Big Data Systems

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Big Data Systems

A 360-degree Approach

Jawwad Ahmed Shamsi
Muhammad Ali Khojaye

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First edition published 2021
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data

ISBN: 978-1-4987-5270-1 (hbk)
ISBN: 978-0-3677-5523-2 (pbk)
ISBN: 978-0-429-15544-4 (ebk)

Typeset in Computer Modern font
by KnowledgeWorks Global Ltd.

Jawwad A. Shamsi would like to dedicate this book to his parents, his wife, and his children, all of whom have offered unconditional love and support.

Muhammad Ali Khojaye would like to dedicate this book to his parents, to his wife, and to his son Rayan. He also like to thank them for their enormous amount of support during this long process of writing this book.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Contents

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Preface

In a simplistic term, a system which handles big data is considered as a big data system. This handling could include different services such as computation, storage, and networking. The term big data refers to enormous size of data, which could challenge the conventional systems for storing and computation. While there is no standard range, which can classify big data, normally systems which can handle a few terabytes of data can be considered as big data systems.

Large amount of data is useful in many ways. Organizations can get deep insights about many aspects and operations. End users can get enhanced services. For instance, a large search engine can offer customized search results to its users. Similarly, a government can enhance its security system through smart surveillance systems based on big data. Both these examples reflect the emerging potential of big data. While the benefits of big data are still emerging, it is necessary to understand and study different systems and platforms which can facilitate big data.

Objectives

The purpose of this book is to study different concepts related to big data. It has following major objectives:

- To elucidate different platforms for processing of big data systems.
- To highlight the role of cloud computing in computing and storing big data systems.
- To explain security and privacy issues in big data.
- To provide an overview of different networking technologies in big data systems.
- To describe pros and cons of different computational platforms for big data.
- To elaborate on different case studies of big data.
- To enlighten programming and algorithmic techniques available for big data processing.

Organization

The book is organized into five parts and 14 chapters:

Section 1: The first part covers introductory concepts. It consists of three chapters. The first chapter covers concepts related to fundamental concepts of big data including difference between analytical and transactional systems and requirements and challenges of big data. The second chapter elaborates on architectural and organizational concepts related to big data. The chapter explains the difference between lambda and kappa architectures. It also highlights cluster computing and different organization schemes for clusters. The last chapter in this section, chapter 3, is focused on cloud computing and virtualization. Cloud provides an important platform for big data and the chapter is focused on elaborating this requirement.

Section II: The second section is focused on elaborating efficient platforms for storing and processing big data. Chapter 4 explains various topics related to Hadoop and MapReduce. This chapter also covers programming examples using MapReduce. In addition, two important components of Hadoop, i.e., HDFS and HBase are explained. Chapter 5 explains a few important limitations of Hadoop v1, and describes how Hadoop v2 addresses these limitations. This chapter also covers YARN, Pig, Hive, Dremel, Impala, Drill, Sqoop, and Ambari. Chapter 6 explains Spark and its different components. It elaborates on how Spark is useful in solving a few major big data problems. The chapter also includes programming examples. Chapter 7 describes NoSQL systems. These include, column-base, key-value base, document oriented, and graph databases. The chapter covers illustrative examples for enhanced explanation. Chapter 8 is focused on introducing the topic of NewSQL systems. The chapter has introductory coverage of four different NewSQL systems. These include VoltDB, NuoDB, Spanner, and HRDBMS.

Section III: Section III elaborates on networking, security, and privacy for big data systems. There are several illustrative examples in this part. Chapter 9 explains various topics related to networking for big data. This chapter highlights different requirements for efficient networks for big data systems and provides an overview of existing networking solutions. Chapter 10 explains security requirements and solutions for big data. It includes various topics such as requirements, attack types and mechanisms, attack detection, and prevention. The last chapter in this section, chapter 11, provides an overview of privacy concerns in big data and explains existing solutions in ensuring privacy.

Section IV: The fourth part of this book includes computation for big data. It contains chapter 12 and chapter 13. Chapter 12 explains HPC solutions, which are being utilized by big data systems. This chapter describes the functionality of GPUs, TPUs, and supercomputing. Chapter 13 introduces the topic of deep learning. It covers explanations of various deep learning solutions. These include Feed Forward Network, RNN, and CNN. The chapter explains various examples of big data which can get leverage from deep learning solutions.

Section V: Section V contains chapter 14. This chapter covers various case studies on big data. These include organizational solutions on a Facebook, Uber, LinkedIn, Microsoft, and Google. The chapter also highlights a few open issues on big data systems.

Target Audience

The book adopts an example-centric approach, where various concepts have been explained using illustrative examples and codes. The book can be used either as a text book or as a reference book. It can be adopted for various courses such as cloud computing and big data systems. Moreover, individual chapters of the book can be used as reference for different courses related to networking, security, privacy, high-performance computing, and deep learning.

Contact Us

If you have any question or comment about this book, please send email to bigdataquestions@gmail.com

We have a web site for the book, where we list examples, errata, and any additional information. For access, please visit our web page at https://sites.google.com/view/bigdatasystems.

Jawwad Ahmed Shamsi and Muhammad Ali Khojaye

Author Bios

Jawwad A. Shamsi completed B.E. (Electrical Engineering) form NED University of Engineering and Technology, Karachi in 1998. He completed his MS in Computer and Information Sciences from University of Michigan-Dearborn, MI, USA in 2002. In 2009, he completed his PhD. from Wayne State University, MI, USA. He has also worked as a Programmer Analyst in USA from 2000 to 2002. In 2009, he joined FAST- National University of Computer and Emerging Sciences (NUCES), Karachi. He has served as the head of computer science department from 2012 to 2017. Currently, he is serving as a Professor of Computer Science and Director of the Karachi Campus. He also leads a research group – syslab (http://syslab.khi.nu.edu.pk). His research is focused on developing systems which can meet the growing needs of scalability, security, high performance, robustness, and agility. His research has been funded by different International and National agencies including NVIDIA and Higher Education Commission, Pakistan.

Muhammad Ali Khojaye has more than a decade of industrial experience ranging from architecting cloud-native applications to distributed systems design, continuous integration/delivery, and infrastructure. His current technical interests revolve around big data, cloud, containers, and systems architecture for distributed platforms. He holds a master's degree in Computer Science from the University of Leicester. Born in the mountain village of Chitral Pakistan, Ali currently lives in Glasgow with his wife and son. When he is not at work, Ali enjoys cycling, traveling, and spending time with family and friends.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Acknowledgments

The authors would like to acknowledge the contributions from Muhammad Nouman Durrani, Ali Akber, and Bara Raza Mangnani from National University of Computer and Emerging Sciences. The authors are also thankful to Abid Hasnain from Visionet Systems and Narmeen Bawany from Jinnah University for Women.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

List of Examples

List of Figures

List of Tables

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

I

Introduction

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Introduction to Big Data Systems

CONTENTS

BIG DATA has been increasingly used in our daily lives. From social networks to mobile applications, and internet search, a huge amount of data is being generated, collected, and processed. The purpose of this chapter is to explain the fundamental concepts related to big data that are useful in understanding the functionality and execution of big data systems.

1.1 INTRODUCTION: REVIEW OF BIG DATA SYSTEMS

In the year 2006, LinkedIn – the social networking giant, started analyzing profiles of its users by suggesting people they may know. The rationale behind this feature was to encourage users to expand their social network based on their interest and give them relevant suggestions. Through this feature, LinkedIn observed that most of its suggestions in inviting people were successful [352]. Similarly, in the year 2012, for the US presidential elections, President Obama's campaign experienced massive boost and success through predictive analysis using a big dataset consisting of voter's profiles, their likes, and their patterns [207].

The two examples cited above highlights the enormous potential of analyzing, linking, and extracting useful information from large data. The science of prediction requires a massive amount of data and methodological linkage of different attributes on big data.

While big data carries huge potential for information extraction; it requires thorough understanding and comprehension of underlying systems (software and hardware) that are used for storing, processing, linking, and analyzing.

1.1.1 Purpose of the Book

The purpose of this book is to explain different systems that are used for storing, processing, and analyzing big data. The book adopts an example-oriented approach in which each chapter contains examples and illustrations for better explanation.

1.2 UNDERSTANDING BIG DATA

The term big data has gained popularity [298]. It refers to the huge amount of data such that managing, analyzing, and understanding data at volumes and rates that push the frontiers of current technologies [225].

1.2.1 Important Characteristics of Big Data

Researchers have highlighted a few important characteristics of big data [369]. These are often referred to as five V's of big data.

1. **Volume**: Big data refers to the massive volume of data such that the amount of data challenges the storage and processing requirements. While there is no specific distinction about the volume of data, normally the volume could vary from Terabytes (10^{12}) to exabytes (10^{18}) and beyond.

2. **Velocity**: Data is being generated at a very fast pace. The high rate of data generation signifies the importance of data. The high velocity of data can be assessed by the fact that a large proportion of data being used belong to the recent past.

3. **Variety**: Data under consideration could be obtained from numerous sources such as web logs, Internet of Things (IoT) devices, URLs, user tweets, and search patterns etc. Similarly, data could have different formats such as Comma Separated Values (CSV), tables, text documents, and graphs. Further, it could either be structured, semi-structured, or unstructured.

4. **Veracity**: Data may vary in terms of veracity; i.e., data under consideration may be inconsistent or it may be highly consistent across all replicas; it may be useless or it may be of high value. Veracity refers to the trustworthiness, accuracy, or authenticity of data.

5. **Value**: Data must be of high value; i.e., stale data has limited value.

1.2.2 Big Data: How Big Is Enough?

One of the fundamental questions in big data is that for a given big data problem in consideration, how much data is enough? That is, how much data is needed to be analyzed in order to compute the result? The answer to this question is not trivial.

Often in data analysis, data is sampled to process results. For instance, opinion polls are based on data samples. In a similar context, gender-wise assessment and population are based on data sampling. Sampling induces likelihood of error – a scenario, where the data sample collected may not reflect true results.

An example of data sampling error could be derived from Google Flu Trends (GFT) [236]. In 2009, Google tracked the spread of Flu in the United States. The prediction was based on GFT, the search trends available on Google. The prediction was so successful that it outnumbered the prediction from the Center for Disease Control (CDC). However, in Feb

2013, a similar prediction appeared as erroneous. It was observed that the GFT prediction was overstated by a factor of more than two. The problem was that Google's algorithm was simply considering the search terms on the Google Search Engine. It was assumed that all the related searches made on Google are related to spread of flu. The Google team was unable to find the correlation between search terms and flu.

In a similar context, data sampling error could be induced during an election campaign. For instance, data for election tweets may favor a specific candidate. However, it may be the case that voters of the candidate are pro-active on social media as compared to the voters of other candidates. Similarly, sample size in any big data problem could have its own biases.

Determining the correct size of data for a given big data problem is not trivial. In addition, collecting or gathering the complete data is also an issue. Many experts believe that in the case of big data, N=ALL is a good reference point for data analysis [191]. That is, all the data needs to be analyzed. However, collecting such a large amount of data or determining what is included in N=ALL is not trivial. Therefore, in many cases, a big data problem is analyzed on **found data** – a term which is referred to donate the data which has found for analysis.

While collecting more data is often more useful for analysis; it is not necessary that more data would yield improved results. In this context, relevance of data being collected is also important [350].

1.3 TYPE OF DATA: TRANSACTIONAL OR ANALYTICAL

An important question is what type of data can be referred to as big data. In the literature, two types of systems have been mentioned:

1. **Transactional Systems**: These are the types of systems which support transaction processing. Subsequently, these systems adhere to ACID (Atomicity, Consistency, Isolation, and Durability) properties. They have proper schema and data for each transaction is uniquely identified.

2. **Analytical Systems**: Such systems do not necessarily hold ACID properties. Consequently, data does not necessarily adhere to a proper schema. It may have duplicates and missing values etc. Such systems are more appropriate for analyzing data.

Traditionally, the term big data has been associated for analytical systems – specifically due to the fact because such systems do not require strong consistency and have schema-less data with duplicates, multi-formatting, and missing values. However, as we will study in chapter 8 big data systems have evolved to include transactional systems bearing ACID properties.

1.3.1 CAP Theorem

The database community initially argued that lack of consistency and normalization in big data systems was a major limitation. However, later it was felt that big data systems do not need to maintain strong consistency and they can exploit CAP theorem to avail the benefit of relaxed consistency and improved performance.

CAP theorem was proposed by Eric Brewer [108]. It explains important characteristics for distributed systems. The fundamental idea of the CAP theorem is that in a distributed system, there are three important characteristics, i.e., Consistency, Availability, and Partition Tolerance (CAP). CAP theorem states that in case of a partition (or network failure) both consistency and availability cannot be offered together. That is, when network failure

occurs and the network is partitioned, a distributed system can either offer consistency or availability but not both. Note that when there is no network failure, a distributed system can offer both availability and consistency together. Example 1.1 explains the concept of CAP theorem.

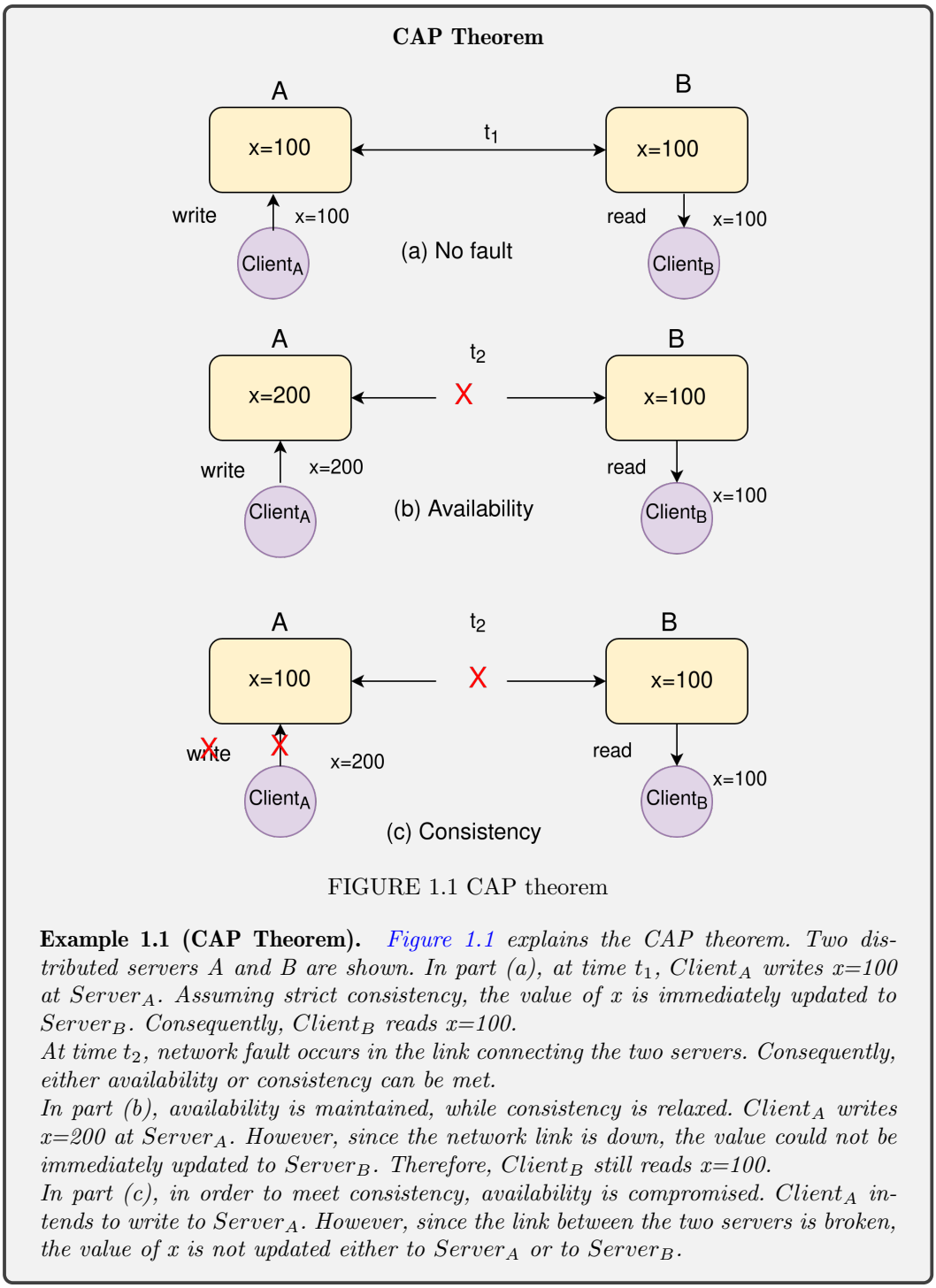

FIGURE 1.1 CAP theorem

Example 1.1 (CAP Theorem). *Figure 1.1 explains the CAP theorem. Two distributed servers A and B are shown. In part (a), at time t_1, $Client_A$ writes x=100 at $Server_A$. Assuming strict consistency, the value of x is immediately updated to $Server_B$. Consequently, $Client_B$ reads x=100.*
At time t_2, network fault occurs in the link connecting the two servers. Consequently, either availability or consistency can be met.
In part (b), availability is maintained, while consistency is relaxed. $Client_A$ writes x=200 at $Server_A$. However, since the network link is down, the value could not be immediately updated to $Server_B$. Therefore, $Client_B$ still reads x=100.
In part (c), in order to meet consistency, availability is compromised. $Client_A$ intends to write to $Server_A$. However, since the link between the two servers is broken, the value of x is not updated either to $Server_A$ or to $Server_B$.

Many big data systems exploit CAP theorem to provide availability at the cost of consistency. However, consistency is not necessarily compromised for availability. It can also be compromised for latency [74]. Systems which have relaxed consistency requirements are normally considered appropriate for data analytics as they do not necessarily maintain ACID properties. Maintaining ACID properties in the context of big data are challenging due to massive size and distributed nature of data. Therefore, by-n-large, big data systems have been normally associated with analytical systems. Figure 1.2 illustrates the difference between transactional systems and analytical systems.

1.3.2 ACID vs BASE

For distributed systems, meeting ACID guarantees is really challenging. Therefore, many big data systems employ BASE properties [87, 299]. BASE is an acronym for Basically Available Soft state Eventual consistency. BASE implies that in case of network failure, big data systems tend to compromise on consistency in order to provide availability. The main focus of such systems is to ensure availability, whereas eventual consistency model is followed.

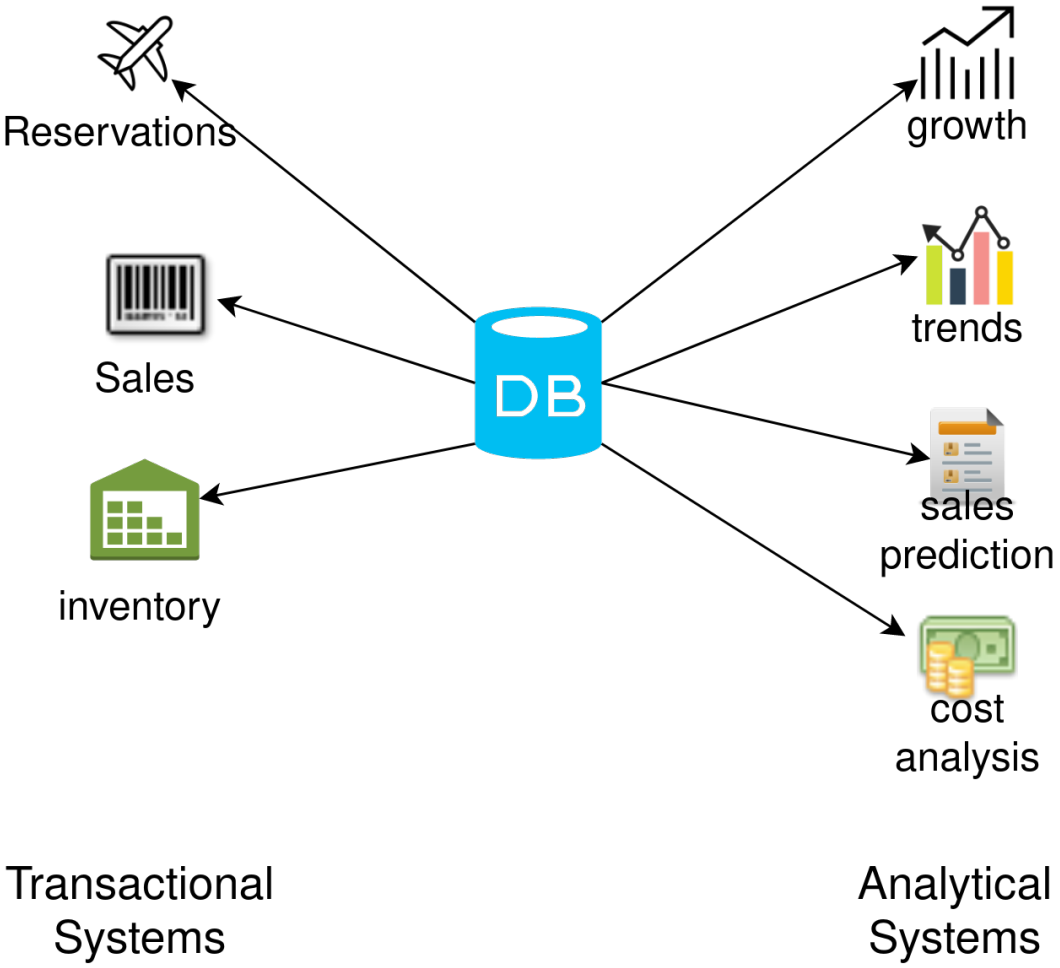

FIGURE 1.2 Transactional systems vs. analytical systems

1.4 REQUIREMENTS AND CHALLENGES OF BIG DATA

For big data systems, there are a few specific research challenges. These are needed to be catered:

1. **Scalability**: The foremost requirement for big data systems is to provide massive capability for processing and storage of huge amounts of data. Scalability should be achieved without any noticeable degradation in performance.

2. **Availability and Fault Tolerance**: An efficient big data system should be able to tolerate faults. Faults could either be transient such as network congestion, CPU availability, and packet loss, or they could be persistent such as disk failure, power faults, and network outages.

3. **Efficient Network Setup**: As big data system consists of a large number of machines and workstations, efficient networking setup is an important requirement. The network should be capable of providing access to big data, with low communication latency. Network setup should facilitate building big data systems through both Local Area Network (LAN) and Wide Area Network (WAN).

4. **Flexibility**: Big data systems may contain data from multiple sources such as textual data, images, videos, and graphs. Similarly, data can be assessed and analyzed through multiple means including visualizations, raw data, aggregated data, and queries. Big data systems should facilitate flexible mechanisms for accessing and storing big data systems.

5. **Privacy and Access Control**: As big data systems gather data from a large number of sources, privacy and access control are likely to be one of the major concerns. Questions such as which data should be made public, what information should be assessed, and who has the ownership of data are important and needed to be identified.

6. **Elasticity**: In a big data system, the number of users varies over time. An efficient system should be able to meet user's needs. Elasticity refers to the capability of the system in meeting these needs.

7. **Batch Processing and Interactive Processing** With the passage of time, big data systems have expanded from batch processing to interactive processing. For capable big data systems, possessing the ability to analyze and process big data in batch mode as well streaming mode is necessary.

8. **Efficient Storage**: As data is replicated in big data systems, efficient mechanisms for replication and storage are significant in reducing the overall cost.

9. **Multi-tenancy**: Big data systems are accessed by multiple users at a time. Multi-tenancy refers to the capability of the system in providing fair, persistent, and isolated services to the users of big data.

10. **Efficient Processing**: As data is massive, efficient algorithms, techniques, and hardware are needed for large-scale computation of big data. In this context, effective techniques for parallelization are also significant. Similarly, iterative computation for machine learning and data analytics are also important.

11. **Efficient Scheduling**: With multiple parallel tasks and concurrent users, methods and techniques for efficient scheduling are needed.

The above set of requirements are significant for big data systems. There are numerous solutions which have been developed to cater these needs. Over the course of this book, we will study various solutions pertaining to these challenges.

1.5 CONCLUDING REMARKS

Big data systems implement extensive analysis of the massive amount of data for prediction and analysis. These systems entail specific challenges and considerations related to system architecture, fault tolerance, computation and processing, replication, consistency, scalability, and storage. This book is aimed at explaining existing solutions which address these challenges.

The remainder of this book is organized as follows: Chapter 2 elaborates on architectural and organizational concepts related to big data. Chapter 3 describes cloud computing. These two concepts are well established for big data systems. Chapter 4 describes Hadoop – a popular platform for storing and processing big data. Chapter 5 explains a few important limitations of Hadoop v1, and describes how Hadoop v2 addresses these limitations whereas chapter 6 explains Spark. In chapters 7 and 8, we discuss NoSQL and NewSQL systems. Chapter 9 describes networking issues and solutions for big data systems. Chapter 10 explains security requirements and solutions for big data whereas chapter 11 discusses privacy issues in big data. Chapter 12 highlights high-performance computing systems and their impact for big data systems whereas chapter 13 introduces deep learning – an emerging concept for analytics in big data systems. Chapter 14 is the last chapter of the book. It describes different case studies of big data systems. It also highlights a few open issues on big data systems.

1.6 FURTHER READING

Examples of big data and its impact in analytics and predictions have been discussed in references [207, 236].

[191] provides further discussion on analytics and the size challenges for big data.

CAP Theorem and its impact on design of big data been discussed in many research papers. It was initially proposed in the year 2000 [111]. Later, in many research papers, different design principles of distributed systems have been explained [107–109, 178].

Difference between ACID and Base properties has been further elaborated in the literature. References [122, 294] explains these properties in detail.

Research Issues in big data systems have been discussed in detail in many research papers. These include architectural and hardware related issues, software, and platform related research, as well applications of big data [192, 238].

[208, 231] and [317] present a detailed discussion on all different challenges of big data.

1.7 EXERCISE QUESTIONS

1. Explain five V's of big data.
2. Explain CAP Theorem. How is it useful for big data?
3. Explain the difference between found data and all data.
4. What are the major challenges for big data systems?
5. What are ACID guarantees? Are they needed for big data systems?
6. Highlight major differences between transactional systems and analytical systems.

7. Explain major characteristics of BASE systems.
8. Explain how much data is sufficient to achieve high accuracy in big data systems?
9. Explain why Google flu trends were inaccurate in estimation?
10. Explain the difference between scalability and elasticity.

GLOSSARY

ACID: These are the set of properties which identify a database system. These stand for Atomicity, Consistency, Isolation, and Durability.

Analytical Systems: These types of systems are used to provide analytics on historically large volumes of data.

Atomicity: It is a database concept that a transaction either succeeds or fails in its entirety.

BASE: It stands for Basically Available Soft state Eventual consistency. These are the types of systems which provide relaxed consistency requirements than ACID.

CAP Theorem: It is a theorem which identifies a design model for distributed systems. It states that in case of a network partition (failure), a distributed system can either provide consistency or availability.

Cluster computing: It is a type of computing which allows multiple computers to work together to either solve common computing problems or provide large storage. It requires a cluster management layer to handle communication between the individual nodes and work coordination.

Data Visualization: It presents meaningful data graphically (from raw data) in order to understand complex big data.

Eventual Consistency: It is a type of consistency model, which support BASE model to the ACID model.

Fault Tolerant: It is a property of a system to make sure it recovers automatically even if certain parts of the system fail.

Processing: It refers to extracting valuable information from large datasets.

Reliability: The probability that a given system will perform its intended functions continuously and correctly in a specified environment for a specified duration.

Resiliency: A resilient system is one that can gracefully handle unexpected situations and bounce back from failures.

Transactional Systems: These are the types of systems which support transaction processing.

Bibliography

[1] Accumulo. https://accumulo.apache.org/.

[2] Amazon Hybrid Cloud. http://aws.amazon.com/directconnect/.

[3] Ambari. https://ambari.apache.org.

[4] An in-depth look at Googles first Tensor Processing Unit (TPU). https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu.

[5] Apache Drill – Schema-free SQL for Hadoop, NoSQL and Cloud Storage. https://drill.apache.org/.

[6] Apache Flink: Stateful Computations over Data Streams. https://flink.apache.org/.

[7] Apache Flume. https://flume.apache.org/.

[8] Apache Hadoop Distributed Copy – Distcp Guide. https://hadoop.apache.org/docs/r1.2.1/distcp.html.

[9] Apache Hadoop YARN. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.

[10] Apache HBase. http://hbase.apache.org/acid-semantics.html.

[11] Apache HBase Docs. http://hbase.apache.org/poweredbyhbase.html.

[12] Apache Kafka for Beginners. https://blog.cloudera.com/blog/2014/09/apache-kafka-for-beginners/.

[13] Apache Nifi. https://nifi.apache.org/.

[14] Apache Samza: A Distributed Stream Processing Framework. https://samza.apache.org/.

[15] Apache Spark Cluster Managers Yarn, Mesos and Standalone. https://data-flair.training/blogs/apache-spark-cluster-managers-tutorial/.

[16] Apache Storm. https://storm.apache.org/.

[17] BeyondCorp - Enterprise Security. https://cloud.google.com/beyondcorp/.

[18] Big Data at Uber. https://eng.uber.com/.

[19] Bigquery: Cloud Data Warehouse – Google Cloud. https://cloud.google.com/bigquery/.

[20] Boto Library for AWS. https://boto3.amazonaws.com/v1/documentation/api/latest/index.html.

[21] Data Breach Investigations Report 2016. http://www.verizonenterprise.com/resources/reports/rp_DBIR_2016_Report_en_xg.pdf.

[22] Deep Learning with Speech Recognition. https://www.coursera.org.

[23] Docker Documentation. https://www.docker.com.

[24] Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4j.

[25] Graphx Programming Guide. https://spark.apache.org/docs/latest/graphx-programming-guide.html.

[26] Handling Five Billion Sessions a Day. https://blog.twitter.com/engineering/en_us/a/2015/handling-five-billion-sessions-a-day-in-real-time.html.

[27] How to Read: Character Level Deep Learning. https://offbit.github.io/how-to-read/.

[28] Impala. https://impala.apache.org.

[29] Internet Small Computer System Interface. Technical report.

[30] Kafka. https://yahooeng.tumblr.com/post/109994930921/kafka-yahoo.

[31] Kafka Access. https://cwiki.apache.org/confluence/display/KAFKA/.

[32] Kafka Documentation. https://kafka.apache.org/documentation.html.

[33] Kafka Ecosystem at LinkedIn. https://engineering.linkedin.com/blog/2016/04/kafka-ecosystem-at-linkedin.

[34] Kafka Inside Keystone Pipeline. https://netflixtechblog.com/kafka-inside-keystone-pipeline-dd5aeabaf6bb.

[35] Kubernetes API. https://kubernetes.io/docs/reference/using-api/.

[36] Kubernetes Deployments. https://kubernetes.io/docs/concepts/workloads/controllers/deployment/.

[37] Kubernetes Ingress. https://kubernetes.io/docs/concepts/services-networking/ingress/.

[38] Kubernetes Objects. https://kubernetes.io/docs/concepts/overview/working-with-objects/.

[39] Kubernetes Pods. https://kubernetes.io/docs/concepts/workloads/pods/.

[40] Kubernetes Service. https://kubernetes.io/docs/concepts/services-networking/service/.

[41] LinkedIn Engineering. https://engineering.linkedin.com/.

[42] Linux Containers. https://linuxcontainers.org/.

[43] Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling.

[44] Machine learning is fun. https://medium.com/.

[45] Machine Learning Library MLlib Guide. https://spark.apache.org/docs/latest/ml-guide.html.

[46] MLlib: RDD-based API. https://spark.apache.org/docs/latest/mllib-guide.html.

[47] MPI Hello World Tutorial. https://mpitutorial.com/tutorials/.

[48] OpenMP Hello World Tutorial. https://www.dartmouth.edu.

[49] Pig Guide. https://pig.apache.org/docs/r0.17.0/func.html.

[50] Pig Optimizations. https://cwiki.apache.org/confluence/display/PIG/.

[51] Powered By Spark. http://spark.apache.org/powered-by.html.

[52] Presto – Distributed SQL Query Engine for Big Data. https://prestosql.io.

[53] Spark Documentation. https://spark.apache.org/docs.

[54] Spark ML online documentation. https://spark.apache.org/docs/latest/ml-pipeline.html.

[55] Spark Programming Guide. https://spark.apache.org/docs/latest/rdd-programming-guide.html.

[56] Spark SQL. https://spark.apache.org/docs/latest/sql-programming-guide.html.

[57] Spark Streaming. https://spark.apache.org/docs/latest/streaming-kafka-integration.html.

[58] Spark Streaming Example. http://spark.apache.org/.

[59] Spark Streaming Guide. https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html.

[60] Spark Structured Streaming Guide. https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html.

[61] Sqoop. https://sqoop.apache.org.

[62] Sqoop Docs. https://sqoop.apache.org/docs/.

[63] TensorFlow. https://www.tensorflow.org/.

[64] The Official Kubernetes Documentation. https://kubernetes.io/docs/home/.

[65] The Official SPARK Graphx Documentation. https://spark.apache.org/docs/latest/graphx-programming-guide.html.

[66] Theano Documentation. http://deeplearning.net/.

[67] Top Ten Supercomputers. https://www.networkworld.com/article/3236875/embargo-10-of-the-worlds-fastest-supercomputers.html.

[68] The Trouble with Kappa Architecture. https://www.linkedin.com/pulse/trouble-kappa-architecture-michael-segel.

[69] VMware Hybrid Cloud. http://www.vmware.com/products/vcloud-hybrid-service.

[70] VMware Private Cloud. http://www.vmware.com/cloud-computing/private-cloud.html.

[71] Xen Cloud Platform. http://www-archive.xenproject.org/products/cloudxen.html.

[72] Jans Aasman. Allegro Graph: RDF Triple Database. *Cidade: Oakland Franz Incorporated*, 17, 2006.

[73] Mohammad Aazam and Eui-Nam Huh. Fog Computing and Smart Gateway Based Communication for Cloud of Things. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, page 464–470. IEEE, 2014.

[74] Abadi. Problems with Cap and Yahoos Little Known NOSQL System. 2010. http://dbmsmusings.blogspot.com/2010/04/problems-with-cap-andyahoos-little.html.

[75] Martjn Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A System for Large-scale Machine Learning. In *OSDI*, volume 16, page 265–283, 2016.

[76] Tim Abels, Puneet Dhawan, and Balasubramanian Chandrasekaran. An Overview of Xen Virtualization. *Dell Power Solutions*, (8):109–111, 2005.

[77] Keith Adams and Ole Agesen. A Comparison of Software and Hardware Techniques for x86 Virtualization. *ACM SIGARCH Computer Architecture News*, 34(5):2–13, 2006.

[78] Ramil Agliamzanov, Muhammed Sit, and Ibrahim Demir. Hydrology@ Home: A Distributed Volunteer Computing Framework for Hydrological Research and Applications. *Journal of Hydroinformatics*, 22(2):235–248, 2020.

[79] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. Data Center TCP DCTCP. In *Proceedings of the ACM SIGCOMM 2010 Conference*, page 63–74, 2010.

[80] Mohammad Alizadeh, Adel Javanmard, and Balaji Prabhakar. Analysis of DCTCP: Stability, Convergence, and Fairness. *ACM SIGMETRICS Performance Evaluation Review*, 39(1):73–84, 2011.

[81] EC Amazon. Amazon Elastic Compute Cloud (Amazon EC2). *Amazon Elastic Compute Cloud (Amazon EC2)*, 2010.

[82] J Chris Anderson, Jan Lehnardt, and Noah Slater. *CouchDB: The Definitive Guide: Time to Relax*. "O'Reilly Media, Inc.", 2010.

[83] Michael Armbrust, Reynold S Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K Bradley, Xiangrui Meng, Tomer Kaftan, Michael J Franklin, Ali Ghodsi, et al. Spark SQL: Relational Data Processing in Spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, page 1383–1394. ACM, 2015.

[84] Jason Arnold, Boris Glavic, and Ioan Raicu. Hrdbms: Combining the Best of Modern and Traditional Relational Databases. *arXiv preprint arXiv:1901.08666*, 2019.

[85] Francesco Asnicar, Nadir Sella, Luca Masera, Paolo Morettin, Thomas Tolio, Stanislau Semeniuta, Claudio Moser, Enrico Blanzieri, and Valter Cavecchia. TN-Grid and Gene@ Home Project: Volunteer Computing for Bioinformatics. In *BOINC: FAST 2015 International Conference BOINC: FAST 2015Second International Conference BOINC-based High Performance Computing: Fundamental Research and Development*. Russian Academy of Sciences, 2015.

[86] Kyle Banker. *MongoDB in Action*. Manning Publications Co., 2011.

[87] Narsimha Banothu, ShankarNayak Bhukya, and K Venkatesh Sharma. Big-data: Acid Versus Base for Database Transactions. In *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*, page 3704–3709. IEEE, 2016.

[88] Feng Bao and Robert H Deng. A Signcryption Scheme with Signature Directly Verifiable by Public Key. In *International Workshop on Public Key Cryptography*, page 55–59. Springer, 1998.

[89] Michael Barbaro, Tom Zeller, and Saul Hansell. A Face is Exposed for AOL Searcher no. 4417749. *New York Times*, 9(2008):8For, 2006.

[90] Cristian Andrei Baron et al. NoSQL Key-Value DBs Riak and Redis. *Database Systems Journal*, 4:3–10, 2016.

[91] Daniel C Barth-Jones. The 'Re-identification' of Governor William Weld's Medical Information: A Critical Re-examination of Health Data Identification Risks and Privacy Protections, Then and Now. *Then and Now (July 2012)*, 2012.

[92] Frederic Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[93] Narmeen Zakaria Bawany and Jawwad A Shamsi. SEAL: SDN Based Secure and Agile Framework for Protecting Smart City Applications from DDoS Attacks. *Journal of Network and Computer Applications*, 145:102381, 2019.

[94] Narmeen Zakaria Bawany, Jawwad A Shamsi, and Khaled Salah. DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. *Arabian Journal for Science and Engineering*, 42(2):425–441, 2017.

[95] Adam L Beberg, Daniel L Ensign, Guha Jayachandran, Siraj Khaliq, and Vijay S Pande. Folding@ home: Lessons from Eight Years of Volunteer Distributed Computing. In *2009 IEEE International Symposium on Parallel & Distributed Processing*, page 1–8. IEEE, 2009.

[96] Andras Beleczki and Balint Molnar. Modeling Framework for Designing and Analyzing Document-Centric Information Systems based on HyperGraphDB. In *CEUR Workshop Proceedings (ISSN: 1613-0073)*, volume 2046, page 17–22, 2016.

[97] Andrew John Bernoth. Identifying Additional Firewall Rules that may be needed, July 3 2018. US Patent 10,015,140.

[98] Janki Bhimani, Zhengyu Yang, Miriam Leeser, and Ningfang Mi. Accelerating Big Data Applications Using Lightweight Virtualization Framework on Enterprise Cloud. In *High Performance Extreme Computing Conference (HPEC), 2017 IEEE*, page 1–7. IEEE, 2017.

[99] Carsten Binnig, Andrew Crotty, Alex Galakatos, Tim Kraska, and Erfan Zamanian. The End of Slow Networks: It's Time for a Redesign. *Proceedings of the VLDB Endowment*, 9(7):528–539, 2016.

[100] MKABV Bittorf, Taras Bobrovytsky, CCACJ Erickson, Martin Grund Daniel Hecht, MJIJL Kuff, Dileep Kumar Alex Leblang, NLIPH Robinson, David Rorke Silvius Rus, John Russell Dimitris Tsirogiannis Skye Wanderman, and Milne Michael Yoder. Impala: A Modern, Open-Source SQL Engine for Hadoop. In *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research*, 2015.

[101] Linda Camilla Boldt, Vinothan Vinayagamoorthy, Florian Winder, Melanie Schnittger, Mats Ekran, Raghava Rao Mukkamala, Niels Buus Lassen, Benjamin Flesch, Abid Hussain, and Ravi Vatrapu. Forecasting Nike's Sales using Facebook Data. In *2016 IEEE International Conference on Big Data (Big Data)*, page 2447–2456. IEEE, 2016.

[102] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog Computing and its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, page 13–16. ACM, 2012.

[103] Dhruba Borthakur, Jonathan Gray, Joydeep Sen Sarma, Kannan Muthukkaruppan, Nicolas Spiegelberg, Hairong Kuang, Karthik Ranganathan, Dmytro Molkov, Aravind Menon, Samuel Rash, et al. Apache Hadoop goes Realtime at Facebook. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, page 1071–1080. ACM, 2011.

[104] Roland N Boubela, Klaudius Kalcher, Wolfgang Huf, Christian Nasel, and Ewald Moser. Big Data Approaches for the Analysis of Large-Scale fMRI Data using Apache Spark and GPU Processing: A Demonstration on Resting-State fMRI Data from the Human Connectome Project. *Frontiers in Neuroscience*, 9:492, 2016.

[105] David Bradley, Richard Harper, and Steven Hunter. Power-Aware Workload Balancing using Virtual Machines, March 17 2005. US Patent App. 10/663,285.

[106] Rodrigo Braga, Edjard Mota, and Alexandre Passito. Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow. In *IEEE Local Computer Network Conference*, page 408–415. IEEE, 2010.

[107] Eric Brewer. A Certain Freedom: Thoughts on the Cap Theorem. In *Proceedings of the 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, page 335–335. ACM, 2010.

[108] Eric Brewer. Cap Twelve Years Later: How the "Rules" Have Changed. *Computer*, 45(2):23–29, 2012.

[109] Eric Brewer. Pushing the cap: Strategies for Consistency and Availability. *Computer*, 45(2):23–29, 2012.

[110] Eric Brewer. Spanner, Truetime and the CAP Theorem. 2017.

[111] Eric A Brewer. Towards Robust Distributed Systems. In *PODC*, volume 7, 2000.

[112] Martin C Brown. *Getting Started with Couchbase Server: Extreme Scalability at Your Fingertips*. "O'Reilly Media, Inc.", 2012.

[113] Barbara Brynko. NuoDB: Reinventing the Database. *Information Today*, 29(9):9–9, 2012.

[114] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D Ernst. Haloop: Efficient Iterative Data Processing on Large Clusters. *Proceedings of the VLDB Endowment*, 3(1-2):285–296, 2010.

[115] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D Ernst. The Haloop Approach to Large-Scale Iterative Data Analysis. *The VLDB Journal The International Journal on Very Large Data Bases*, 21(2):169–190, 2012.

[116] Brendan Burns, Brian Grant, David Oppenheimer, Eric Brewer, and John Wilkes. Borg, omega, and kubernetes. *Queue*, 14(1):70–93, 2016.

[117] Rajkumar Buyya et al. High Performance Cluster Computing: Architectures and Systems (volume 1). *Prentice Hall, Upper Saddle River, NJ, USA*, 1:999, 1999.

[118] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud Computing and Emerging it Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599–616, 2009.

[119] Christian Cachin et al. Architecture of the Hyperledger Blockchain Fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, volume 310, page 4, 2016.

[120] Stefano Calzavara, Sebastian Roth, Alvise Rabitti, Michael Backes, and Ben Stock. A Tale of Two Headers: A Formal Analysis of Inconsistent Click-jacking Protection on the Web. 2020.

[121] Josiah L Carlson. *Redis in Action*. Manning Shelter Island, 2013.

[122] Rick Cattell. Scalable SQL and NoSQL Data Stores. *Acm Sigmod Record*, 39(4):12–27, 2011.

[123] Ugur Cetintemel, Nesime Tatbul, Kristin Tufte, Hao Wang, Stanley Zdonik, Jiang Du, Tim Kraska, Samuel Madden, David Maier, John Meehan, et al. S-store: A Streaming NewSQL System for Big Velocity Applications. 2014.

[124] Mallikarjun Chadalapaka, Hemal Shah, Uri Elzur, Patricia Thaler, and Michael Ko. A Study of iSCSI extensions for RDMA (iSER). In *Proceedings of the ACM SIGCOMM Workshop on Network-I/O Convergence: Experience, Lessons, Implications*, page 209–219, 2003.

[125] Prabhakar Chaganti and Rich Helms. *Amazon SimpleDB Developer Guide*. Packt Publishing Ltd, 2010.

[126] Swetha Prabha Chaganti. Voldemort NoSQL Database. 2016.

[127] Bill Chambers and Matei Zaharia. *Spark: The Definitive Guide: Big Data Processing Made Simple*. "O'Reilly Media, Inc.", 2018.

[128] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed Storage System for Structured Data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):1–26, 2008.

[129] Rocky KC Chang. Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial. *IEEE Communications Magazine*, 40(10):42–51, 2002.

[130] Jack Chen, Samir Jindel, Robert Walzer, Rajkumar Sen, Nika Jimsheleishvilli, and Michael Andrews. The MemSQL Query Optimizer: A Modern Optimizer for Real-Time Analytics in a Distributed Database. *Proceedings of the VLDB Endowment*, 9(13):1401–1412, 2016.

[131] Kai Chen, Ankit Singla, Atul Singh, Kishore Ramachandran, Lei Xu, Yueping Zhang, Xitao Wen, and Yan Chen. Osa: An optical switching architecture for data center networks with unprecedented flexibility. *IEEE/ACM Transactions on Networking (TON)*, 22(2):498–511, 2014.

[132] Qun Chen, Song Bai, Zhanhuai Li, Zhiying Gou, Bo Suo, and Wei Pan. GraphHP: A Hybrid Platform for Iterative Graph Processing. *arXiv preprint arXiv:1706.07221*, 2017.

[133] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. *arXiv preprint arXiv:1512.01274*, 2015.

[134] Wen Chen, Peng Cheng, Fengyuan Ren, Ran Shu, and Chuang Lin. Ease the Queue Oscillation: Analysis and Enhancement of DCTCP. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, page 450–459. IEEE, 2013.

[135] Xue-Wen Chen and Xiaotong Lin. Big Data Deep Learning: Challenges and Perspectives. *IEEE access*, 2:514–525, 2014.

[136] Yanpei Chen, Rean Griffit, David Zats, and Randy H Katz. Understanding TCP incast and its implications for Big Data Workloads. *University of California at Berkeley, Tech. Rep*, 2012.

[137] Yi Chen, Zhi Qiao, Hai Jiang, Kuan-Ching Li, and Won Woo Ro. MGMR: Multi-GPU based MapReduce. In *International Conference on Grid and Pervasive Computing*, page 433–442. Springer, 2013.

[138] Avery Ching, Sergey Edunov, Maja Kabiljo, Dionysios Logothetis, and Sambavi Muthukrishnan. One trillion edges: Graph Processing at Facebook-Scale. *Proceedings of the VLDB Endowment*, 8(12):1804–1815, 2015.

[139] Kristina Chodorow. *Scaling MongoDB: Sharding, Cluster Setup, and Administration*. "O'Reilly Media, Inc.", 2011.

[140] Kristina Chodorow. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. "O'Reilly Media, Inc.", 2013.

[141] Mrs Rupali M Chopade and Nikhil S Dhavase. MongoDB, Couchbase: Performance Comparison for Image Dataset. In *2017 2nd International Conference for Convergence in Technology (I2CT)*, page 255–258. IEEE, 2017.

[142] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated Feedback Recurrent Neural Networks. In *International Conference on Machine Learning*, page 2067–2075, 2015.

[143] Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. An End-to-End View of DNSSEC Ecosystem Management. *; login:*, 42(4), 2017.

[144] Dan Ciresan, Ueli Meier, Jonathan Masci, and Jurgen Schmidhuber. Multi-Column Deep Neural Network for Traffic Sign Classification. *Neural Networks*, 32:333–338, 2012.

[145] Cisco. Data Center: Load Balancing Data Center Services.

[146] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live Migration of Virtual Machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, page 273–286. USENIX Association, 2005.

[147] Michael Colesky, Jaap-Henk Hoepman, and Christiaan Hillen. A Critical Analysis of Privacy Design Strategies. In *Security and Privacy Workshops (SPW), 2016 IEEE*, page 33–40. IEEE, 2016.

[148] James C Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, Jeffrey John Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, et al. Spanner: Googles Globally Distributed Database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):1–22, 2013.

[149] Antonio Corradi, Mario Fanelli, and Luca Foschini. Vm consolidation: A Real Case Based on Openstack Cloud. *Future Generation Computer Systems*, 32:118–127, 2014.

[150] Laizhong Cui, F Richard Yu, and Qiao Yan. When Big Data Meets Software-Defined Networking: SDN for Big Data and Big Data for SDN. *IEEE Network*, 30(1):58–65, 2016.

[151] Ian Curry. An Introduction to Cryptography and Digital Signatures. *Entrust Securing Digital Identities and Information*, 2001.

[152] Nhu-Ngoc Dao, Junho Park, Minho Park, and Sungrae Cho. A Feasible Method to Combat Against DDoS Attack in SDN Network. In *2015 International Conference on Information Networking (ICOIN)*, page 309–311. IEEE, 2015.

[153] Marieke De Goede. The politics of Privacy in the Age of Preemptive Security. *International Political Sociology*, 8(1):100–104, 2014.

[154] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[155] Jeffrey Dean and Sanjay Ghemawat. MapReduce: A Flexible Data Processing Tool. *Communications of the ACM*, 53(1):72–77, 2010.

[156] Casimer DeCusatis. Optical Interconnect Networks for Data Communications. *Journal of Lightwave Technology*, 32(4):544–552, 2014.

[157] Kevin Deierling. Ethernet Just Got a Big Performance Boost with Release of Soft RoCE, 2015.

[158] OrientDB Developers. OrientDB. *Hybrid Document-Store and Graph NoSQL Database [online]*, 2012.

[159] David HC Du, Tai-Sheng Chang, Jenwei Hsieh, Sangyup Shim, and Yuewei Wang. Two Emerging Serial Storage Interfaces for Supporting Digital Libraries: Serial Storage Architecture (SSA) and Fiber Channel-Arbitrated Loop (FC-AL). *Multimedia Tools and Applications*, 10(2):179–203, 2000.

[160] Muhammad Nouman Durrani and Jawwad A Shamsi. Volunteer Computing: Requirements, Challenges, and Solutions. *Journal of Network and Computer Applications*, 39:369–380, 2014.

[161] Michael Erbschloe. *Trojans, Worms, and Spyware: A Computer Security Professional's Guide to Malicious Code*. Elsevier, 2004.

[162] Hamza Es-Samaali, Aissam Outchakoucht, and Jean Philippe Leroy. A Blockchain-Based Access Control for Big Data. *International Journal of Computer Networks and Communications Security*, 5(7):137, 2017.

[163] Christian Esposito, Aniello Castiglione, and Kim-Kwang Raymond Choo. Challenges in Delivering Software in the Cloud as Microservices. *IEEE Cloud Computing*, (5):10–14, 2016.

[164] Reza Farivar, Daniel Rebolledo, Ellick Chan, and Roy H Campbell. A Parallel Implementation of K-means Clustering on GPUs. In *PDPTA*, volume 13, page 212–312, 2008.

[165] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers. *ACM SIGCOMM Computer Communication Review*, 40(4):339–350, 2010.

[166] Maria Fazio, Antonio Celesti, Rajiv Ranjan, Chang Liu, Lydia Chen, and Massimo Villari. Open Issues in Scheduling Microservices in the Cloud. *IEEE Cloud Computing*, 3(5):81–88, 2016.

[167] M Fenn, MA Murphy, J Martin, and S Goasguen. An Evaluation of KVM for use in Cloud Computing. In *Proceedings of the 2nd International Conference on the Virtual Computing Initiative, RTP, NC, USA*, 2008.

[168] Michael J Flynn. Very High-Speed Computing Systems. *Proceedings of the IEEE*, 54(12):1901–1909, 1966.

[169] Julien Forgeat. Data Processing Architectures-Lambda and Kappa. *Ericsson Research Blog*, 2015.

[170] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud Computing and Grid Computing 360-Degree Compared. In *Grid Computing Environments Workshop, 2008. GCE'08*, page 1–10. IEEE, 2008.

[171] Michael Frampton. *Big Data Made Easy: A Working Guide to the Complete Hadoop Toolset*. Apress, 2014.

[172] Vincent Garcia, Eric Debreuve, and Michel Barlaud. Fast K Nearest Neighbor Search using GPU. *arXiv preprint arXiv:0804.1448*, 2008.

[173] Alan Gates and Daniel Dai. *Programming Pig: Dataflow Scripting with Hadoop*. "O'Reilly Media, Inc.", 2016.

[174] Alan Gates, Jianyong Dai, and Thejas Nair. Apache Pig's Optimizer. *IEEE Data Engineering Bulletin*, 36(1):34–45, 2013.

[175] Alan F Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan M Narayanamurthy, Christopher Olston, Benjamin Reed, Santhosh Srinivasan, and Utkarsh Srivastava. Building A High-Level Dataflow System on Top of Map-Reduce: The Pig Experience. *Proceedings of the VLDB Endowment*, 2(2):1414–1425, 2009.

[176] Lars George. *HBase: The Definitive Guide: Random Access to your Planet-Size Data*. "O'Reilly Media, Inc.", 2011.

[177] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System. In *ACM SIGOPS Operating Systems Review*, volume 37, page 29–43. ACM, 2003.

[178] Seth Gilbert and Nancy A Lynch. Perspectives on the Cap Theorem. *Computer*, 45(2):30–36, 2012.

[179] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[180] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27, 2017.

[181] Alex Graves, Santiago Fernandez, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd International Conference on Machine learning*, page 369–376. ACM, 2006.

[182] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech Recognition with Deep Recurrent Neural Networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, page 6645–6649. IEEE, 2013.

[183] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: A Scalable and Flexible Data Center Network. In *ACM SIGCOMM Computer Communication Review*, volume 39, page 51–62. ACM, 2009.

[184] Steven L Grobman. Server Pool Kerberos Authentication Scheme, March 21 2017. US Patent 9,602,275.

[185] Katarina Grolinger, Wilson A Higashino, Abhinav Tiwari, and Miriam AM Capretz. Data Management in Cloud Environments: NoSQL and NewSQL Data Stores. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1):22, 2013.

[186] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. Bcube: A High Performance, Server-Centric Network Architecture for Modular Data Centers. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, page 63–74, 2009.

[187] Himanshu Gupta, Subhash Mondal, Srayan Ray, Biswajit Giri, Rana Majumdar, and Ved P Mishra. Impact of SQL Injection in Database Security. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, page 296–299. IEEE, 2019.

[188] Irfan Habib. Virtualization with KVM. *Linux Journal*, 2008(166):8, 2008.

[189] William G Halfond, Jeremy Viegas, Alessandro Orso, et al. A Classification of SQL-Injection Attacks and Countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering*, volume 1, page 13–15. IEEE, 2006.

[190] Jing Han, E Haihong, Guan Le, and Jian Du. Survey on NoSQL Database. In *2011 6th International Conference on Pervasive Computing and Applications*, page 363–366. IEEE, 2011.

[191] T Harford. Big Data: Are We Making a Big Mistake? [internet]. London: Ft magazine; c2014 [cited at 2015 sep 28].

[192] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The Rise of "Big Data on Cloud Computing: Review and Open Research Issues. *Information Systems*, 47:98–115, 2015.

[193] Michael Hausenblas and Jacques Nadeau. Apache Drill: Interactive Ad-hoc Analysis at Scale. *Big Data*, 1(2):100–104, 2013.

[194] Bingsheng He, Wenbin Fang, Qiong Luo, Naga K Govindaraju, and Tuyong Wang. Mars: A MapReduce Framework on Graphics Processors. In *Parallel Architectures and Compilation Techniques (PACT), 2008 International Conference on*, page 260–269. IEEE, 2008.

[195] Ying He, F Richard Yu, Nan Zhao, Victor CM Leung, and Hongxi Yin. Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach. *IEEE Communications Magazine*, 55(12):31–37, 2017.

[196] Thomas A Hengeveld. Multi-Tunnel Virtual Private Network, March 29 2016. US Patent 9,300,570.

[197] Maurice Herlihy. Blockchains From a Distributed Computing Perspective. *Communications of the ACM*, 62(2):78–85, 2019.

[198] Bai Hong-Tao, He Li-li, Ouyang Dan-tong, Li Zhan-shan, and Li He. K-Means on Commodity GPUs with CUDA. In *2009 World Congress on Computer Science and Information Engineering*, page 651–655. IEEE, 2009.

[199] Weisheng Hu, Weiqiang Sun, Yaohui Jin, Wei Guo, and Shilin Xiao. An Efficient Transportation Architecture for Big Data Movement. In *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, page 1–5. IEEE, 2013.

[200] Yin Huai, Ashutosh Chauhan, Alan Gates, Gunther Hagleitner, Eric N Hanson, Owen O'Malley, Jitendra Pandey, Yuan Yuan, Rubao Lee, and Xiaodong Zhang. Major Technical Advancements in Apache Hive. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, page 1235–1246. ACM, 2014.

[201] Patrick Hunt, Mahadev Konar, Flavio Paiva Junqueira, and Benjamin Reed. Zookeeper: Wait-Free Coordination for Internet-Scale Systems. In *USENIX Annual Technical Conference*, volume 8. Boston, MA, USA, 2010.

[202] Jalal B Hur and Jawwad A Shamsi. A Survey on Security Issues, Vulnerabilities and Attacks in Android Based Smartphone. In *2017 International Conference on Information and Communication Technologies (ICICT)*, page 40–46. IEEE, 2017.

[203] Intel. Understanding iWARP: Delivering Low Latency to Ethernet.

[204] Borislav Iordanov. HyperGraphDB: A Generalized Graph Database. In *International Conference on Web-Age Information Management*, page 25–36. Springer, 2010.

[205] Waheed Iqbal. Service Level Agreement Driven Adaptive Resource Management for Web Applications on Heterogeneous Compute Clouds. Master's thesis, 2009.

[206] Tania Iram, Jawwad Shamsi, Usama Alvi, Saif ur Rahman, and Muhammad Maaz. Controlling Smart-city Traffic using Machine Learning. In *2019 International Conference on Frontiers of Information Technology (FIT)*, page 203–2035. IEEE, 2019.

[207] Sasha Issenberg. How President Obamas Campaign used Big Data to Rally Individual Voters, 2012.

[208] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big Data and its Technical Challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[209] Raj Jain and Subharthi Paul. Network Virtualization and Software Defined Networking for Cloud Computing: A Survey. *IEEE Communications Magazine*, 51(11):24–31, 2013.

[210] Meiko Jensen. Challenges of Privacy Protection in Big Data Analytics. In *Big Data (BigData Congress), 2013 IEEE International Congress on*, page 235–238. IEEE, 2013.

[211] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter Performance Analysis of a Tensor Processing Unit. *arXiv preprint arXiv:1704.04760*, 2017.

[212] Flavio P Junqueira and Benjamin C Reed. The Life and Times of a Zookeeper. In *Proceedings of the Twenty-First Annual Symposium on Parallelism in Algorithms and Architectures*, page 46–46. ACM, 2009.

[213] Dharmesh Kakadia. *Apache Mesos Essentials*. Packt Publishing Ltd, 2015.

[214] Robert Kallman, Hideaki Kimura, Jonathan Natkins, Andrew Pavlo, Alexander Rasin, Stanley Zdonik, Evan PC Jones, Samuel Madden, Michael Stonebraker, Yang Zhang, et al. H-store: A High-Performance, Distributed Main Memory Transaction Processing System. *Proceedings of the VLDB Endowment*, 1(2):1496–1499, 2008.

[215] Seny Kamara and Kristin Lauter. Cryptographic Cloud Storage. In *International Conference on Financial Cryptography and Data Security*, page 136–149. Springer, 2010.

[216] Karthik Kambatla, Giorgos Kollias, Vipin Kumar, and Ananth Grama. Trends in Big Data Analytics. *Journal of Parallel and Distributed Computing*, 74(7):2561–2573, 2014.

[217] Debabrata Kar, Suvasini Panigrahi, and Srikanth Sundararajan. SQLiGoT: Detecting SQL Injection Attacks Using Graph of Tokens and SVM. *Computers & Security*, 60:206–225, 2016.

[218] Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia. *Learning Spark: Lightning-Fast Big Data Analysis*. "O'Reilly Media, Inc.", 2015.

[219] Wayne Karpoff and Brian Lake. Storage Virtualization System and Methods, August 18 2009. US Patent 7,577,817.

[220] Karambir Kaur and Monika Sachdeva. Performance Evaluation of NewSQL Databases. In *2017 International Conference on Inventive Systems and Control (ICISC)*, page 1–5. IEEE, 2017.

[221] Sawinder Kaur and Karamjit Guide Kaur. *Visualizing Class Diagram using OrientDB NoSQL Data-Store*. PhD thesis, 2016.

[222] Robert W Kembel and Horst L Truestedt. Fibre Channel Arbitrated Loop. 1996.

[223] David B Kirk and W Hwu Wen-Mei. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2016.

[224] O Kolkman and R Gieben. DNSSEC Operational Practices. Technical report, RFC 4641, September, 2006.

[225] Richard T Kouzes, Gordon A Anderson, Stephen T Elbert, Ian Gorton, and Deborah K Gracio. The Changing Paradigm of Data-Intensive Computing. *Computer*, (1):26–34, 2009.

[226] Jay Kreps. Parallel Hardware Architecture. *Oracle, Dec.*

[227] Jay Kreps. The Log: What Every Software Engineer Should Know About Real-Time Datas Unifying Abstraction. *Linkedin. com, Dec*, 16, 2013.

[228] Jay Kreps. Questioning the Lambda Architecture. *Online Article, July*, page 205, 2014.

[229] Jay Kreps, Neha Narkhede, Jun Rao, et al. Kafka: A Distributed Messaging System for Log Processing. In *Proceedings of the NetDB*, page 1–7, 2011.

[230] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[231] Alexandros Labrinidis and Hosagrahar V Jagadish. Challenges and Opportunities with Big Data. *Proceedings of the VLDB Endowment*, 5(12):2032–2033, 2012.

[232] Kashif Laeeq and Jawwad A Shamsi. A Study of Security Issues, Vulnerabilities and Challenges in Internet of Things. *Securing Cyber-Physical Systems*, 10, 2015.

[233] Daniel Guimaraes do Lago, Edmundo RM Madeira, and Luiz Fernando Bittencourt. Power-Aware Virtual Machine Scheduling on Clouds using Active Cooling Control and DVFS. In *Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science*, page 2. ACM, 2011.

[234] Avinash Lakshman and Prashant Malik. Cassandra: A Decentralized Structured Storage System. *ACM SIGOPS Operating Systems Review*, 44(2):35–40, 2010.

[235] Phillip A Laplante. Who's Afraid of Big Data? *IT Professional*, 15(5):6–7, 2013.

[236] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The Parable of Google Flu: Traps in Big Data Analysis. *Science*, 343(14 March), 2014.

[237] Brian Lebiednik, Aman Mangal, and Niharika Tiwari. A Survey and Evaluation of Data Center Network Topologies. *arXiv preprint arXiv:1605.01701*, 2016.

[238] Jae-Gil Lee and Minseo Kang. Geospatial Big Data: Challenges and Opportunities. *Big Data Research*, 2(2):74–81, 2015.

[239] Ken Ka-Yin Lee, Wai-Choi Tang, and Kup-Sze Choi. Alternatives to Relational Database: Comparison of NoSQL and XML Approaches for Clinical Data Storage. *Computer Methods and Programs in Biomedicine*, 110(1):99–109, 2013.

[240] Sangdo Lee and Jun-Ho Huh. An Effective Security Measures for Nuclear Power Plant Using Big Data Analysis Approach. *The Journal of Supercomputing*, 75(8):4267–4294, 2019.

[241] Joe Lennon. *Beginning CouchDB*. Apress, 2010.

[242] Hu Li, Tianjia Chen, and Wei Xu. Improving Spark Performance with Zero-Copy Buffer Management and RDMA. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, page 33–38. IEEE, 2016.

[243] Peilong Li, Yan Luo, Ning Zhang, and Yu Cao. Heterospark: A Heterogeneous CPU/GPU Spark Platform for Machine Learning Algorithms. In *Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on*, page 347–348. IEEE, 2015.

[244] Qi Li, Jianfeng Ma, Rui Li, Ximeng Liu, Jinbo Xiong, and Danwei Chen. Secure, Efficient and Revocable Multi-Authority Access Control System in Cloud Storage. *Computers & Security*, 59:45–59, 2016.

[245] Jimmy Lin and Chris Dyer. *Data Intensive Text processing with MapReduce*. Morgan Claypool Publishers, 2010.

[246] Xuan-Yi Lin, Yeh-Ching Chung, and Tai-Yi Huang. A Multiple LID Routing Scheme for Fat-Tree-Based infiniband Networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 11. IEEE, 2004.

[247] Maria Lindh and Jan Nolin. Information We Collect: Surveillance and Privacy in the Implementation of Google Apps for Education. *European Educational Research Journal*, 15(6):644–663, 2016.

[248] Alex X Liu and Mohamed G Gouda. Diverse Firewall Design. *IEEE Transactions on Parallel and Distributed Systems*, 19(9):1237–1251, 2008.

[249] Stephanie Q Liu and Anna S Mattila. Airbnb: Online Targeted Advertising, Sense of Power, and Consumer Decisions. *International Journal of Hospitality Management*, 60:33–41, 2017.

[250] Yimeng Liu, Yizhi Wang, and Yi Jin. Research on the Improvement of MongoDB Auto-Sharding in Cloud Environment. In *Computer Science & Education (ICCSE), 2012 7th International Conference on*, page 851–854. IEEE, 2012.

[251] Win-Tsung Lo, Yue-Shan Chang, Ruey-Kai Sheu, Chun-Chieh Chiu, and Shyan-Ming Yuan. Cudt: A CUDA Based Decision Tree Algorithm. *The Scientific World Journal*, 2014, 2014.

[252] Noel Lopes and Bernardete Ribeiro. Gpumlib: An Efficient Open-Source GPU Machine Learning Library. *International Journal of Computer Information Systems and Industrial Management Applications*, 3:355–362, 2011.

[253] Adam Lopez. Statistical Machine Translation. *ACM Computing Surveys (CSUR)*, 40(3):8, 2008.

[254] Yingping Lu and David HC Du. Performance Study of iSCSI-Based Storage Subsystems. *IEEE Communications Magazine*, 41(8):76–82, 2003.

[255] Marko Luksa. *Kubernetes in action*. Manning Publications Shelter Island, 2018.

[256] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[257] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy Beyond K-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[258] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: A System for Large-Scale Graph Processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, page 135–146, 2010.

[259] Claudio Martella, Roman Shaposhnik, Dionysios Logothetis, and Steve Harenberg. *Practical Graph Analytics with Apache Giraph*, volume 1. Springer, 2015.

[260] Nathan Marz. How to Beat the Cap Theorem. *Thoughts from the Red Planet*, 2011.

[261] Matthew L Massie, Brent N Chun, and David E Culler. The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. *Parallel Computing*, 30(7):817–840, 2004.

[262] Yuan Mei, Luwei Cheng, Vanish Talwar, Michael Y Levin, Gabriela Jacques-Silva, Nikhil Simha, Anirban Banerjee, Brian Smith, Tim Williamson, Serhat Yilmaz, et al. Turbine: Facebooks Service Management Platform for Stream Processing. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, page 1591–1602. IEEE, 2020.

[263] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing (draft). *NIST Special Publication*, 800(145):7, 2011.

[264] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. Dremel: Interactive Analysis of Web-Scale Datasets. *Proceedings of the VLDB Endowment*, 3(1–2):330–339, 2010.

[265] Laraib U Memon, Narmeen Z Bawany, and Jawwad A Shamsi. A Comparison of Machine Learning Techniques for Android Malware Detection using Apache Spark. *Journal of Engineering Science and Technology*, 14(3):1572–1586, 2019.

[266] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine Learning in Apache Spark. *The Journal of Machine Learning Research*, 17(1):1235–1241, 2016.

[267] Dirk Merkel. Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux Journal*, 2014(239):2, 2014.

[268] Ahmed Metwally and Christos Faloutsos. V-smart-join: A Scalable MapReduce Framework for All-Pair Similarity Joins of Multisets and Vectors. *Proceedings of the VLDB Endowment*, 5(8):704–715, 2012.

[269] Shivlal Mewada, Pradeep Sharma, and SS Gautam. Classification of Efficient Symmetric Key Cryptography Algorithms. *International Journal of Computer Science and Information Security*, 14(2):105, 2016.

[270] Yajie Miao, Hao Zhang, and Florian Metze. Distributed Learning of Multilingual DNN Feature Extractors using GPUs. 2014.

[271] Microsft. Data Center Bridging (DCB) Overview.

[272] Justin J Miller. Graph Database Applications and Concepts with Neo4j. In *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*, volume 2324, 2013.

[273] Christopher Mitchell, Yifeng Geng, and Jinyang Li. Using One-Sided RDMA Reads to Build a Fast, CPU-Efficient Key-Value Store. In *USENIX Annual Technical Conference*, page 103–114, 2013.

[274] Radhika Mittal, Alex Shpiner, Aurojit Panda, Eitan Zahavi, Arvind Krishnamurthy, Sylvia Ratnasamy, and Scott Shenker. Revisiting Network Support for RDMA.

[275] Inder Monga, Eric Pouyoul, and Chin Guok. Software-Defined Networking for Big-Data Science-Architectural Models from Campus to the Wan. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, page 1629–1635. IEEE, 2012.

[276] Roberto Morabito, Jimmy Kjällman, and Miika Komu. Hypervisors vs. Lightweight Virtualization: A Performance Comparison. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, page 386–393. IEEE, 2015.

[277] Ruchika Muddinagiri, Shubham Ambavane, and Simran Bayas. Self-hosted kubernetes: Deploying docker containers locally with minikube. In *2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET)*, pages 239–243. IEEE, 2019.

[278] Nitin Naik. Building a virtual system of systems using docker swarm in multiple clouds. In *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–3. IEEE, 2016.

[279] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep Learning Applications and Challenges in Big Data Analytics. *Journal of Big Data*, 2(1):1, 2015.

[280] Arvind Narayanan and Vitaly Shmatikov. Robust De-Anonymization of Large Sparse Datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, page 111–125. IEEE, 2008.

[281] Arvind Narayanan and Vitaly Shmatikov. Myths and Fallacies of Personally Identifiable Information. *Communications of the ACM*, 53(6):24–26, 2010.

[282] Rimma Nehme and Nicolas Bruno. Automated Partitioning Design in Parallel Database Systems. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, page 1137–1148. ACM, 2011.

[283] B Clifford Neuman and Theodore Ts'o. Kerberos: An Authentication Service for Computer Networks. *IEEE Communications Magazine*, 32(9):33–38, 1994.

[284] Krishna Nibhanupudi and Rimmi Devgan. Data Center Ethernet.

[285] Muhammad Nouman Durrani and Jawwad A Shamsi. Volunteer Computing: Requirements, Challenges, and Solutions. *Journal of Network and Computer Applications*, 2013.

[286] Daniel Nurmi, Richard Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The Eucalyptus Open-Source Cloud-Computing System. In *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on*, page 124–131. IEEE, 2009.

[287] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig Latin: A Not-so-Foreign Language for Data Processing. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, page 1099–1110. ACM, 2008.

[288] Oyindamola Oluwatimi, Daniele Midi, and Elisa Bertino. Overview of Mobile Containerization Approaches and Open Research Directions. *IEEE Security & Privacy*, 15(1):22–31, 2017.

[289] Claus Pahl. Containerization and the Paas Cloud. *IEEE Cloud Computing*, 2(3):24–31, 2015.

[290] Claus Pahl, Antonio Brogi, Jacopo Soldani, and Pooyan Jamshidi. Cloud Container Technologies: A State-of-the-Art Review. *IEEE Transactions on Cloud Computing*, 2017.

[291] Rakesh Patel, Mara Nicholl, and Lindsey Harju. Access Control System for Implementing Access Restrictions of Regulated Database Records while Identifying and Providing Indicators of Regulated Database Records Matching Validation Criteria, September 19 2017. US Patent 9,767,309.

[292] Andrew Pavlo and Matthew Aslett. What's Really New with newSQL? *ACM Sigmod Record*, 45(2):45–55, 2016.

[293] Gregory F Pfister. An Introduction to the Infiniband Architecture. *High Performance Mass Storage and Parallel I/O*, 42:617–632, 2001.

[294] Jaroslav Pokorny. NoSQL Databases: A Step to Database Scalability in Web Environment. *International Journal of Web Information Systems*, 9(1):69–82, 2013.

[295] Lesandro Ponciano, Francisco Brasileiro, Robert Simpson, and Arfon Smith. Volunteers' Engagement in Human Computation for Astronomy Projects. *Computing in Science & Engineering*, 16(6):52–59, 2014.

[296] Andrea Possemato, Andrea Lanzi, Simon Pak Ho Chung, Wenke Lee, and Yanick Fratantonio. Clickshield: Are You Hiding Something? Towards Eradicating Clickjacking on Android. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, page 1120–1136, 2018.

[297] Steve Pousty and Katie Miller. *Getting Started with OpenShift: A Guide for Impatient Beginners.* " O'Reilly Media, Inc.", 2014.

[298] Gil Press. A Very Short History of Big Data. *FORBES. Recuperado May*, 12:2014, 2013.

[299] Dan Pritchett. Base: An Acid Alternative. *Queue*, 6(3):48–55, 2008.

[300] Peng Qin, Bin Dai, Benxiong Huang, and Guan Xu. Bandwidth-Aware Scheduling with SDN in Hadoop: A New Trend for Big Data. *IEEE Systems Journal*, 2015.

[301] US Rackspace. Inc.,The Rackspace Cloud, 2010.

[302] Peter Rausch, Alaa F Sheta, and Aladdin Ayesh. *Business Intelligence and Performance Management: Theory, Systems and Industrial Applications.* Springer Publishing Company, Incorporated, 2013.

[303] Tejaswi Redkar and Tony Guidici. *Windows Azure Platform.* Apress, 2011.

[304] Antony Rowstron, Dushyanth Narayanan, Austin Donnelly, Greg O'Shea, and Andrew Douglas. Nobody Ever Got Fired for Using Hadoop on a cluster. In *Proceedings of the 1st International Workshop on Hot Topics in Cloud Data Processing*, page 2. ACM, 2012.

[305] Sherif Sakr, Faisal Moeen Orakzai, Ibrahim Abdelaziz, and Zuhair Khayyat. *Large-scale Graph Processing using Apache Giraph.* Springer, 2016.

[306] Semih Salihoglu, Jaeho Shin, Vikesh Khanna, Ba Quan Truong, and Jennifer Widom. Graft: A Debugging Tool for Apache Giraph. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, page 1403–1408. ACM, 2015.

[307] Juha Salo. Data Center Network Architectures.

[308] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming.* Addison-Wesley Professional, 2010.

[309] Kaz Sato, Cliff Young, and David Patterson. An in-Depth Look at Googles First Tensor Processing Unit (TPU). *Google Cloud Big Data and Machine Learning Blog*, 12, 2017.

[310] Julian Satran and Kalman Meth. Internet Small Computer Systems Interface (iSCSI). 2004.

[311] Gigi Sayfan. *Mastering kubernetes.* Packt Publishing Ltd, 2017.

[312] Mathijs Jeroen Scheepers. Virtualization and Containerization of Application Infrastructure: A Comparison. In *21st Twente Student Conference on IT*, volume 1, page 1–7, 2014.

[313] Jürgen Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, 2015.

[314] Friedhelm Schmidt. The SCSI Bus and IDE Interface Protocols. *Application and Programming, Addison-Wesley, New York*, 1995.

[315] Nicolas Seyvet and Ignacio Mulas Viela. Applying the Kappa Architecture in the Telco Industry. https://www.oreilly.com/ideas/applying-the-kappa-architecture-in-the-telco-industry (visited: 2019-11-09).

[316] Jawad Ali Shah, Hassaan Haider, Kushsairy Abdul Kadir, and Sheroz Khan. Sparse Signal Reconstruction of Compressively Sampled Signals Using Smoothed 0-Norm. In *Signal and Image Processing Applications (ICSIPA), 2017 IEEE International Conference on*, page 61–65. IEEE, 2017.

[317] Jawwad Shamsi, Muhammad Ali Khojaye, and Mohammad Ali Qasmi. Data-intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions. *Journal of Grid Computing*, 11(2):281–310, 2013.

[318] Jawwad A Shamsi, Sufian Hameed, Waleed Rahman, Farooq Zuberi, Kaiser Altaf, and Ammar Amjad. Clicksafe: Providing Security Against Clickjacking Attacks. In *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, page 206–210. IEEE, 2014.

[319] Jawwad A Shamsi and Muhammad Khojaye. Understanding Privacy Violations in Big Data Systems. *IT Professional*.

[320] Jawwad A Shamsi, Sherali Zeadally, and Zafar Nasir. Interventions in Cyberspace: Status and Trends. *IT Professional*, 18(1):18–25, 2016.

[321] Jawwad A Shamsi, Sherali Zeadally, Fareha Sheikh, and Angelyn Flowers. Attribution in Cyberspace: Techniques and Legal Implications. *Security and Communication Networks*, 9(15):2886–2900, 2016.

[322] Toby Sharp. Implementing Decision Trees and Forests on a GPU. In *European Conference on Computer Vision*, page 595–608. Springer, 2008.

[323] Alexander Shpiner, Eitan Zahavi, Omar Dahley, Aviv Barnea, Rotem Damsker, Gennady Yekelis, Michael Zus, Eitan Kuta, and Dean Baram. Roce Rocks without PFC: Detailed Evaluation. In *Proceedings of the Workshop on Kernel-Bypass Networks*, page 25–30. ACM, 2017.

[324] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The Hadoop Distributed File System. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, page 1–10. IEEE, 2010.

[325] Konstantin V Shvachko. HDFS Scalability: The Limits to Growth. *; login: The Magazine of USENIX & SAGE*, 35(2):6–16, 2010.

[326] Kamran Siddique, Zahid Akhtar, Edward J Yoon, Young-Sik Jeong, Dipankar Dasgupta, and Yangwoo Kim. Apache Hama: An Emerging Bulk Synchronous Parallel Computing Framework for Big Data Applications. *IEEE Access*, 4:8879–8887, 2016.

[327] Tooba Siddiqui and Jawwad Ahmed Shamsi. Generating Abstractive Summaries Using Sequence to Sequence Attention Model. In *2018 International Conference on Frontiers of Information Technology (FIT)*, page 212–217. IEEE, 2018.

[328] Nanki Sidhu, Edzer Pebesma, and Gilberto Câmara. Using google earth engine to detect land cover change: Singapore as a use case. *European Journal of Remote Sensing*, 51(1):486–500, 2018.

[329] Dilpreet Singgh and Chandan K Reddy. A Survey on Platforms for Big Data Analytics. *Journal of Big Data*, 1(8):1–20, 2014.

[330] Aameek Singh, Madhukar Korupolu, and Dushmanta Mohapatra. Server-Storage Virtualization: Integration and Load Balancing in Data Centers. In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, page 53. IEEE Press, 2008.

[331] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical Analysis of Big Data Challenges and Analytical Methods. *Journal of Business Research*, 70:263–286, 2017.

[332] Swaminathan Sivasubramanian. Amazon DynamoDB: A Seamlessly Scalable Non-Relational Database Service. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, page 729–730. ACM, 2012.

[333] Joseph D Sloan. *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI*. O'reilly, 2009.

[334] Aleksander Slominski, Vinod Muthusamy, and Rania Khalaf. Building a Multi-tenant Cloud Service from Legacy Code with Docker Containers. In *2015 IEEE International Conference on Cloud Engineering (IC2E)*, page 394–396. IEEE, 2015.

[335] Daniel J Solove and Danielle Keats Citron. Risk and Anxiety: A Theory of Data-Breach Harms. *Texas Law Review*, 96:737, 2017.

[336] Stephen Soltesz, Herbert Ptzl, Marc E Fiuczynski, Andy Bavier, and Larry Peterson. Container-based Operating System Virtualization: A Scalable, High-Performance Alternative to Hypervisors. In *ACM SIGOPS Operating Systems Review*, volume 41, page 275–287. ACM, 2007.

[337] Jordi Soria-Comas and Josep Domingo-Ferrert. Differential Privacy Via T-closeness in Data Publishing. In *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, page 27–35. IEEE, 2013.

[338] Kristopher A Standish, Sam Amiri, Misbah Mubarak, Louisa J Bellis, Takanori Fujiwara, and John L Rayner. Advances in Supercomputing. *Advances in Supercomputing*, page 157, 2020.

[339] Nick Steele, Stan Hawkins, Joe Maranville, and Andrew Bradnan. Single Sign-on for Access to a Central Data Repository, March 27 2018. US Patent 9,928,508.

[340] Stergios Stergiou. Scaling Pagerank to 100 Billion Pages. In *Proceedings of The Web Conference 2020*, page 2761–2767, 2020.

[341] Thomas Lawrence Sterling. *Beowulf Cluster Computing with Linux*. MIT press, 2002.

[342] Michael Stonebraker, Daniel Abadi, David J DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, and Alexander Rasin. MapReduce and Parallel DBMSs: Friends or Foes? *Communications of the ACM*, 53(1):64–71, 2010.

[343] Michael Stonebraker and Ariel Weisberg. The VoltDB Main Memory DBMS. *IEEE Data Engineering Bulletin*, 36(2):21–27, 2013.

[344] Nikko Strom. Scalable Distributed DNN Training Using Commodity GPU Cloud Computing. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[345] Jeff A Stuart, Cheng-Kai Chen, Kwan-Liu Ma, and John D Owens. Multi-GPU Volume Rendering using MapReduce. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, page 841–848. ACM, 2010.

[346] Jeff A Stuart and John D Owens. Multi-GPU MapReduce on GPU Clusters. In *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*, page 1068–1079. IEEE, 2011.

[347] Michelle Suh, Sae Hyong Park, Byungjoon Lee, and Sunhee Yang. Building Firewall over the Software-Defined Network Controller. In *16th International Conference on Advanced Communication Technology*, page 744–748. IEEE, 2014.

[348] Alexey Svyatkovskiy, Kosuke Imai, Mary Kroeger, and Yuki Shiraito. Large-scale text processing pipeline with apache spark. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3928–3935. IEEE, 2016.

[349] Colin Tankard. Big Data Security. *Network Security*, 2012(7):5–8, 2012.

[350] Linnet Taylor and Ralph Schroeder. Is Bigger Better? the Emergence of Big Data as a Tool for International Development Policy. *GeoJournal*, 80(4):503–518, 2015.

[351] Claudio Tesoriero. *Getting Started with OrientDB*. Packt Publishing Ltd, 2013.

[352] D.J. Patil Thomas H. Davenport. Data Scientist: The Sexiest Job of the 21st Century.

[353] Ashish Thusoo, Zheng Shao, Suresh Anthony, Dhruba Borthakur, Namit Jain, Joydeep Sen Sarma, Raghotham Murthy, and Hao Liu. Data Warehousing and Analytics Infrastructure at Facebook. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, page 1013–1020. ACM, 2010.

[354] Muhammad Tirmazi, Adam Barker, Nan Deng, Md Ehtesam Haque, Zhijing Gene Qin, Steven Hand, Mor Harchol-Balter, and John Wilkes. Borg: the Next Generation. In *EuroSys'20*, Heraklion, Crete, 2020.

[355] Uchi Ugobame Uchibeke, Kevin A Schneider, Sara Hosseinzadeh Kassani, and Ralph Deters. Blockchain Access Control Ecosystem for Big Data Security. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, page 1373–1378. IEEE, 2018.

[356] Sarah Underwood. Blockchain Beyond Bitcoin, 2016.

[357] Olivier Huynh Van and Jeff Gray. Systems and Methods for Determining Endpoint Configurations for Endpoints of a Virtual Private Network (VPN) and Deploying the Configurations to the Endpoints, April 19 2016. US Patent 9,319,300.

[358] Abhishek Verma, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. Large-scale cluster management at google with borg. In *Proceedings of the Tenth European Conference on Computer Systems*, pages 1–17, 2015.

[359] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems. In *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, page 243–264. Springer-Verlag New York, Inc., 2008.

[360] LLC VoltDB. VoltDB Technical Overview, Whitepaper, 2010.

[361] Denny Vrandecic. Architecture for a Multilingual Wikipedia. Technical report, Google, 2020.

[362] Aleksa Vukotic, Nicki Watt, Tareq Abedrabbo, Dominic Fox, and Jonas Partner. *Neo4j in Action*. Manning Publications Co., 2014.

[363] Sameer Wadkar and Madhu Siddalingaiah. Apache Ambari. In *Pro Apache Hadoop*, page 399–401. Springer, 2014.

[364] Guohui Wang, David G Andersen, Michael Kaminsky, Konstantina Papagiannaki, TS Eugene Ng, Michael Kozuch, and Michael Ryan. C-Through: Part-time Optics in Data Centers. In *Proceedings of the ACM SIGCOMM 2010 conference*, page 327–338, 2010.

[365] Guohui Wang, TS Eugene Ng, and Anees Shaikh. Programming Your Network at Run-Time for Big Data Applications. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, page 103–108, 2012.

[366] Rory Ward and Betsy Beyer. BeyondCorp: A New Approach to Enterprise Security. 2014.

[367] Eric W. Weisstein. "convolution." from MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/Convolution.html".

[368] W Hwu Wen-mei. *Programming Massively Parallel Processors*. Morgan Kaufmann, 2010.

[369] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data Mining With Big Data. *Knowledge and Data Engineering, IEEE Transactions on*, 26(1):97–107, 2014.

[370] Miguel G Xavier, Israel C De Oliveira, Fabio D Rossi, Robson D Dos Passos, Kassiano J Matteussi, and Cesar AF De Rose. A Performance Isolation Analysis of Disk-Intensive Workloads on Container-Based Clouds. In *Parallel, Distributed and Network-Based Processing (PDP), 2015 23rd Euromicro International Conference on*, page 253–260. IEEE, 2015.

[371] Miguel Gomes Xavier, Marcelo Veiga Neves, and Cesar Augusto Fonticielha De Rose. A Performance Comparison of Container-based Virtualization Systems for MapReduce Clusters. In *Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on*, page 299–306. IEEE, 2014.

[372] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[373] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. The Microsoft 2016 Conversational Speech Recognition System. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, page 5255–5259. IEEE, 2017.

[374] Qiang Xu, Xin Wang, Jianxin Li, Qingpeng Zhang, and Lele Chai. Distributed Subgraph Matching on Big Knowledge Graphs Using Pregel. *IEEE Access*, 7:116453–116464, 2019.

[375] Xiaolong Xu, Qingxiang Liu, Yun Luo, Kai Peng, Xuyun Zhang, Shunmei Meng, and Lianyong Qi. A Computation Offloading Method Over Big Data for IoT-Enabled Cloud-Edge Computing. *Future Generation Computer Systems*, 95:522–533, 2019.

[376] Corinna Cortes Christopher J.C. Yann LeCun, Courant Institute. The MNIST Database of Handwritten Digits.

[377] Fan Yao, Jingxin Wu, Guru Venkataramani, and Suresh Subramaniam. A Comparative Analysis of Data Center Network Architectures. In *2014 IEEE International Conference on Communications (ICC)*, page 3106–3111. IEEE, 2014.

[378] Xiaomeng Yi, Fangming Liu, Jiangchuan Liu, and Hai Jin. Building a Network Highway for Big Data: Architecture and Challenges. *IEEE Network*, 28(4):5–13, 2014.

[379] Shui Yu, Meng Liu, Wanchun Dou, Xiting Liu, and Sanming Zhou. Networking for Big Data: A Survey. *IEEE Communications Surveys & Tutorials*, 19(1):531–549, 2016.

[380] Yuan Yuan, Meisam Fathi Salmi, Yin Huai, Kaibo Wang, Rubao Lee, and Xiaodong Zhang. Spark-GPU: An Accelerated in-Memory Data Processing Engine on Clusters. In *Big Data (Big Data), 2016 IEEE International Conference on*, page 273–283. IEEE, 2016.

[381] Saima Zafar, Abeer Bashir, and Shafique Ahmad Chaudhry. On Implementation of DCTCP on Three-Tier and Fat-Tree Data Center Network Topologies. *SpringerPlus*, 5(1):766, 2016.

[382] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster Computing with Working Sets. *HotCloud*, 10(10-10):95, 2010.

[383] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. Apache Spark: A Unified Engine for Big Data Processing. *Communications of the ACM*, 59(11):56–65, 2016.

[384] Alexander Zahariev. Google App Engine. *Helsinki University of Technology*, 2009.

[385] Zuzana Zatrochova. *Analysis and Testing of Distributed NoSQL Datastore Riak*. PhD thesis, Masarykova univerzita, Fakulta informatiky, 2015.

[386] Dongpo Zhang. Big Data Security and Privacy Protection. In *8th International Conference on Management and Computer Science (ICMCS 2018)*. Atlantis Press, 2018.

[387] Hao Zhang, Gang Chen, Beng Chin Ooi, Kian-Lee Tan, and Meihui Zhang. In-Memory Big Data Management and Processing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 27(7):1920–1948, 2015.

[388] Xiang Zhang and Yann LeCun. Text Understanding from Scratch. *arXiv preprint arXiv:1502.01710*, 2015.

[389] Ying Zhang, Mohammad Pezeshki, Philemon Brakel, Saizheng Zhang, Cesar Laurent Yoshua Bengio, and Aaron Courville. Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks. *arXiv preprint arXiv:1701.02720*, 2017.

[390] Yu Zhang, William Chan, and Navdeep Jaitly. Very Deep Convolutional Networks for End-to-End Speech Recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, page 4845–4849. IEEE, 2017.