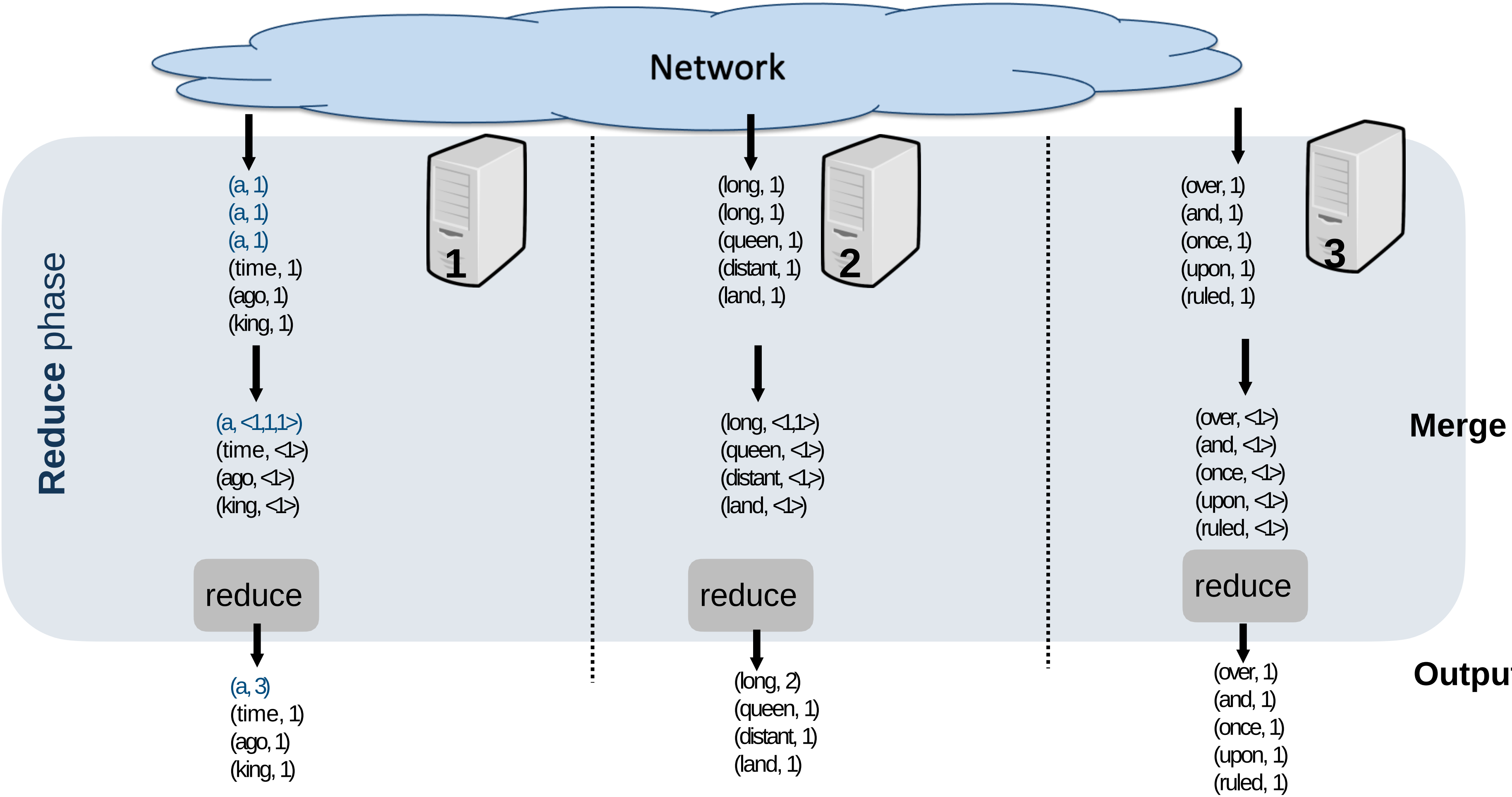


3-nodes cluster

Map Phase


Once upon a time,
king and queen ruled
long, long ago a
over a distant land.
Input
map
map
map
map
1
2
3
Map phase
(once, 1)
(upon, 1)
(a, 1)
(time, 1)
(king, 1)
(and, 1)
(queen, 1)
(ruled, 1)
(long, 1)
(long, 1)
(ago,1)
(a, 1)
(over, 1)
(a, 1)
(distant, 1)
(land, 1)
Network
Shuffle & Group

Reduce Phase



Pseudocode

```
map(String key, String value):
```

?

```
reduce(String key, Iterator values):
```

?

Pseudocode

MapReduce

```
map(String key, String value):
  // key: document name
  // value: document contents
  for each word w in value:
    EmitIntermediate(w, "1");

reduce(String key, Iterator values):
  // key: a word
  // values: a list of counts
  int result = 0;
  for each v in values:
    result += ParseInt(v);
  Emit(AsString(result));
```

Source: [Dean, Ghemawat; MapReduce: Simplified Data Processing on Large Clusters; OSDI'2004]

SQL

select word, count(*) from docs
group by word

MapReduce Summary

Early tool to operate on Big Data

- Invented by Google in 2003
- Initially great tool for distributed processing of large amounts of data
- Used to build the web page index, many other applications
- Google moved away in 2010, killed it in 2014
- Apache Hadoop's market share on significant decline

Classes of Big Data Processing Systems

MapReduce Systems

Strengths

- Distributed storage
- Great scalability
- Able to process great variety of data sets (structured, semi-structured, unstructured)

Weaknesses

https://www.menti.com/al2n6oqco3vf

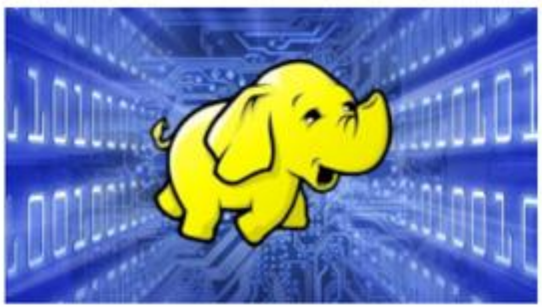
hadoop CLOUDERA HORTONWORKS

Hadoop Critique

March 13, 2017

Hadoop Has Failed Us, Tech Experts Say

Alex Woodie



The Hadoop dream of unifying data and compute in a distributed manner has all but failed in a smoking heap of cost and complexity, according to technology experts and executives who spoke to *Datanami*.

"I can't find a happy Hadoop customer. It's sort of as simple as that," says Bob Muglia, CEO of Snowflake Computing, which develops and runs a cloud-based relational data warehouse offering. "It's very clear to me, technologically, that it's not the technology base the world will be built on going forward."

Why bother?

Teaches foundational concepts in distributed systems

- **Parallel computing:** MapReduce divides data processing tasks into small, parallelizable units
- **Data processing:** Easy to understand data flow through the system
- **Fault tolerance:** internal mechanism to recover from failures
- **Scalability:** demonstrates how to scale “horizontally” (across many servers)

Lecture Outline

Learning Outcome 6: Discuss the pros and cons of different classes of data systems for modern analytics and data science applications

- Characteristics of Big Data
- Classes of Big Data Processing Systems
- **Example:** MapReduce, **Spark**

Spark

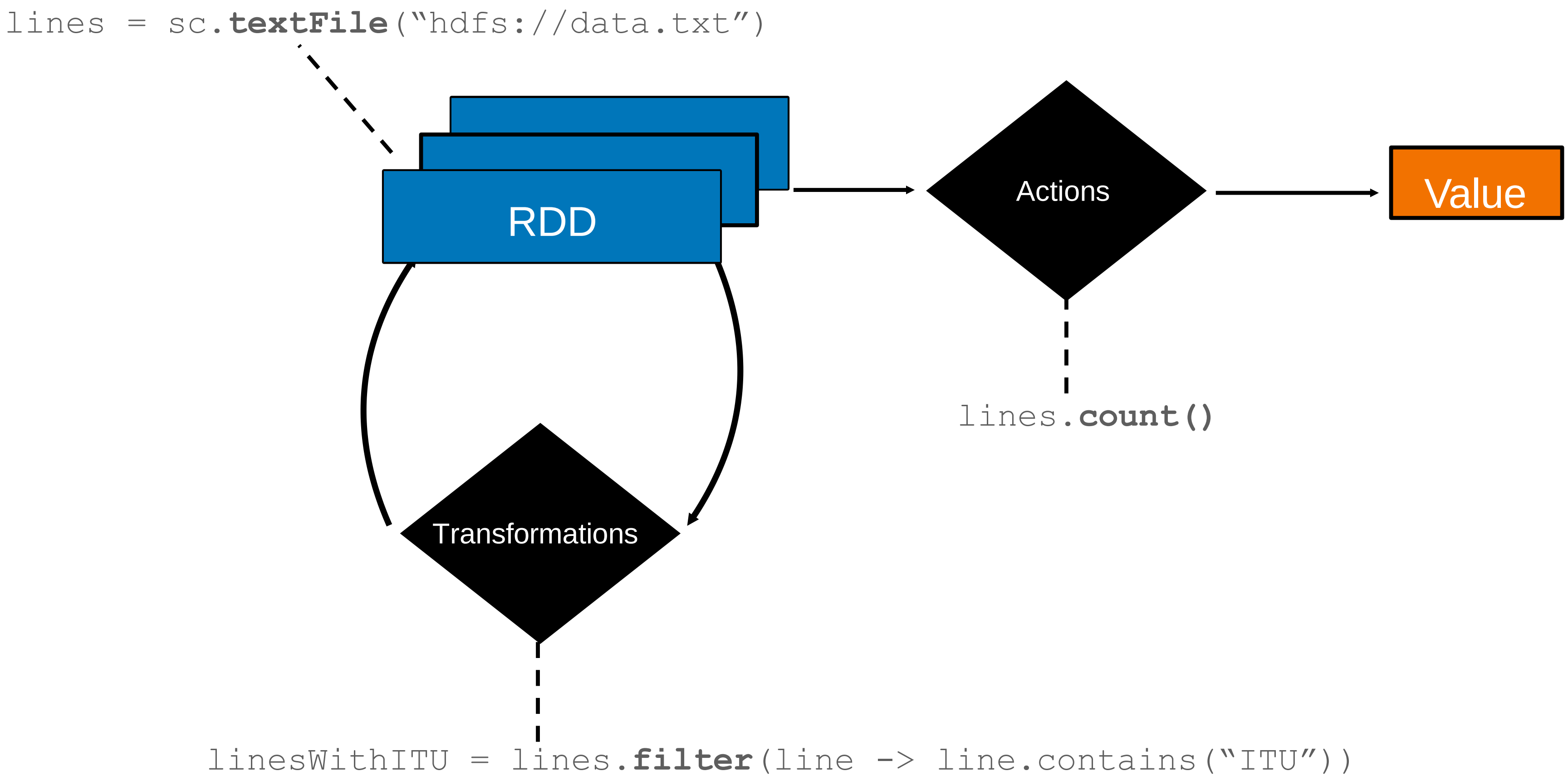
Improvement over MapReduce, developed in 2012

Data model: Resilient Distributed Datasets (RDDs)

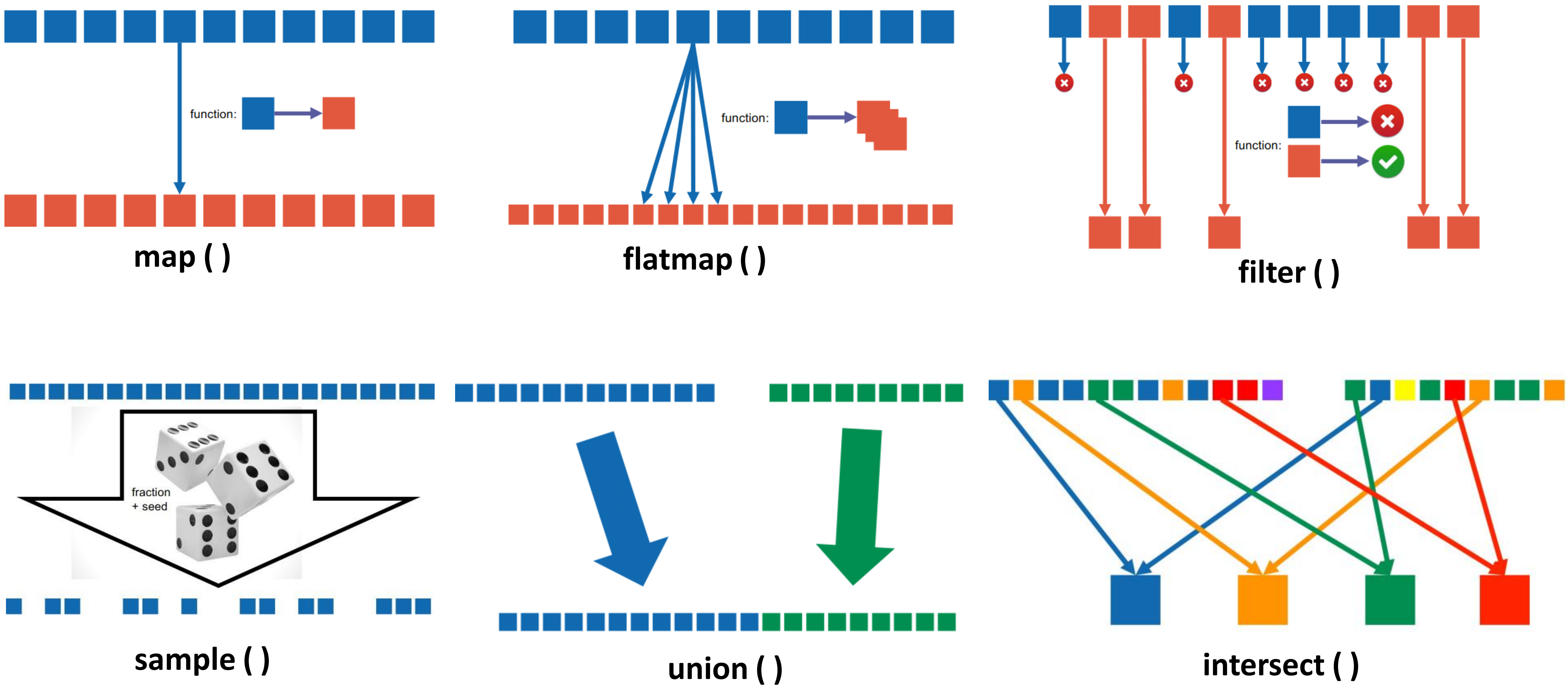
- Transform one RDD to another via operators
- Lady execution – optimizations

Write programs in terms of distributed datasets and operations on them

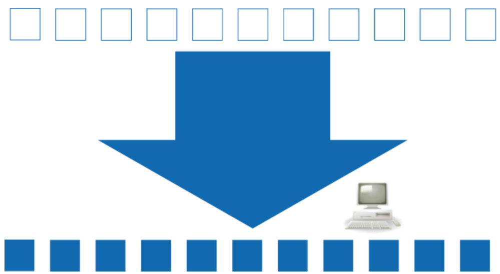
Working with RDDs

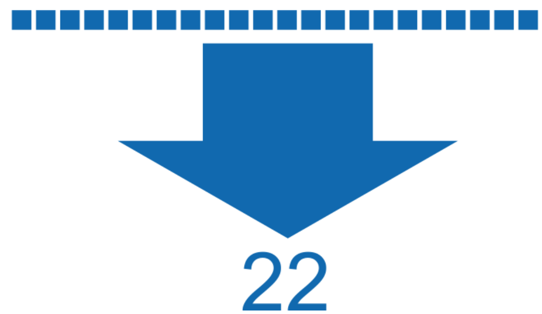


Example Transformations



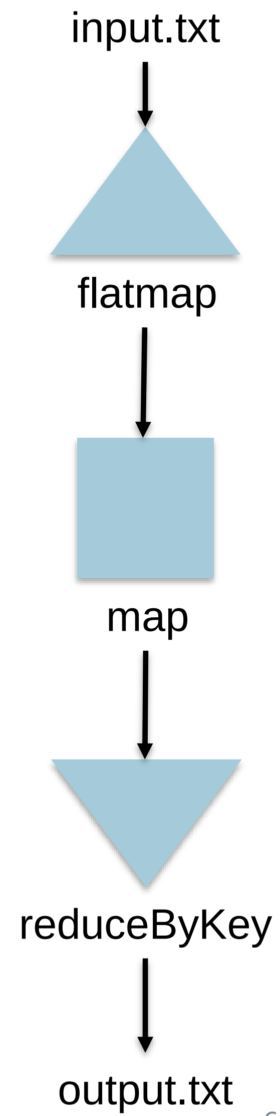
Example Actions





Spark Wordcount Example

```
file = sc.textFile("hdfs://...")
counts = file.flatMap(lambda line: line.split(" "))
                .map(lambda word: (word, 1))
                .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```



Take Aways

Learning Outcome 6: Discuss the pros and cons of different classes of data systems for modern analytics and data science applications

- Data sizes and requirements on systems continue to grow
- Many systems to process big data initially built to solve difficult problems (e.g. MapReduce for large, distributed data manipulation)
- SQL systems mostly caught up, pushed specialized systems out
- Pattern repeating (LLM training, vector databases)

Recommended reading

- Michael Stonebraker, Andrew Pavlo, What Goes Around Comes Around… And Around…, SIGMOD Record, June 2024
  https://db.cs.cmu.edu/papers/2024/whatgoesaround-sigmodrec2024.pdf