EDUFI Winter School 2018

Introduction to Big Data Management and UDBMS research in Helsinki

Jiaheng Lu

Department of Computer Science

University of Helsinki

Big number, small number – from data to understanding

Outline

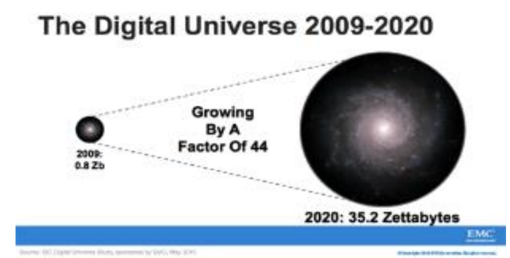
- Introduction to Big Data
- Cloud computing
- MapReduce programming model
- Our research on multi-model databases and big data

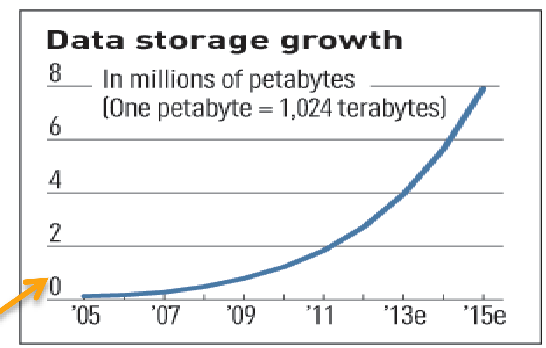
Four V’s

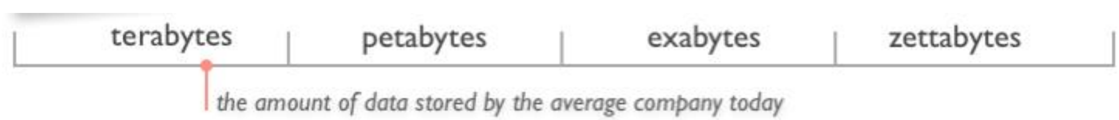

BIG DATA
VOLUME
DATA SIZE
VELOCITY
SPEED OF CHANGE
VARIETY
DIFFERENT FORMS OF DATA SOURCES
VERACITY
UNCERTAINTY OF DATA

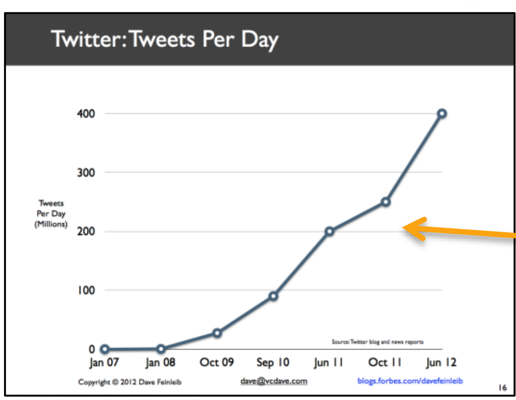
Volume (Scale)

- **Data Volume**
  - 44x increase from 2009 to 2020
  - From 0.8 Zettabytes to 35 Zb
- Data volume is increasing exponentially





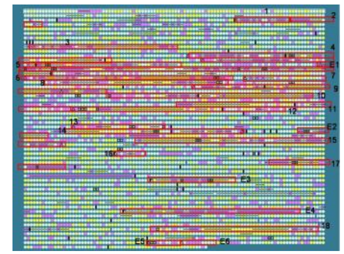


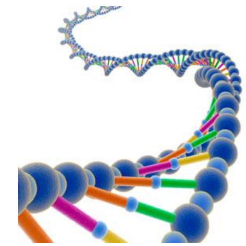


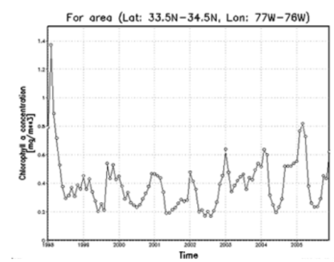
Exponential increase in collected/generated data

Variety (Complexity)

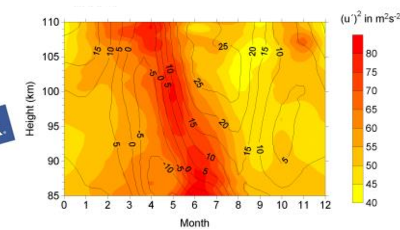
- Relational Data (Tables/Transaction/Legacy Data)
- Text Data (Web)
- Semi-structured Data (XML)
- Graph Data
  - Social Network, Semantic Web (RDF), …







To extract knowledge➔ all these types of data need to linked together

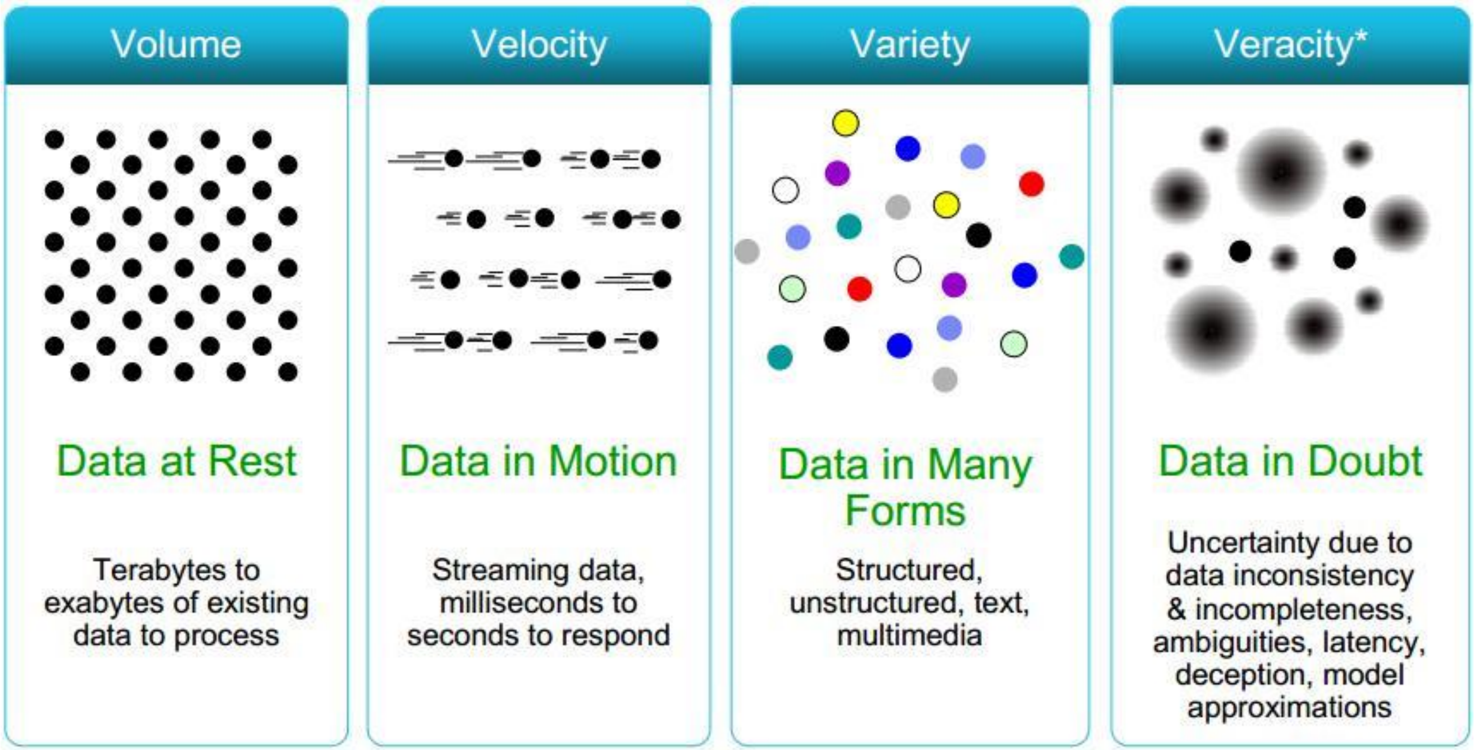


Velocity (Speed)

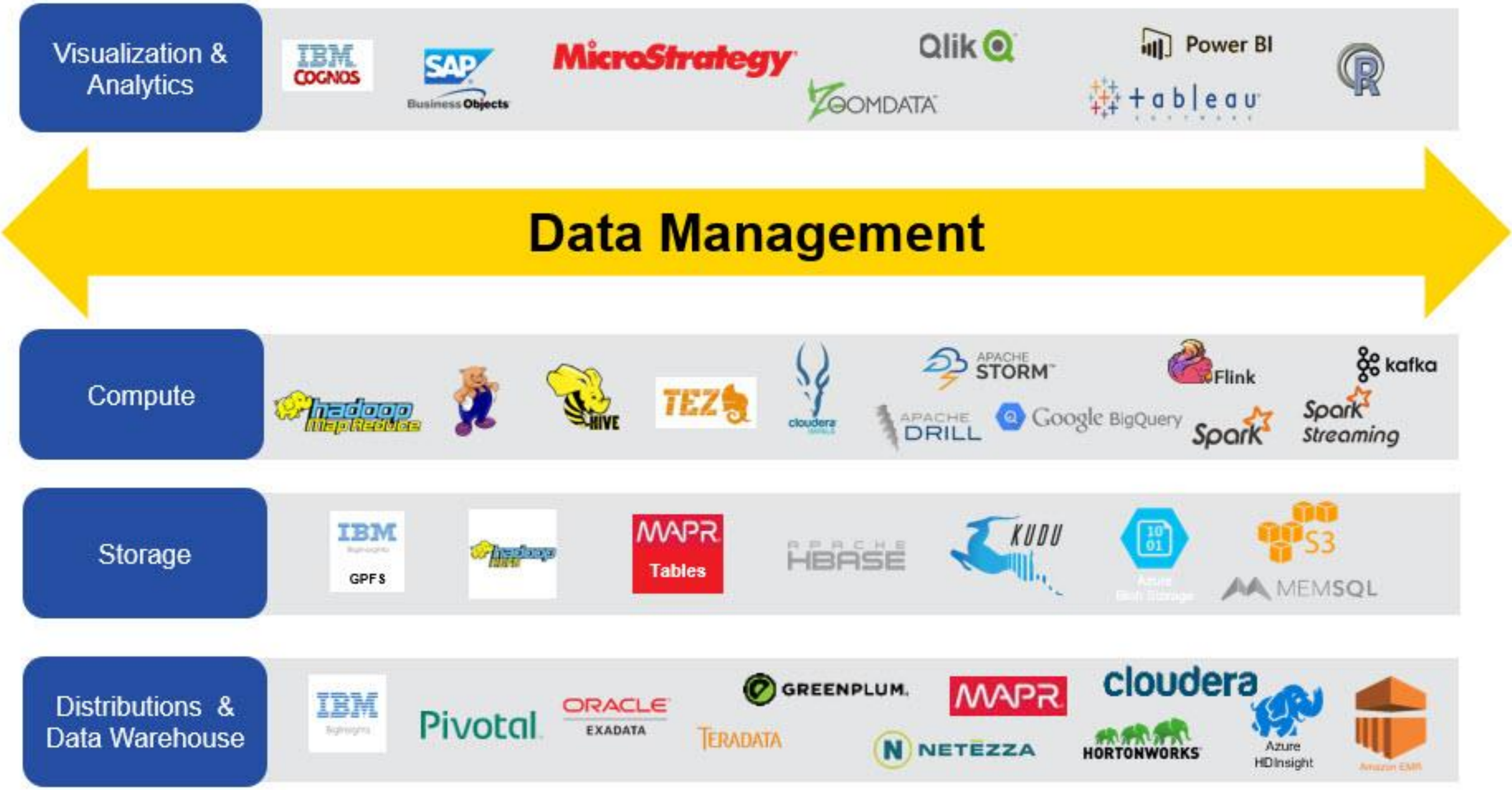
- Data is generated fast and needs to be processed fast
- Late decisions ➔ missing opportunities

- **Examples**
  - **E-Promotions:** Based on your current location, your purchase history, what you like ➔ send promotions right now for store next to you
  - **Healthcare monitoring:** sensors monitoring your activities and body ➔ any abnormal measurements require immediate reaction

Big data 4V's


Volume
Data at Rest
Terabytes to exabytes of existing data to process
Velocity
Data in Motion
Streaming data, milliseconds to seconds to respond
Variety
Data in Many Forms
Structured, unstructured, text, multimedia
Veracity*
Data in Doubt
Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

Big data technologies



Data Management should abstract developers from technology change

Two technologies on Big Data Management

- Cloud computing
- Hadoop and MapReduce

Why we use cloud computing?

Why we use cloud computing?

Case 1:

Write a file

Save

Computer down, file is lost

Files are always stored in cloud, never lost

Why we use cloud computing?

Case 2:

Use MS Word --- download, install, use

Use Skype --- download, install, use

Use C++ IDE --- download, install, use

……

Get the serve from the cloud

What is cloud and cloud computing?

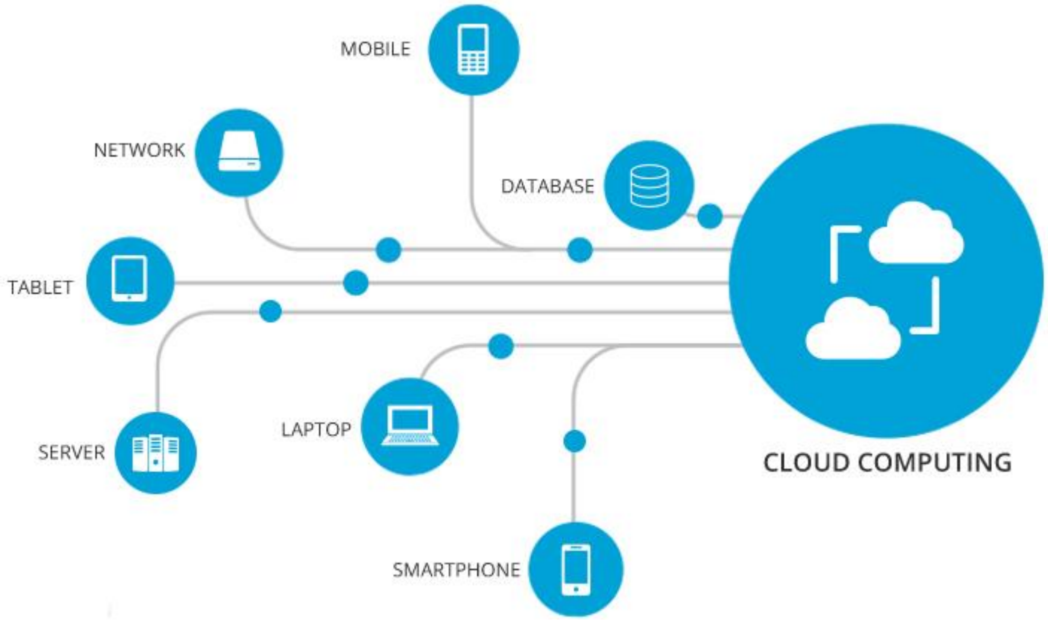
Cloud

Demand resources or services over Internet

scale and reliability of a data center.

What is cloud and cloud computing?

Cloud computing is a style of computing in which **dynamically scalable** and often **virtualized** resources are provided as a serve over the Internet.



Characteristics of cloud computing

- **Virtual.**

  software, databases, Web servers, operating systems, storage and networking as virtual servers.

- **On demand.**

  add and subtract processors, memory, network bandwidth, storage.

Types of cloud service

SaaS
Software as a Service

PaaS
Platform as a Service

IaaS
Infrastructure as a Service

SaaS

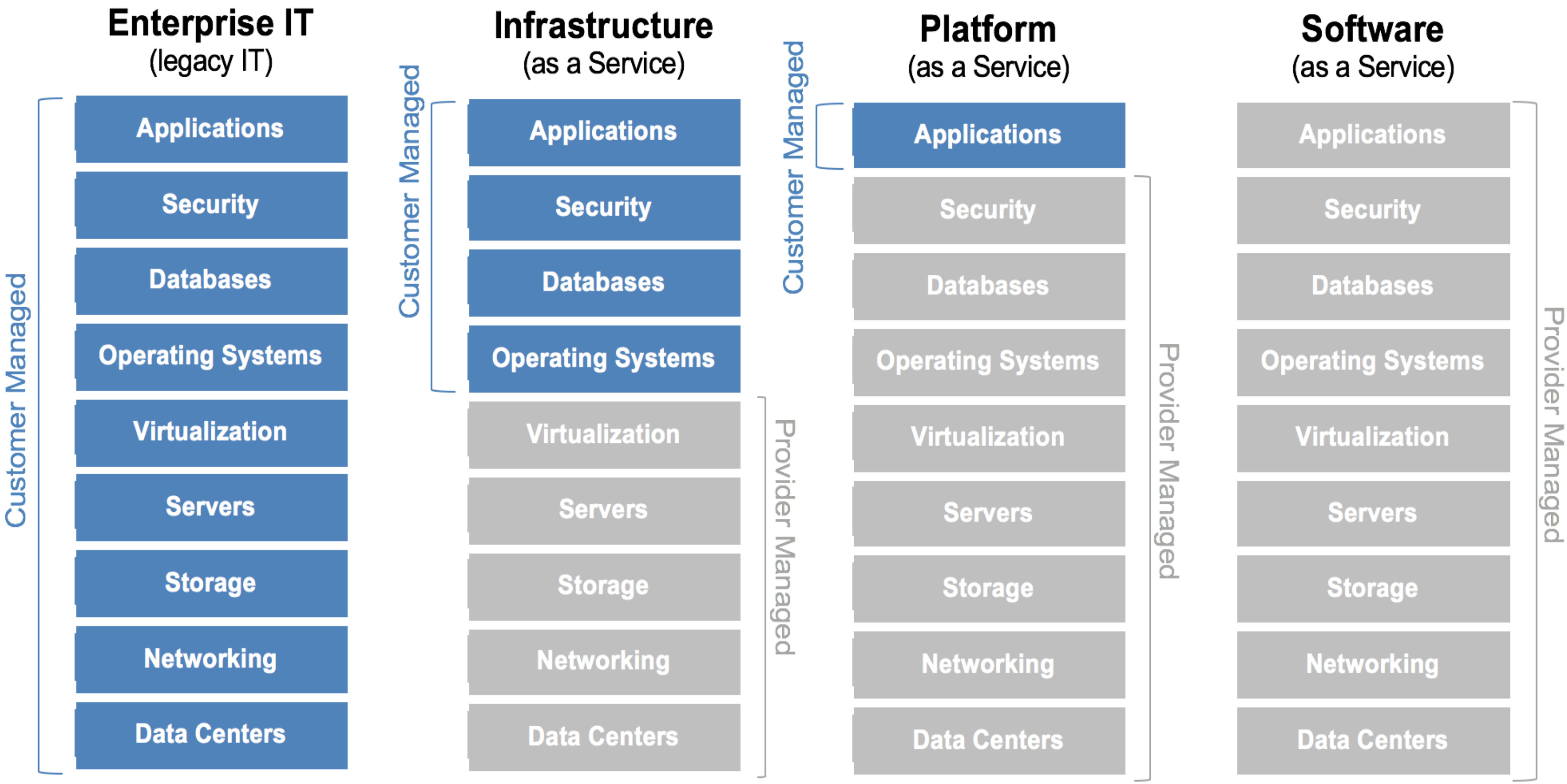
- Software as a service
- Operating environment largely irrelevant, fully functional applications provided, e.g. CRM, ERP, email

PaaS

- Platform as a service
- Operating environment included, e.g. Windows/.NET, Linux/J2EE, applications of choice deployed

IaaS

- Infrastructure as a service
- Virtual platform on which required operating environment and application are deployed
- Includes storage as a service offerings


Enterprise IT
(legacy IT)
Customer Managed
Applications
Security
Databases
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers
Infrastructure
(as a Service)
Customer Managed
Applications
Security
Databases
Operating Systems
Provider Managed
Virtualization
Servers
Storage
Networking
Data Centers
Platform
(as a Service)
Customer Managed
Applications
Provider Managed
Security
Databases
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers
Software
(as a Service)
Provider Managed
Applications
Security
Databases
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers

Two technologies on Big Data Management

- Cloud computing

- Hadoop and MapReduce

What is Hadoop?

- Apache top level project, open-source implementation of frameworks for reliable, scalable, distributed computing and data storage.

Google Origins

2003

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

2004

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

2006

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com

Google, Inc.

Abstract

gtable is a distributed storage system for managing
tured data that is designed to scale to a very large
petabytes of data across thousands of commodity
rs. Many projects at Google store data in Bigtable,
ding web indexing, Google Earth, and Google Fi-
e. These applications place very different demands
igtable, both in terms of data size (from URLs to

achieved scalability and high performance, but Big
provides a different interface than such systems. Big
does not support a full relational data model; inste
provides clients with a simple data model that sup
dynamic control over data layout and format, an
lows clients to reason about the locality properties o
data represented in the underlying storage. Data i
dexed using row and column names that can be arbi
strings. Bigtable also treats data as uninterpreted str

Hadoop's Developers

2005: Doug Cutting and Michael J. Cafarella developed Hadoop to support distribution for the Nutch search engine project.

The project was funded by Yahoo.

2006: Yahoo gave the project to Apache Software Foundation.

Some Hadoop Milestones

- **2008 - Hadoop Wins Terabyte Sort Benchmark (**sorted 1 terabyte of data in 209 seconds, compared to previous record of 297 seconds)

- **2010 - Hadoop's Hbase, Hive and Pig subprojects completed**, adding more computational power to Hadoop framework

- **2013 - Hadoop 1.1.2 and Hadoop 2.0.3 alpha.**

  - Ambari, Cassandra, Mahout have been added

- **2016 - Hadoop 3.0.0 Alpha-1**
-

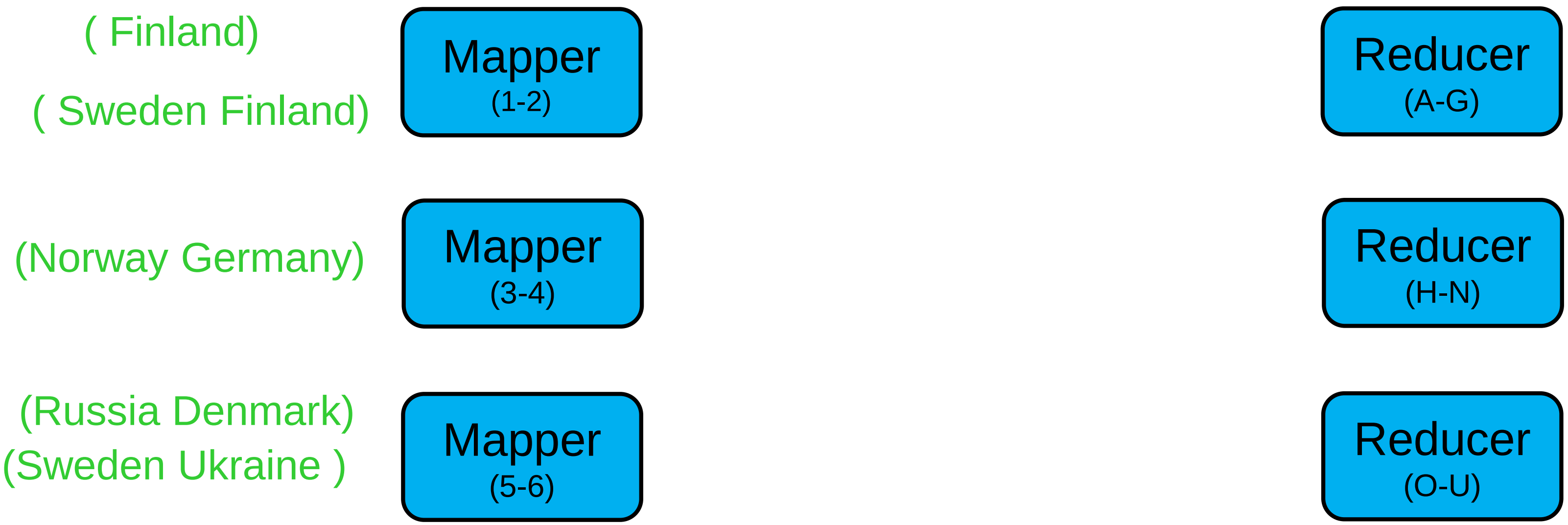
Introduction to MapReduce

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

MapReduce: Insight

- ”Consider the problem of counting the number of frequency of each word in a large collection of documents”
- Word-count problem

Simple example: Word count



1 Each mapper receives some of documents as input

Simple example: Word count

(Finland)

(Sweden Finland)

Mapper (1-2)

(Finland, 1)

(Sweden, 1), (Finland, 1)

Reducer (A-G)

(Norway Germany)

Mapper (3-4)

(Norway, 1), (Germany,1)

Reducer (H-N)

(Russia Denmark)

(Sweden Ukraine)

Mapper (5-6)

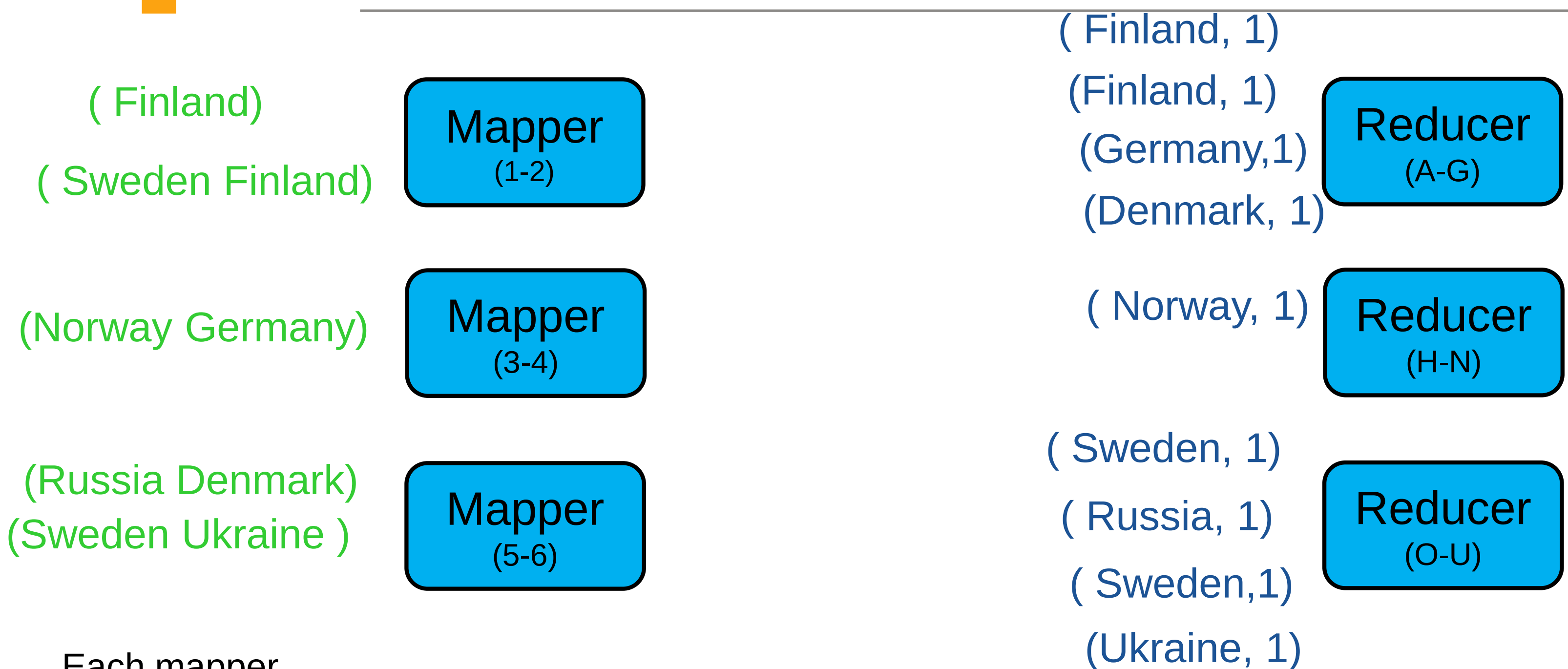
(Russia, 1), (Denmark, 1)

(Sweden,1),(Ukraine, 1)

Reducer (O-U)

1. Each mapper receives some of documents as input
2. Mappers process the KV-pairs.

Simple example: Word count



1. Each mapper receives some of documents as input
2. Mappers process the KV-pairs.
3. Each KV-pair output by the mapper is sent to the reducer

Simple example: Word count

(Finland)

(Sweden Finland)

(Norway Germany)

(Russia Denmark)
(Sweden Ukraine)

Mapper (1-2)

Mapper (3-4)

Mapper (5-6)

(Denmark, 1)
(Finland, 1)
(Finland, 1)
(Germany,1)

Reducer (A-G)

(Norway, 1)

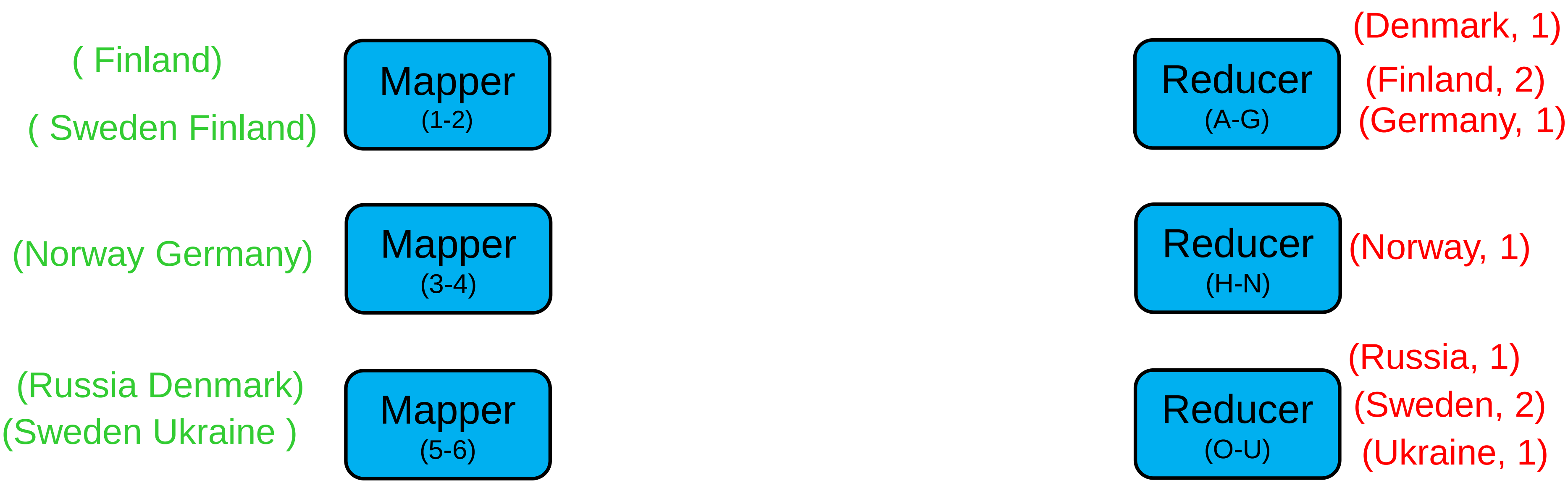
Reducer (H-N)

(Russia, 1)
(Sweden,1)
(Sweden, 1)
(Ukraine, 1)

Reducer (O-U)

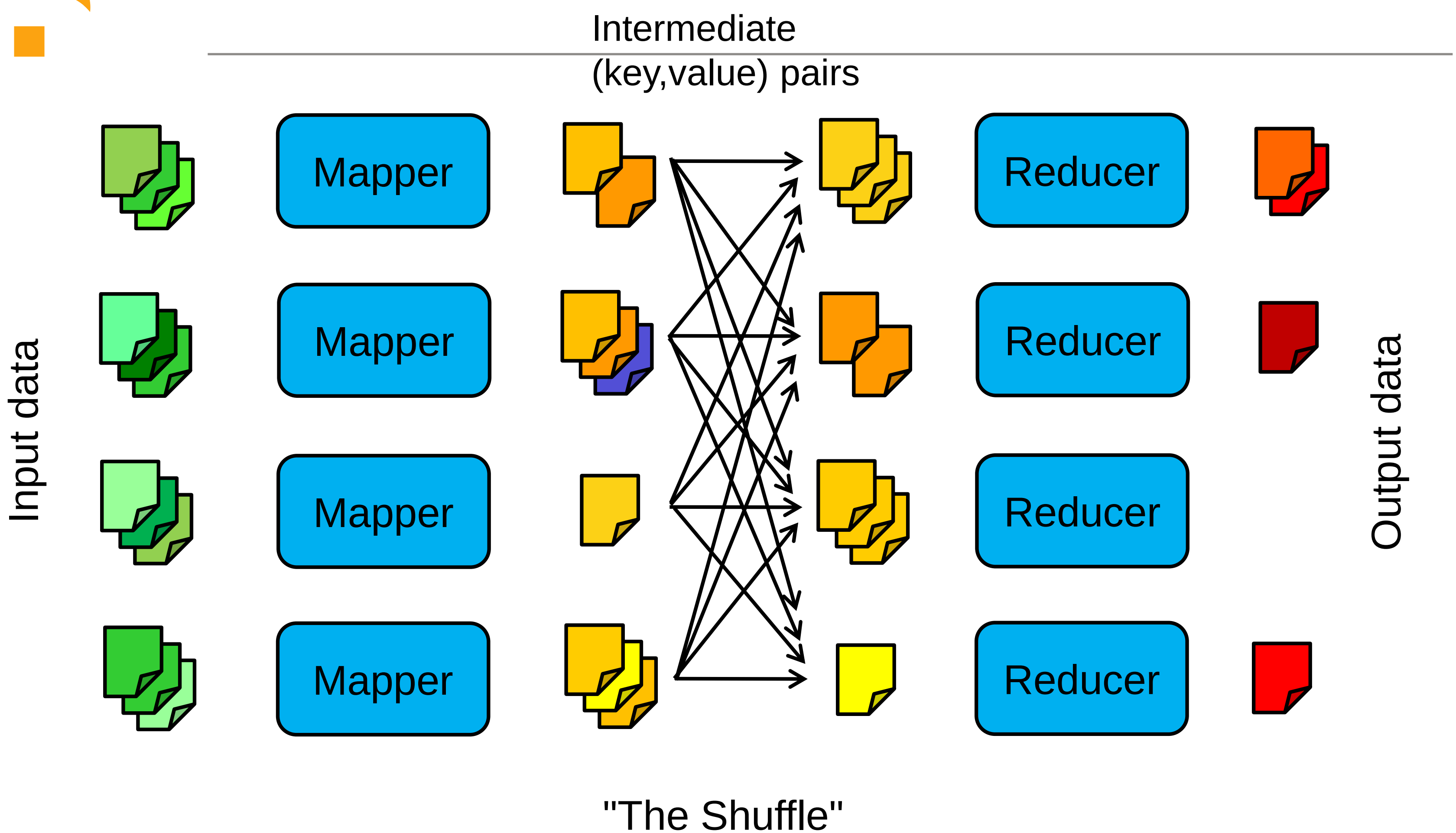
1. Each mapper receives some of documents as input
2. Mappers process the KV-pairs.
3. Each KV-pair output by the mapper is sent to the reducer
4. The reducers sort their input by key

Simple example: Word count



1. Each mapper receives some of documents as input
2. Mappers process the KV-pairs.
3. Each KV-pair output by the mapper is sent to the reducer
4. The reducers sort their input by key
5. The reducers process their input

MapReduce dataflow

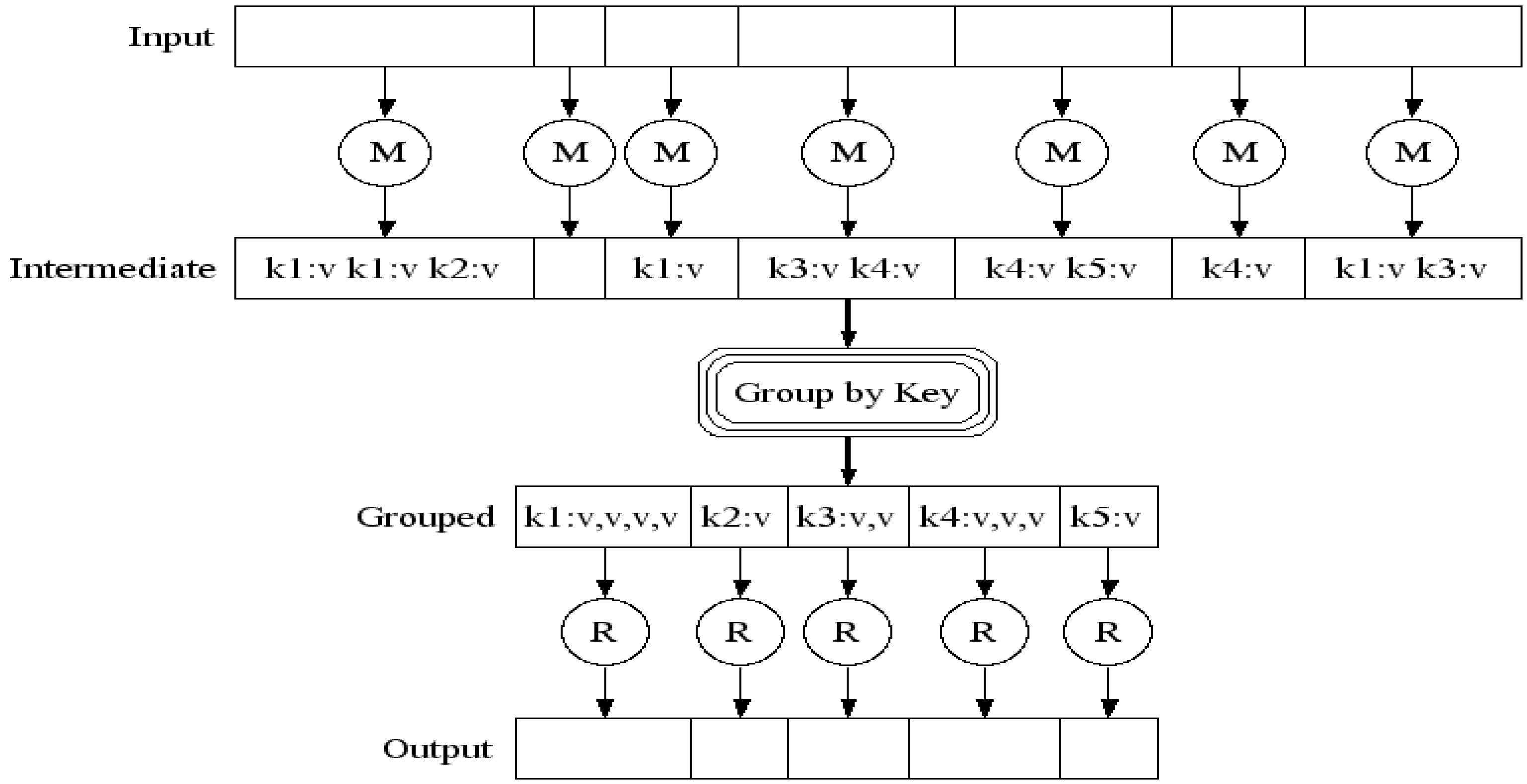


Pseudo-code

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

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

MapReduce: Example



Outline

- Introduction to Big Data
- Cloud computing
- MapReduce programming model
- Our research on multi-model databases

A grand challenge on Variety

- Big data: Volume, Variety, Velocity, Veracity
- Variety: tree data (XML, JSON), graph data (RDF, property graphs, networks), tabular data (CSV), temporal and spatial data, text

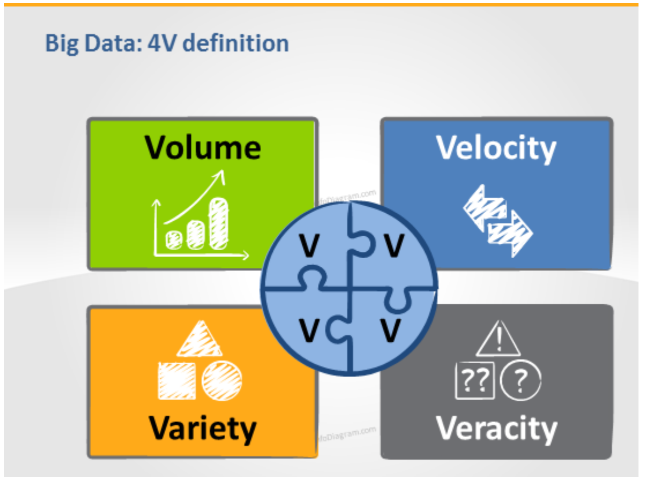


Photo downloaded from: https://blog.infodiagram.com/2014/04/visualizing-big-data-concepts-strong.html

NoSQL database types

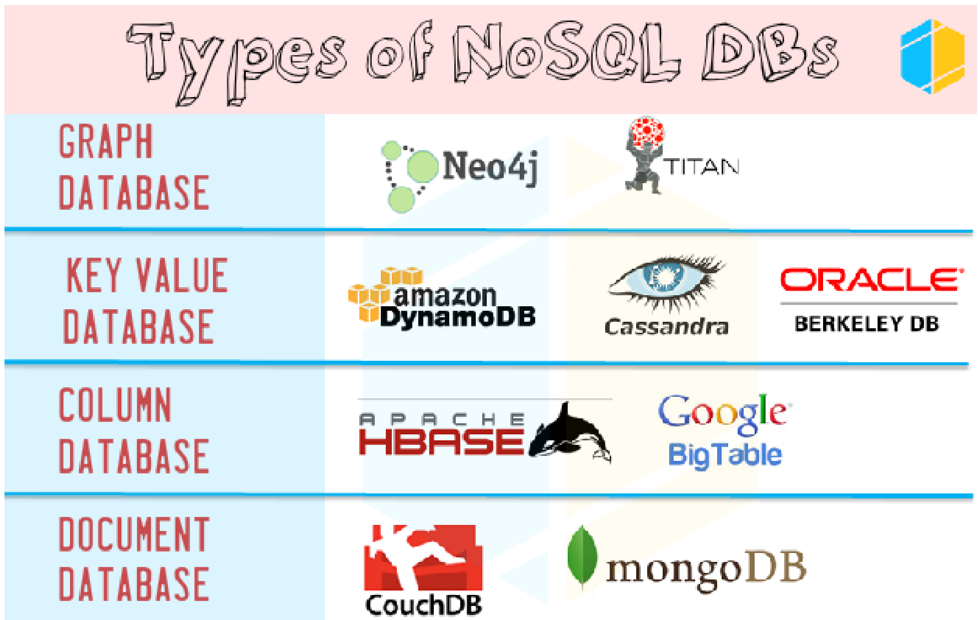
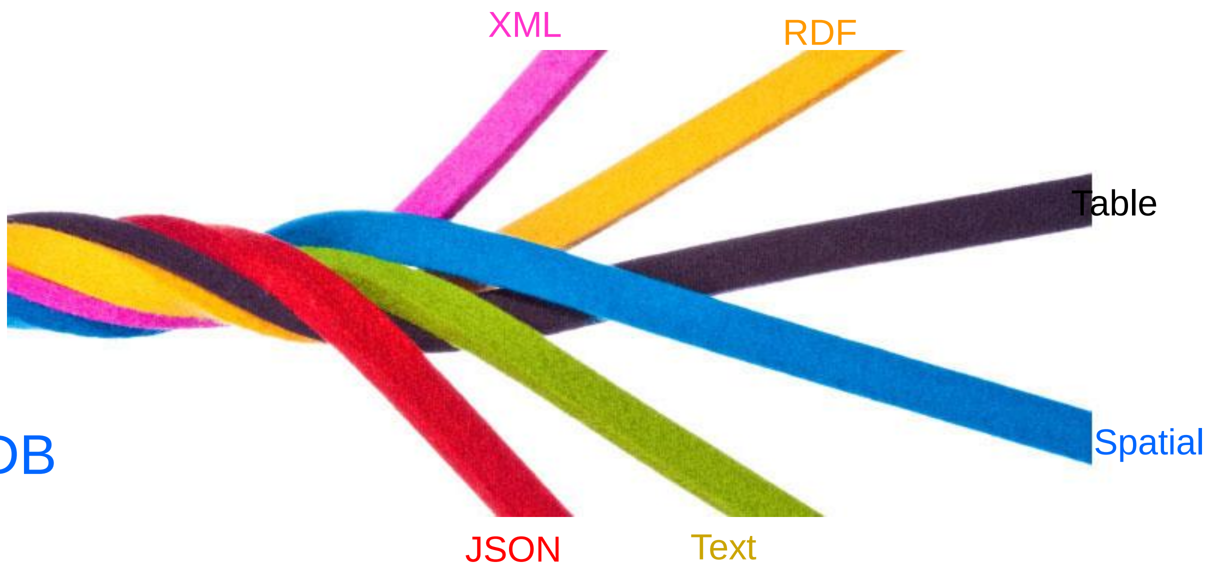


Photo downloaded from: http://www.vikramtakkar.com/2015/12/nosql-types-of-nosql-database-part-2.html

Multi-model DB

- One unified database for multi-model data



Multi-model DB

Multi-model databases

- A multi-model database is designed to support multiple data models against a single, integrated backend.

- Document, graph, relational, and key-value models are examples of data models that may be supported by a multi-model database.

Conclusion

Big data era: Volume, Variety, Velocity, Veracity

Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a serve over the Internet.

MapReduce is a software programming model for distributed big data processing

Task on data analysis for computer linguistic model

A data processing task for computational linguistic model.

Each group will be given an article, and the students need to complete the following three steps to visualize and analyze the document.

References

(1) Jinchuan Chen, Yueguo Chen, Xiaoyong Du, Cuiping Li, Jiaheng Lu, Suyun Zhao, Xuan Zhou: Big data challenge: a data management perspective. Frontiers Comput. Sci. 7(2): 157-164 (2013)

(2) Yu Liu, Jiaheng Lu, Hua Yang, Xiaokui Xiao, Zhewei Wei: Towards Maximum Independent Sets on Massive Graphs. PVLDB 8(13): 2122-2133 (2015)

(3) Jiaheng Lu, Chunbin Lin, Wei Wang, Chen Li, Xiaokui Xiao: Boosting the Quality of Approximate String Matching by Synonyms. ACM Trans. Database Syst. 40(3): 15:1-15:42 (2015)

(4) Juwei Shi, Jia Zou, Jiaheng Lu, Zhao Cao, Shiqiang Li, Chen Wang: MRTuner: A Toolkit to Enable Holistic Optimization for MapReduce Jobs. PVLDB 7(13): 1319-1330 (2014)

References

(5) Tao Guo, Xin Cao, Gao Cong, Jiaheng Lu, Xuemin Lin: Distributed Algorithms on Exact Personalized PageRank. SIGMOD Conference 2017: 479-494

(6) Jiaheng Lu, Irena Holubová: Multi-model Data Management: What's New and What's Next? EDBT 2017: 602-605

(7) Yu Liu, Bolong Zheng, Xiaodong He, Zhewei Wei, Xiaokui Xiao, Kai Zheng, Jiaheng Lu: ProbeSim: Scalable Single-Source and Top-k SimRank Computations on Dynamic Graphs. PVLDB 11(1): 14-26 (2017)

(8) Jiaheng Lu, Zhen Hua Liu, Pengfei Xu, Chao Zhang: UDBMS: Road to Unification for Multi-model Data Management. CoRR abs/1612.08050 (2016)

References

(9) Jiaheng Lu, Chunbin Lin, Wei Wang, Chen Li, Haiyong Wang: String similarity measures and joins with synonyms. SIGMOD Conference 2013: 373-384

(10) Jiaheng Lu, Pierre Senellart, Chunbin Lin, Xiaoyong Du, Shan Wang, Xinxing Chen: Optimal top-k generation of attribute combinations based on ranked lists. SIGMOD Conference 2012: 409-420

(11) Jiaheng Lu, Ying Lu, Gao Cong: Reverse spatial and textual k nearest neighbor search. SIGMOD Conference 2011: 349-360

(12) Jiaheng Lu, Jialong Han, Xiaofeng Meng: Efficient algorithms for approximate member extraction using signature-based inverted lists. CIKM 2009: 315-324


Thanks!


Q&A