Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd

ISSN,e-ISSN: 2548-1304
Volume:7, Issue:2, pp:111-123, 2022
https://doi.org/10.53070/bbd.1204112
Research Paper

Real-Time Big Data Processing and Analytics: Concepts, Technologies, and Domains

Uğur KEKEVİ [1], Ahmet Arif AYDIN *[1]

[1] Inonu University, Department of Computer Engineering, Malatya, Turkey

(ugurkekevi@gmail.com, arif.aydin@inonu.edu.tr)

Received:Nov.14,2022 Accepted:Nov.27,2022 Published:Dec.07,2022

Abstract— In the digital era, data is one of the most important assets since it conceals valuable information. Developers of data-intensive systems have new challenges at each level of streaming, storing, and processing large quantities of data in a variety of forms and speeds. Obtaining useful information at the proper time and place is also crucial. Since the value of information is inversely proportional to time, real-time data processing and analytics are receiving more attention. Due to the importance of real-time data processing and analytics, this study focuses on real-time data processing concepts and terminology, popular technologies used in real-time data processing and analytics, popular NoSQL storage technologies used in real-time data processing, and real-time data processing application areas. The purpose of this paper is to provide researchers of real-time analysis and developers of data-intensive systems with a comparative perspective on real-time data processing by highlighting the key characteristics of real-time data processing technologies, NoSQL storage technologies, their application domains, and selected examples from previous studies.

Keywords : *Big data analysis, real-time data processing, streaming technologies, NoSQL.*

1. Introduction

The rapid development of affordable technology, widespread use of the internet, and the development of smart and portable devices that can connect to the internet via wireless or cellular networks, as well as the increasing number of different sectors and individuals utilizing these technologies, have significantly increased the amount of data generated in various sizes, formats, speeds, and types (Khan et al., 2021). Moreover, this digital transformation has triggered the generation of data and profoundly altered people's daily activities and habits as a result of the implementation of various applications in education, finance, health care, business, commerce, manufacturing, retail, and government services to facilitate daily activities (Dutta & Jayapal, 2016; Kejariwal et al., 2017; Syed et al., 2021; Tang et al., 2022).

Numerous technologies create data with diverse forms, rates, and volumes in the era of big data. Almost every organization has been streaming, storing, and analyzing data, as well as facing the issues of big data, in order to make profitable decisions for their organization. In the context of big data literature, "*big data*" refers to data sets so enormous and complicated that data processing tools cannot manage them all at once (Yaqoob et al., 2016). Moreover, big data is frequently characterized by using characteristics displayed with Vs. *Volume* refers to large amounts of data; *variety* refers to data in various forms, such as structured, semi-structured, and unstructured data; *velocity* refers to the rate at which incoming data from various sources is collected; *veracity* refers to the dependability or trustworthiness of data; and *value* refers to the beneficial and useful information derived from data to make advantageous decisions (Philip Chen & Zhang, 2014).

Due to the fact that the value of received information is inversely proportional to time, getting relevant information in a short amount of time is a difficult and important endeavor. As a result, an increasing number of firms are emphasizing real-time data processing and analytics in order to preserve their position in the job market. It is vital, for instance, to provide rapid responses to client enquiries, promote products, detect abnormalities, identify fraudulent credit card activity, and support crisis management. In a variety of industries, functional technologies are crucial, especially when quick judgments must be made. High-capacity data flows necessitate real-time data analysis for continuous data updating during information processing, quick decision-making, and rapid data-to-information translation (Lv et al., 2017). Real-time data analysis is essential because of the frequent data changes generated by sensors, internet services, consumers, traffic, and medical systems (Zheng et al., 2015).

In addition, developers of data-intensive systems and database designers have issues in collecting diverse large volumes of data in organized, semi-structured, and unstructured formats without losing data; leveraging the correct

collection of technologies to temporarily store data for real-time analytics; and storing continually increasing data in a feasible data storage system permanently with a proper data model of the target database and database design to accomplish scalability (Aydin & Anderson, 2017). Moreover, data privacy and security are essential concepts that may be realized in a variety of ways, such as by encrypting files, employing the appropriate key-value systems, authenticating users, and granting individuals authorization to access data (Bagga & Sharma, 2019).

Due to the significance of real-time data processing and analytics, we will concentrate on technologies that address the issues of velocity, volume, and veracity. First, we provide key terminology used in real-time data processing and analytics, followed by popular technologies used in real-time data processing and analytics, NoSQL storage technologies utilized in real-time data processing, and application areas for real-time data processing. The primary objective of this paper is to provide a comparative perspective on technologies used in real-time data processing and analytics, presenting popular NoSQL technologies especially used for storing real-time data and providing application domains to assist researchers of big data processing and analytics domains and developers of data-intensive systems in selecting the most appropriate tool for the job.

The paper is structured as follows: in Section 2, relevant research to our own are presented. The third section presents an overview of data processing and analytics terminology. The characteristics of current real-time data processing systems are detailed in Section 4. The Section 5 describes prevalent data storage methods used in real-time analytics. In Section 6, the mentioned application areas and usage scenarios of the technologies for use in real-time data processing are provided. In Section 7, discussion and future work are provided, while the conclusion is presented in Section 8.

2. Related Works

This section highlights research on big data principles, tools, real-time data processing and analytics technology, and big data processing challenges. Numerous studies have been produced as a result of the significance of real-time, big data processing technology. In this part, however, only a chosen set of publications are studied.

In (Yaqoob et al., 2016), a general perspective on the history of big data, its present and future is presented. The term "*big data*" is explained, and the 3Vs of big data, which are volume, variety, and speed, are also given. Structured, unstructured, and semi-structured data from big data sources are mentioned. Technologies that analyze this data are classified, and detailed analyses are made of batch processing and stream processing technologies.

In (Philip Chen & Zhang, 2014), general definitions and challenges of big data, including data collection, transfer, curation, analysis, and visualization, are mentioned. By evaluating big data analysis methods and technology in particular, the positives and cons are highlighted.

In (Acharjya & Ahmed P, 2016), the 3V (volume, velocity, and variety) definition of big data and the fourth V (value) are discussed, and the challenges, problems, and tools of big data are highlighted by categorizing the challenges of big data into the following categories: data storage and analysis, knowledge discovery, computational complexities, scaling, visualization, and information security. In addition, open research issues in big data analytics are presented regarding IoT, cloud computing, bio-inspired computing, and quantum computing. Lastly, features of big data analytics tools are presented.

In (Bajaber et al., 2016), it is discussed how much room big data consumes in our lives and how much information it generates. Four classes of big data analysis systems are studied. In general, big data processing systems and technologies are discussed, as well as big SQL processing systems, graphic analysis systems, and real-time data processing technologies. Data processing issues and the advantages and disadvantages of various technologies are discussed.

In (Liu et al., 2014), focusing on Hadoop, the Hadoop engine, which plays a crucial role in the analysis of large data sets, is presented. In-depth research was conducted on real-time system designs and data integration challenges, and their working principles were explained. The most prominent systems for real-time data analysis are outlined.

In (Dutta & Jayapal, 2016), real-time data analysis tools and techniques used in big data analytics are examined. Resources in the literature about the advantages and disadvantages of these technologies have been compiled. In addition, the use cases of real-time technologies are discussed.

In (Gürcan & Berigel, 2018), the authors classify analytics, data storage, analysis, and assessment, as well as data retrieval for real-time data processing. In addition, the authors provide a lifecycle for real-time data processing and existing tools and technology associated with its phases and they discuss the characteristics of the following big data tools: Flume, Kafka, Nifi, Storm, Spark Streaming, S4, Flink, Samza, Hbase, Hive, Cassandra, Splunk, and Sap Hana. In addition, challenges associated with the volume, diversity, velocity, scalability, and visualization of real-time big data processing are highlighted.

In (Doğuç & Aydin, 2019), the authors presented streaming and batch-style data processing and an overview of the CAP theorem. Based on CAP's theorem, the authors also discussed the characteristics of popular NoSQL databases, particularly those used for streaming processing, so that the appropriate NoSQL databases could be selected for the task.

In (Saranya et al., 2020), the authors provide a classification of traditional data-streaming technologies (random sampling, sliding window, histograms, and sketches), data mining (clustering, classification, pattern recognition, association rule mining), and big data technologies (Apache Storm, Apache Spark, Apache Flume, Apache Kafka, and Apache Flink). It was demonstrated that big data technologies are the best solution to manage streaming data since they can rapidly handle massive volumes of organized and unstructured data.

In (Nasr, 2021), the author provided a background for streaming and batch data processing. This research detailed the architecture and programming paradigms of the Apache Beam, Apache Spark, and Apache Flink data processing frameworks. Then, the performance of these three technologies is evaluated by comparing their responses to three queries.

In (Saloot & Pham, 2021), the authors provide an overview of Apache Kafka, Apache Storm, and distributed processing. With Apache Storm and Apache Kafka, they also provide a real-time framework for natural language processing. Apache Kafka is an essential component of stream processing since it allows developers to store, read, and write data from data sources. Apache Kafka does not, however, provide distributed computing. Therefore, Apache Storm is utilized to process text streams since it is not constrained by a certain programming paradigm, language, or data structure.

3. Data Processing Paradigms

This section defines key terms used in real-time data processing and analytics, including "*streaming analytics*," *"real-time analytics,"* and "*batch data processing"* that are utilized to extract valuable information from data. Figure 1 presents an overview of big data journey starting from data generation sources to permanent databases.

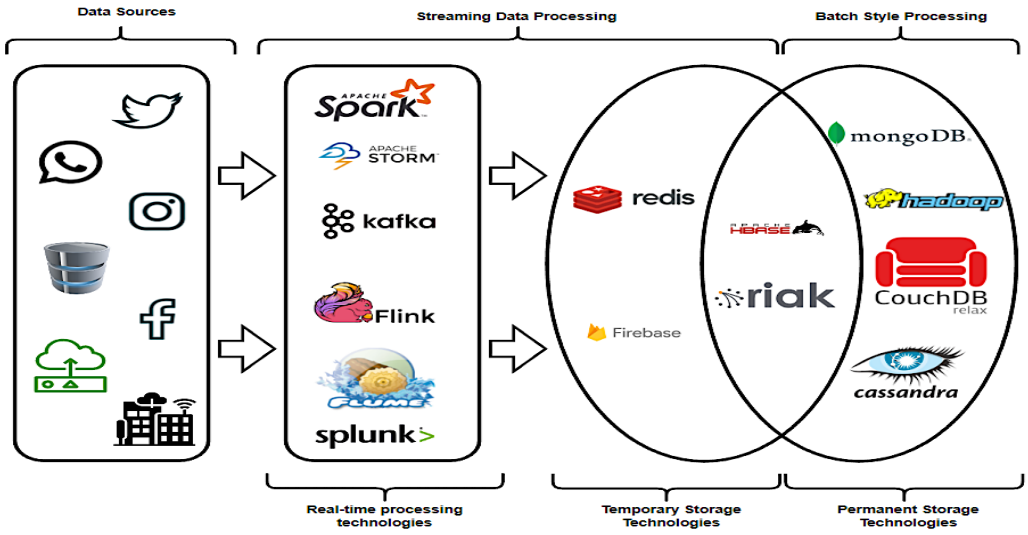

Figure 1. An overview of data processing and storage technologies

Real-time analytics (also called near real-time analytics) or streaming analytics are performed on fast data (data in motion). Real-time analytics is crucial since it aims to produce rapid responses in a short period of time after data is captured from data generation sources. Due to the significance of real-time data processing and analytics, real-time analytics is in high demand for applications in the following fields: banking applications for fraud detection; network applications to detect and attract; and crisis informatics applications for immediate emergency response (Han et al., 2014).

On the other hand, using data mining, machine learning, and statistical techniques, batch-style data processing performs operations on completed datasets with the purpose of uncovering hidden patterns, making predictions, or performing exploratory analytics such as user behavior prediction, product recommendation, trend determination, fraud detection, emergency, or disease monitoring (Aydin, 2016).

Moreover, unlike batch data processing, real-time analytics is performed on a small amount of data (data chunks) right after capturing fresh data from various resources. Figure 1 depicts the data processing life cycle, beginning with data generation from various sources; streaming data processing tools that capture data from resources and process it; temporal data storage technologies (in-memory technologies) that temporarily store data

to support real-time analytics; and permanently storage technologies that store large amounts of data for future data processing and analytics to perform batch-style data processing and analytics. Each streaming and storage technology is described in depth in the following sections.

4. Real-Time Data Processing Technologies

This section presents a list of prominent technologies used in real-time data processing that have been recognized in published research. Table 1 provides a comparative list of real-time data processing technologies and key characteristics of each data processing technology is explained next.

Table 1. Feature comparison of the studied real-time data processing technologies

References	Technology	Written in	Supported data formats	Data processing mechanism	Batch Processing
(Liu et al., 2014; Philip Chen & Zhang, 2014; Ryan, 2019; Yaqoob et al., 2016)	Kafka	Scala, Java	Csv, JSON, Avro	Topic	yes
(Bajaber et al., 2016; Gürcan & Berigel, 2018; Nasr, 2021)	Flink	Scala, Java	Csv, JSON, Avro, Raw, Orc, Canal orc, Maxwell cdc	Logic – State Tasks	Yes
(Liu et al., 2014; Philip Chen & Zhang, 2014; Yaqoob et al., 2016)	Storm	Clojure	Hdfs, Avro	Spouts, Bolts, Topology	No
(Abdul Ghani et al., 2021; Saranya et al., 2020; Vohra, 2016)	Flume	Java	Hdfs	Event	Yes
(Acharjya & Ahmed P, 2016; Nasr, 2021; Oussous et al., 2018)	Spark	Scala	Excel, JSON, Csv, Text, Hive tables, Parquet files, Avro files	RDD, Data frame	Yes
(Philip Chen & Zhang, 2014; Yaqoob et al., 2016)	Splunk	C++	Xml, JSON	Serach heas, indexers, forwarders	Yes

In addition, Figure 2 illustrates the popularity of the five real-time data processing technologies discussed in this section. According to Google Trends (*Google Trends*, 2022), Apache Spark and Apache Kafka were the two most popular real-time processing systems over the previous decade as shown in Figure 2.

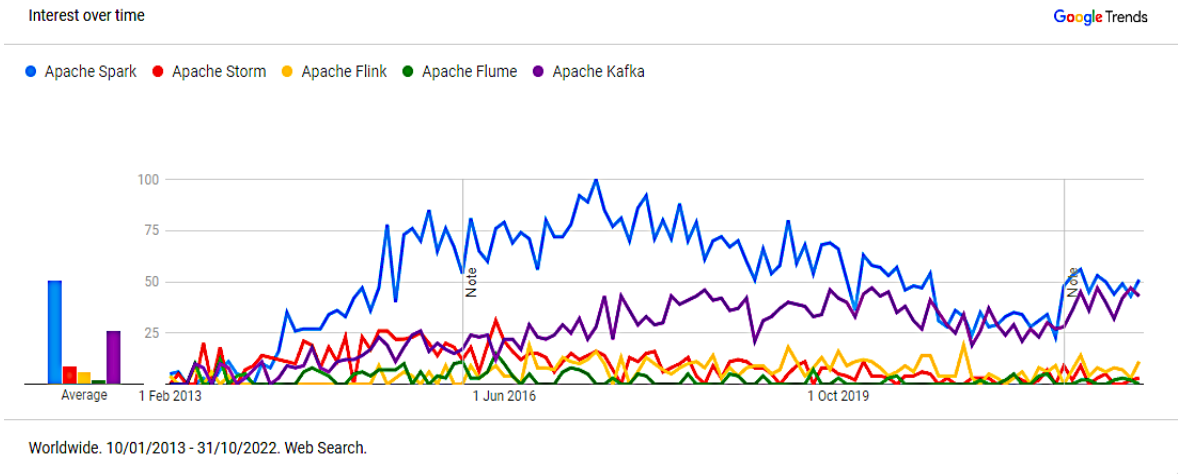

Figure 2. Trends in real-time data processing technologies(*Google Trends*, 2022)

4.1. Apache Storm

Apache Storm (Apache Software Foundation, 2022i) is an open-source distributed computing system designed for real-time data processing. Storm's operational paradigm is Dataflow, in which real-time data flows continuously over a network of transformation assets (Liu et al., 2014). Data moves across spouts and bolts in the Storm topology. Spout is the source of the stream and reads data from the source before distributing it across the topology (Apache Software Foundation, 2022i). Although storm and Hadoop clusters look identical at first glance, storm has a distinct architecture whereas Hadoop employs an application-specific map/reduce (Philip Chen & Zhang, 2014). Two nodes comprise a storm cluster: nimbus and supervisor nodes. Nimbus is responsible for fault detection, task scheduling, and node assignment. The supervisor is responsible for carrying out the tasks assigned by the nimbus and initiating and concluding the procedure (Acharjya & Ahmed P, 2016; Yaqoob et al., 2016).

4.2. Splunk

Splunk (*Splunk*, 2022) is a platform for real-time data analysis developed to handle information technology issues and analyze data generated by industrial machinery. Splunk can display results visually, as a report, or as an alert. Log file work benefits greatly from the processing of both structured and unstructured data, as it produces results in the form of searches, analytical reports, and tables (Acharjya & Ahmed P, 2016; Philip Chen & Zhang, 2014; Yaqoob et al., 2016).

4.3. Apache Kafka

Apache Kafka (Apache Software Foundation, 2022g) is a scalable software framework for managing and making judgments regarding massive volumes of streaming data utilizing in-memory analytics (Ryan, 2019). Four prominent properties of Kafka are distributed processing, high throughput, persistent messaging, and disk structures. It combines online and offline processing of these datasets with real-time processing of activity data and operational data (Philip Chen & Zhang, 2014; Yaqoob et al., 2016). Kafka distributes a partition across several nodes because it assigns sequential identities to messages. Messages are kept configured for a predetermined amount of time before being erased. Due to its capacity to monitor data pipelines feeding operational data and collect statistical information from remote applications, Kafka is ideally suited for real-time data processing and analysis (Liu et al., 2014).

4.4. Apache Spark

Apache Spark (Apache Software Foundation, 2022h) in contrast to Hadoop, employs system memory to run complex analyses on large - scale data volumes (Oussous et al., 2018). Spark's distributed data processing technology, Spark Stack Processing, and the Spark Streaming Library are capable of doing streaming data processing. Spark, a data processing framework that utilizes micro-heaps, offers data processing support through a range of libraries, including those for SQL queries, graphics, and machine learning (Nasr, 2021). Since it is based on the HDFS architecture of Hadoop, it supports all Hadoop-supported file systems (Oussous et al., 2018). Spark was designed with the Scala programming language and runs on the Java virtual machine. It is interoperable with popular programming languages such as Java, Python, and R (Acharjya & Ahmed P, 2016).

4.5. Apache Flink

Apache Flink (Apache Software Foundation, 2022c) is a versatile alternative to MapReduce that supports both batch and real-time data processing. It's an open-source data processing project created by Apache using Java and Scala (Bajaber et al., 2016; Gürcan & Berigel, 2018). Flink analyzes data tuple by tuple and offers a large number of libraries for machine learning and graphics processing (Nasr, 2021).

4.6. Apache Flume

Apache Flume (Apache Software Foundation, 2022d) was built by Apache primarily as a tool for rapidly gathering, aggregating, and streaming enormous volumes of daily web browsing logs onto HDFS. There are three main phases to Flume's operation: the Flume source, the Flume channel, and the Flume pool. The Flume resource consumes server-provided data. The occurrence, which represents the unit of flume flow, is communicated to the flume channel until it is consumed by the pool (Vohra, 2016). Apache Flume enables the collection of information from large amounts of unstructured and semi-structured data, notably from social media networks, because to their reliability, flexibility, and distributed and robust structure (Abdul Ghani et al., 2021; Saranya et al., 2020; Singh et al., 2018).

5. NoSQL Storage Technologies for Real-Time Data Processing

This section presents an overview of NoSQL data storage systems, focusing on those used for real-time data processing tasks and applications. Table 2 provides a feature comparison of the studied NoSQL data storage technologies.

Table 2. Feature comparison of presented NoSQL storage technologies

References	Technology	Data Model	Storage	Querying Support	Data Types	Consistency
(Azzedin, 2013)	Hadoop	Key-value	Disk	Hive QL	HDFS	Eventual
(Zheng et al., 2015)	Cassandra	Wide-column	Disk	CQL	BLOB, JSON	Eventual
(Baron, 2015)	Riak	Key-value	Memory and Disk	SQL	Flags, Registers, Counters, Sets, Maps	Eventual
(Baron, 2015)	Redis	Key-value	Mermory and Disk	-	Strings, Hashes, lists, sets, sorted sets	Eventual
(Oussous et al., 2018)	Hbase	Wide-column	Memory and Disk	Jaspersoft, Drill	XML, Protobuf, Binary	Strong
(Moroney, 2017a)	Firebase	Document	Disk	FireSQL	JSON	Strong
(Diogo et al., 2019)	MongoDB	Document	Disk	Built-in (find)	BSON, JSON	Eventual Immediate
(Lennon, 2009)	CouchDB	Document	Disk	Javascript HTTP (API)	JSON	Eventual

Moreover, Figure 3 illustrates the popularity of the in-memory and persistent NoSQL storage systems illustrated in Figure 1 and used for real-time data processing. According to DB-Egines (*DB-Engines*, 2022), Figure 3 demonstrates the popularity of the aforementioned NoSQL technologies. Each presented NoSQL data storage technology is explained next.

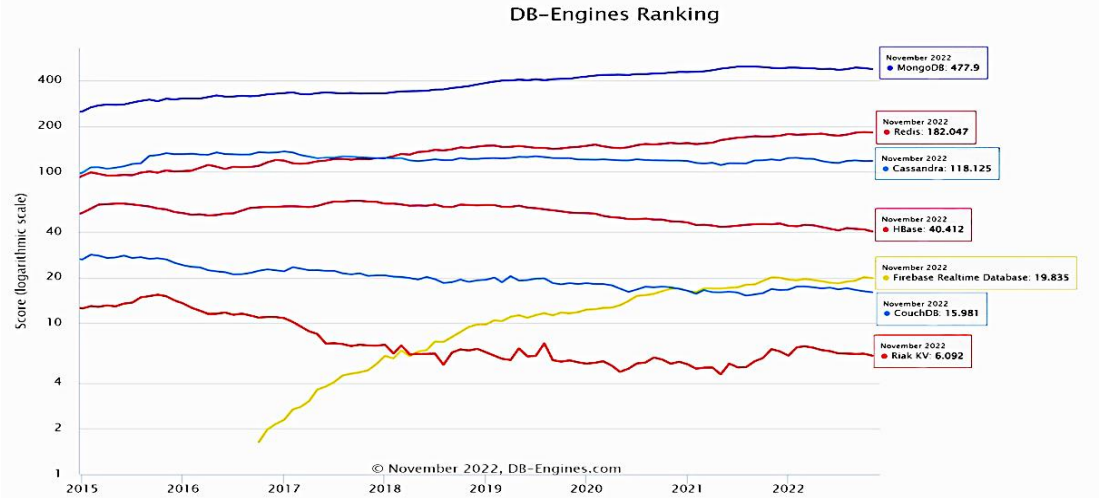

Figure 3. NoSQL storage technology trends

5.1. Apache Hadoop

Apache Hadoop (Apache Software Foundation, 2022e) is the de facto standard for storing massive volumes of big data, and the majority of Apache streaming technologies have been created to be integrated with Apache Hadoop. Hadoop Distributed File System (HDFS) is a Java-based file system for storing massive volumes of unstructured data. Hadoop MapReduce is a framework for processing enormous quantities of data stored in HDFS. It is scalable, cost-effective, and appropriate for massive data sets, once-written and once-read analytics data (Azzedin, 2013). The MapReduce architecture functions initially as a map operation and afterwards as a reduction operation. However, because the intermediate data following the map operation is stored to disk prior to reduction,

this creates a delay in real-time analysis (Philip Chen & Zhang, 2014). The reduction step begins only once the map phase has been completed. Moreover, all intermediate data created during the mapping step are written to disk prior to being transferred to the reducers for the following phase. This results in considerable processing delays. Due to Hadoop's significant latency, real-time analytics are almost difficult to execute. Due to this delay, the development of technologies for real-time data analysis has begun. The speed of data collection, querying, and display with these technologies is comparable to writing or reading to a disk, and they are almost as close to real time (Liu et al., 2014).

5.2. Cassandra

Cassandra (Apache Software Foundation, 2022a), an open-source column-based (wide-column or columnar) NoSQL database, was developed by Facebook (Lakshman & Malik, 2014). The CAP theorem encompasses the availability (A) and partition tolerance (P) components. In data queries, CQL (Cassandra Query Language) is utilized as the query language (Doğuç & Aydin, 2019). The columnar structure of the CQL programming language has rows, a column family, a partition key, and a key field. The partition key is unique to each row and can contain several columns. The object classes are stored in the column family, and each object entry is defined as a row. The important area is an umbrella structure composed of many columns (Diogo et al., 2019). Apache Cassandra is integrated with streaming applications because it enables availability and accessibility and manages heavy write loads (Schram & Anderson, 2012).

5.3. Riak

Riak (*Riak*, 2022) is an open-source key-value data storage technology provides text, JSON, and XML access. It is compatible with several Apache data processing technologies. In addition, with the Riak cloud storage (Riak CS) system built on top of Riak, objects work in harmony with Amazon S3 (Doğuç & Aydin, 2019). Create, read, update, and delete actions are accessible over HTTP REST using Riak. A virtual key field is provided for the key-value structure that stores RIAK objects using the bucket approach. This key field can be used to create non-default configurations. It is possible to complete without a bucket (Baron, 2015). Moreover, Riak is suited for real-time analytics since it utilizes memory to perform quick operations.

5.4. Redis

Redis (*Redis*, 2022) is an in-memory key-value NoSQL database that operates in real time. Additionally, data can be saved on the disk if a command is sent. Due to its in-memory operation, it is particularly important in situations where high performance is required, such as caching and messaging clients. It is appropriate for real-time analytic contexts since it supports several data structures, including strings, hashes, sets, and lists (Doğuç & Aydin, 2019; Guo & Onstein, 2020). With Redis clustering, it achieves great availability and consistency performance (Diogo et al., 2019). Lists and stacks may be used to efficiently store and retrieve data (Baron, 2015).

5.5. Apache HBase

Apache HBase (Apache Software Foundation, 2022f) is a distributed, non-relational, column-based, open-source database. It can accommodate a large number of columns and rows and allows column groups to be stored independently from row-based relational databases. Apache HBase reads and writes are strongly consistent. This means that all reads and writes to a single row in Apache HBase are atomic. Each concurrent reader and writer can make safe assumptions about the state of a row. Multi-versioning and time stamping in Apache HBase contribute to its strongly consistent model. In addition, Apache HBase is appropriate for real-time data because to its dynamic nature and it supports block cache and Bloom filters (Doğuç & Aydin, 2019; Guo & Onstein, 2020). APIs are used instead of traditional query languages to query data. It offers modular and linear scalability, real-time querying, and automatic, tunable table fragmentation. It uses HDFS-like master nodes and has multiple map files (Oussous et al., 2018).

5.6. Firebase

Firebase (Moroney, 2017b) is the cloud foundation that synchronizes client data in real time for synchronized transactions. It has been acquired by Google and operates on Google's cloud infrastructure (Doğuç & Aydin, 2019). The data is recorded in JSON format, and the final data is automatically saved. This enables the user to both update the record and push the change to all connected clients. Firebase is useful for real-time data processing and analytics since it supports the cloud, decreases latency, and enables the connection of IoT and smart devices. In addition, developers may create JavaScript applications and publish them to the Firebase Cloud (Chatterjee et al., 2018; Li et al., 2018).

5.7. MongoDB

MongoDB (*MongoDB*, 2022) is a robust NoSQL database with dynamic document-based schemas that are horizontally scalable. In contrast to relational databases, MongoDB documents are stored in the BSON format

similar to JSON and are organized as collections. A document includes all features of an object's data, whereas a collection is an ordered version (Diogo et al., 2019). C++, JavaScript, Python, and Go are the programming languages employed. Moreover, the change stream capability makes use of real-time data (Doğuç & Aydin, 2019). In addition, MongoDB enables full text search across all fields. This greatly improves MongoDB's popularity.

5.8. Apache CouchDB

Apache CouchDB (Apache Software Foundation, 2022b) is a document-oriented, cross-platform, open-source NoSQL database that was initially created in C++ before being translated to Erlang. It does not restrict the data (such as its size or area) and saves it independently of the schema. The data is stored in JSON format, which enables a variety of formats, such as arrays, objects, and complicated fields. It quickly transfers data through JSON across several server cluster regions. CouchDB works with MapReduce for efficient and high-volume data transport (Lennon, 2009; Miler et al., 2011). CouchDB is designed for semi-structured data. There are also studies on the efficiency of using CouchDB in real-time systems. In (Alhomsi et al., 2019), an example is presented of how CouchDB can be used as an alternative to the latency, compatibility, and maintenance cost of Firebase databases in medical simulators.

6. Application Areas

This section outlines some of the most significant application domains where real-time data processing is already utilized. In addition, Table 3 gives examples of domain-specific real-time data processing applications.

Table 3. Application domains and data processing technologies

Domain	References	Technologies	Usage
Recommendation Systems	(Xie et al., 2016)	Spark	Performance impact of ALS (least squares algorithm) based collaborative filtering algorithm on Spark.
Health Care Services	(Sudhakar Yadav et al., 2018)	Storm, Kafka	Active monitoring of patients and automated telemedicine service.
Smart Cities	(Hegde et al., 2021)	Flume, Flink, Storm, Spark	Service applications such as traffic, agriculture, cybercrime, security service.
Crisis Management	(Aydin & Anderson, 2017; de Castro Martins et al., 2018)	Spark	Supporting emergency situations
Economy	(Gavrilenko et al., 2019)	Spark	Analysis and classification of data by running program codes
Energy Systems	(Gibadullin et al., 2019; Hamadou et al., 2020; Krishnamoorthy & Udhayakumar, 2021)	Kafka, Flume, Spark	Collection, storage, and analysis of high-volume data in energy systems
IOT	(Nambiar et al., 2020; Nasiri et al., 2019; Verma et al., 2017)	Storm, Flink, Spark	Modeling of flow data control, evaluation of real-time technology suitable for IOT devices for smart cities

6.1. Recommendation Systems

The volume challenge of big data makes it difficult to identify the information, product, or location that we seek or that is suitable for us. Numerous categories, such as movies, games, electronic equipment, transport routes, and holiday spots, utilize recommendation algorithms that impact the decision-making process. Due to the inadequacies of one of the most extensively used recommendation algorithms, the least squares algorithm (LSA), the Spark framework was utilized to construct a recommendation algorithm. It has been shown to produce better results than the ALS algorithm in (Xie et al., 2016). In addition to real-time analytical technologies, interactive systems are also available. The process of film review was carried out utilizing a user-scoring system, and suggestions were made by evaluating safe video content (Yang et al., 2017). In (Jiang et al., 2016), individuals were provided trip suggestions based on their own travel information and their interests.

6.2. Health Care Services

In different sections of the healthcare industry, various measuring instruments are used for diagnosis, and the data generated by these equipment is analyzed. It is applied in automatic patient monitoring and telemedicine services, as examples of the aforementioned application sectors (Sudhakar Yadav et al., 2018). In addition to real-time analytical technologies, interactive systems are also available. Patients were exposed to the mHealth application, which use virtual reality glasses to improve the user experience in high-pressure rooms and assess the efficacy of medicinal equipment (Lv, Chirivella, et al., 2016). Voice recordings of patients can be used to analyze the healing process. Using image processing, a variety of diseases have been identified.

6.3. Smart Cities

Geographical location data and urban traffic data are sources of big data generation. Numerous studies have been performed to extract information from this vast volume of data. Additionally, several experiments were undertaken in virtual settings. With the development of urban technology, the usage of intelligent devices has begun to increase. This subject has been the subject of much research, as has the notion of a smart and connected community in cities, especially in light of the development of IoT. The generation of enormous quantities of IoT data for analysis and the analysis of structured data are essential to the urbanization process. In (Hegde et al., 2021), the real-time data processing tools Flink, Spark, Flume, and Storm for the production of smart city data are investigated. Smart cities use both analytical technologies and real-time systems. Using mobile resources and IoT data from the community (Sun et al., 2015), it hopes to adopt more suitable actions for the city's life and future goals. In (Lv, Li, et al., 2016), a geographic information system for virtual reality-based smart city efforts is described. Their web platform includes information on city residents, traffic, and three-dimensional representations of buildings. Real-time loading and processing of data obtained based on customer requirements.

6.4. Crisis Management

In emergency situations, rapid decisions are required. For instance, in the fight against fire, timely decisions by firefighters might prevent possible losses or injuries. The structure of the hazardous elements in the environment and the quick collection of comparable data remove the need for incorrect responses in the case of a potential fire. Teams of graduate students from the Brazilian Aviation Technology Institute (de Castro Martins et al., 2018) conducted a warning and crisis management project using big data and IoT for crisis management by employing Apache Hadoop and Apache Spark real-time analytical technologies. Additional interactive systems are accessible. To address these challenges in firefighting, a smart firefighting system that can interpret data from fire sensors in real time is a helpful tool for firefighters to make timely choices and prevent losses (Beata et al., 2018). Moreover, IDCAP platform was introduced in (Aydin & Anderson, 2017) with the intention of enabling real-time data collecting and analytics to enable crisis informatics research. The IDCAP utilizes cutting-edge data processing (Apache Spark) and NoSQL storage technologies (Redis, Apache Cassandra) to monitor and analyze tweets in real time.

6.5. Economy

Perhaps economists have the most need for data analysis. Rapidly-changing economic data necessitates real-time data analysis to enhance economic forecasting and provide reliable predictions utilizing past data. Apache Spark was utilized by the Simple Grid Project to do statistical analysis and data categorization, as well as execute project programs (Gavrilenko et al., 2019) by enabling the employment of specialist software in the processing of economic data. Croushore and Stark also built quarterly data sets for macroeconomics to determine the impact of data changes on estimations (Croushore & Stark, 2001).

6.6. Energy Systems

Intelligent power grids are becoming increasingly common in the electricity management system. In order to save resources, energy consumption and generation data must be collected, stored, and analyzed using a smart grid. Depending on the application domain, the design of real-time data analysis systems differs. Existing real-time data analysis and NOSQL data storage technologies are investigated for a real-time smart grid design, and a distributed energy data management system is addressed and tested in a real location (Gibadullin et al., 2019). In the context of the Energy Data Lake project, batch and flow data processing, storage, and collection technologies such as Apache NIFI, Apache Flume, Apache Sqoop, and Apache Spark were examined and implemented into the system architecture (Hamadou et al., 2020). In (Krishnamoorthy & Udhayakumar, 2021), a system is shown for analyzing data from wind energy sources and managing energy for power distribution using Hadoop and Apache Spark.

6.7. Internet of Things Technologies

Smart phones, household appliances, and wearable devices that can connect to the internet and communicate with one another outside of computers introduced the phrase "Internet of Things" (IoT) to the world of technology.

Numerous researchers have undertaken IoT related research and continue to do so. The major emphasis of the study was on the system's hardware sensor detection and content delivery networks. The architecture of the IoT network and the storage and analysis of the data have not been studied due to the proliferation of technology, the rise in the amount of data generated, and the complexity of linkages. The investigation conducted by the authors of (Verma et al., 2017) comprised research on IoT analytics use cases, real-time technology analysis, and network technique. Equally essential are data security and management, such as the processing and storage of flow data on IoT devices (Nambiar et al., 2020). For IoT data control, research is conducted on how to specify and implement Apache Storm's access control on streaming data. In smart cities, IoT devices create massive amounts of data that must be processed quickly. Three unique frameworks were selected and examined in terms of scalability and resource consumption (Nasiri et al., 2019) in order to evaluate the effectiveness of high-throughput data processing tools on IoT data for varied applications. In addition, research is conducted on real-time applications that enable mobile devices to become an integral part of our lives and save time due to workload, especially in large cities, and that enable IoT-based smart devices to be controlled remotely and provide instant notification in cases such as home security and fire (Erzi & Aydin, 2020).

7. Discussion and Future Work

Due to the significance of real-time data processing, several technologies have been developed to address the challenges encountered by big data for streaming, storage, and analytics. The purpose of this study is to explain the concepts of streaming and batch data processing. Then, we outline the capabilities of prominent technologies for streaming data processing and real-time analytics, as well as temporary and permanent storage systems, to enable real-time data processing jobs.

In this era of big data, one size does not fit all due to the variety of domains, demands, and requirements. Thus, understanding and internalizing a domain need requires to have systematic meetings with stakeholders is crucial to identify needs, priorities and user requests. In addition, it is critical to understand use cases of data processing and storage technology use in order to internalize usage purposes through the application of prior research and experiences. Thus, we included the examples of application areas to illustrate the relationships between domains and employed technology in published research works. Developers of real-time data processing and analytics systems must follow the best practices in industry and research to accomplish their domain specific design goals.

In our future work, we intend to develop a real-time data-intensive system that incorporates the aforementioned leading technologies for real-time data processing, streaming analytics, and scalable storage. Additionally, we would like to utilize visualization tools to show our analytics data.

8. Conclusions

Real-time data processing, NoSQL storage systems, and big data analytics are essential subjects in this era of big data. Consequently, this study focuses on the subsequent aspects: The terminology and principles relevant to real-time data processing and analytics are provided. It is discussed the capabilities, characteristics, operational logic, and applications of prevalent real-time data processing technologies. Popular NoSQL storage solutions, particularly those used in real-time data processing, are described, along with real-time data processing application areas. This article's primary objective is to provide a comparative study on real-time data processing and NoSQL storage technologies in order to support researchers in data processing and analytics as well as developers of data-intensive systems by highlighting the key features, capabilities, and characteristics of real-time data processing and storage technologies. Moreover, Real-time data processing and analytics works are presented in the context of previous research examples to illustrate which areas utilize real-time data processing technologies.

References

Abdul Ghani, N. B., Hamid, S., Ahmad, M., Saadi, Y., Jhanjhi, N. Z., Alzain, M. A., & Masud, M. (2021). Tracking Dengue on Twitter Using Hybrid Filtration-Polarity and Apache Flume. *Computer Systems Science and Engineering*, *40*(3), 913–926. https://doi.org/10.32604/CSSE.2022.018467

Acharjya, D. P., & Ahmed, K. (n.d.). *A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools*. www.ijacsa.thesai.org

Acharjya, D. P., & Ahmed P, K. (2016). A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools. *International Journal of Advanced Computer Sciences and Applıcatıons*, *7*(2), 511–518.

Alhomsi, Y., Alsalemi, A., al Disi, M., Bensaali, F., Amira, A., & Alinier, G. (2019). CouchDB Based Real-Time Wireless Communication System for Clinical Simulation. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, 1094–1098. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00182

Apache Software Foundation. (2022a). *Cassandra*. https://cassandra.apache.org/_/index.html

Apache Software Foundation. (2022b). *CouchDB*. https://couchdb.apache.org/
Apache Software Foundation. (2022c). *Flink*. https://flink.apache.org/
Apache Software Foundation. (2022d). *Flume*. https://flume.apache.org/
Apache Software Foundation. (2022e). *Hadoop*. https://hadoop.apache.org/
Apache Software Foundation. (2022f). *HBase*. https://hbase.apache.org/
Apache Software Foundation. (2022g). *Kafka*. https://kafka.apache.org/
Apache Software Foundation. (2022h). *Spark*. https://spark.apache.org/
Apache Software Foundation. (2022i). *Storm*. https://storm.apache.org/
Aydin, A. A. (2016). *INCREMENTAL DATA COLLECTION & ANALYTICS THE DESIGN OF NEXT-GENERATION CRISIS INFORMATICS SOFTWARE*.
Aydin, A. A., & Anderson, K. M. (2017). Batch to Real-Time : Incremental Data Collection & Analytics Platform. *Proceedings of the 50th Hawaii International Conference on System Sciences*, 5911–5920.
Azzedin, F. (2013). Towards a scalable HDFS architecture. *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, 155–161. https://doi.org/10.1109/CTS.2013.6567222
Bagga, S., & Sharma, A. (2019). Big Data and Its Challenges: A Review. *Proceedings - 4th International Conference on Computing Sciences, ICCS 2018*, 183–187. https://doi.org/10.1109/ICCS.2018.00037
Bajaber, F., Elshawi, R., Batarfi, O., Altalhi, A., Barnawi, A., & Sakr, S. (2016). Big Data 2.0 Processing Systems: Taxonomy and Open Challenges. *Journal of Grid Computing*, *14*(3), 379–405. https://doi.org/10.1007/s10723-016-9371-1
Baron, C. A. (2015). NoSQL Key-Value DBs Riak and Redis. In *Database Systems Journal: Vol. VI* (Issue 4).
Beata, P. A., Jeffers, A. E., & Kamat, V. R. (2018). Real-Time Fire Monitoring and Visualization for the Post-Ignition Fire State in a Building. *Fire Technology*, *54*(4), 995–1027. https://doi.org/10.1007/s10694-018-0723-1
Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018). Real-time Communication Application Based on Android Using Google Firebase. *International Journal of Advance Research in Computer Science and Management Studies*, *6*(4). www.ijarcsms.com
Croushore, D., & Stark, T. (2001). A real-time data set for macroeconomists. In *Journal of Econometrics* (Vol. 105). www.elsevier.com/locate/econbase
DB-Engines. (2022). https://db-engines.com/en/
de Castro Martins, J., Mancilha Pinto, A. F., Junior, E. E. B., Goncalves, G. S., Louro, H. D. B., Gomes, J. M., Filho, L. A. L., da Silva, L. H. R. C., Rodrigues, R. A., Neto, W. C., da Cunha, A. M., & Dias, L. A. V. (2018). Using big data, internet of things, and agile for crises management. *Advances in Intelligent Systems and Computing*, *558*, 373–382. https://doi.org/10.1007/978-3-319-54978-1_50
Diogo, M., Cabral, B., & Bernardino, J. (2019). Consistency models of NoSQL databases. In *Future Internet* (Vol. 11, Issue 2). MDPI AG. https://doi.org/10.3390/fi11020043
Doğuç, T. B., & Aydin, A. A. (2019). CAP-based Examination of Popular NoSQL Database Technologies in Streaming Data Processing. *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*.
Dutta, K., & Jayapal, M. (2016). *Big Data Analytics for Real Time Systems*. https://www.researchgate.net/publication/304078196
Erzi, H. M., & Aydin, A. A. (2020). IoT Based Mobile Smart Home Surveillance Application. *4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings*. https://doi.org/10.1109/ISMSIT50672.2020.9255303
Gavrilenko, I., Sharma, M., Litmaath, M., Tikhomirova, T., Gavrilenko, I., Sharma, M., Litmaath, M., & Tikhomirova, T. (2019). *DYNAMIC APACHE SPARK CLUSTER FOR ECONOMIC MODELING*.
Gibadullin, R. F., Baimukhametova, G. A., & Perukhin, M. Y. (2019). Service-Oriented Distributed Energy Data Management Using Big Data Technologies; Service-Oriented Distributed Energy Data Management Using Big Data Technologies. In *2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*.
Google Trends. (2022). https://trends.google.com/trends/
Guo, D., & Onstein, E. (2020). State-of-the-art geospatial information processing in NoSQL databases. In *ISPRS International Journal of Geo-Information* (Vol. 9, Issue 5). MDPI AG. https://doi.org/10.3390/ijgi9050331
Gürcan, F., & Berigel, M. (2018). Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges; Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges. In *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*.
Hamadou, H. ben, Bach Pedersen, T., & Thomsen, C. (2020). The Danish National Energy Data Lake: Requirements, Technical Architecture, and Tool Selection. *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, 1523–1532. https://doi.org/10.1109/BigData50022.2020.9378368
Han, H., Yonggang, W., Tat-Seng, C., & Xuelong, L. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. *Access, IEEE*, *2*, 652–687. https://doi.org/0.11 09/ACCESS.2014.2332453

Hegde, G. P., Tech, M., Hegde, N., & Seetha, M. (2021). SMART CITY DATA GENERATION FOR IOT APPLICATIONS USING ESSENTIAL HADOOP FRAMEWORKS. *Embracing Change & Transformation-Breakthrough Innovation and Creativity*, 153–160.

Jiang, S., Qian, X., Mei, T., & Fu, Y. (2016). Personalized Travel Sequence Recommendation on Multi-Source Big Social Media. *IEEE Transactions on Big Data*, *2*(1), 43–56. https://doi.org/10.1109/tbdata.2016.2541160

Kejariwal, A., Kulkarni, S., & Ramasamy, K. (2017). *Real Time Analytics: Algorithms and Systems*. http://arxiv.org/abs/1708.02621

Khan, M. F., Azam, M., Khan, M. A., Algarni, F., Ashfaq, M., Ahmad, I., & Ullah, I. (2021). A Review of Big Data Resource Management: Using Smart Grid Systems as a Case Study. *Wireless Communications and Mobile Computing*, *2021*. https://doi.org/10.1155/2021/3740476

Krishnamoorthy, R., & Udhayakumar, K. (2021). Futuristic bigdata framework with optimization techniques for wind energy resource assessment and management in smart grid. *Proceedings of the 7th International Conference on Electrical Energy Systems, ICEES 2021*, 507–514. https://doi.org/10.1109/ICEES51510.2021.9383710

Lakshman, A., & Malik, P. (2014). Cassandra - A Decentralized Structured Storage System. *Dancing Times*, *105*(1252), 43. https://doi.org/10.1145/1773912.1773922

Lennon, J. (2009). *CouchDB Beginning*.

Li, W. J., Yen, C., Lin, Y. S., Tung, S. C., & Huang, S. M. (2018). JustIoT Internet of Things based on the Firebase real-time database. *Proceedings - 2018 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering, SMILE 2018*, *2018-January*, 43–47. https://doi.org/10.1109/SMILE.2018.8353979

Liu, X., Lftikhar, N., & Xie, X. (2014). Survey of real-time processing systems for big data. *ACM International Conference Proceeding Series*, 356–361. https://doi.org/10.1145/2628194.2628251

Lv, Z., Chirivella, J., & Gagliardo, P. (2016). Bigdata oriented multimedia mobile health applications. *Journal of Medical Systems*, *40*(5). https://doi.org/10.1007/s10916-016-0475-8

Lv, Z., Li, X., Zhang, B., Wang, W., Zhu, Y., Hu, J., & Feng, S. (2016). Managing Big City Information Based on WebVRGIS. *IEEE Access*, *4*, 407–415. https://doi.org/10.1109/ACCESS.2016.2517076

Lv, Z., Song, H., Basanta-Val, P., Steed, A., & Jo, M. (2017). Next-Generation Big Data Analytics: State of the Art, Challenges, and Future Research Topics. *IEEE Transactions on Industrial Informatics*, *13*(4), 1891–1899. https://doi.org/10.1109/TII.2017.2650204

Miler, M., Medak, D., & Odobasic, D. (2011). *Two-Tier Architecture for Web Mapping with NoSQL Database CouchDB*. 62–71. https://www.researchgate.net/publication/236951067

MongoDB. (2022). https://www.mongodb.com/

Moroney, L. (2017a). The Definitive Guide to Firebase. In *The Definitive Guide to Firebase*. Apress. https://doi.org/10.1007/978-1-4842-2943-9

Moroney, L. (2017b). The Definitive Guide to Firebase. In *The Definitive Guide to Firebase*. https://doi.org/10.1007/978-1-4842-2943-9

Nambiar, S., Kalambur, S., & Sitaram, D. (2020). Modeling Access Control on Streaming Data in Apache Storm. *Procedia Computer Science*, *171*, 2734–2739. https://doi.org/10.1016/j.procs.2020.04.297

Nasiri, H., Nasehi, S., & Goudarzi, M. (2019). Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0215-2

Nasr, K. (2021). Comparison of Popular Data Processing Systems KTH Thesis Report. *Degree Project in Computer Science and Engineering*, 76. https://www.diva-portal.org/smash/record.jsf?dswid=6172&pid=diva2%3A1547503

Oussous, A., Benjelloun, F. Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 30, Issue 4, pp. 431–448). King Saud bin Abdulaziz University. https://doi.org/10.1016/j.jksuci.2017.06.001

Philip Chen, C. L., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314–347. https://doi.org/10.1016/j.ins.2014.01.015

Redis. (2022). https://redis.io/

Riak. (2022). https://riak.com/

Ryan, J. (2019). *Big Data Velocity in Plain English*. https://www.voltdb.com/wp-content/uploads/2018/02/VoltDB_BigData_eBook_Feb2018-v2.pdf

Saloot, M. A., & Pham, D. N. (2021). Real-time Text Stream Processing: A Dynamic and Distributed NLP Pipeline. *ACM International Conference Proceeding Series*, 575–584. https://doi.org/10.1145/3459104.3459198

Saranya, K., Chellammal, S., & Chelliah, P. R. (2020). *Ontology-Based Information Retrieval for Healthcare Systems*.

Schram, A., & Anderson, K. M. (2012). *MySQL to NoSQL*. 191. https://doi.org/10.1145/2384716.2384773

Singh, V. K., Taram, M., Agrawal, V., & Baghel, B. S. (2018). A Literature Review on Hadoop Ecosystem and Various Techniques of Big Data Optimization. In *Lecture Notes in Networks and Systems* (Vol. 38, pp. 231–240). Springer. https://doi.org/10.1007/978-981-10-8360-0_22

Splunk. (2022). https://www.splunk.com/

Sudhakar Yadav, N., Eswara Reddy, B., & Srinivasa, K. G. (2018). Cloud-Based Healthcare Monitoring System Using Storm and Kafka. In *Towards Extensible and Adaptable Methods in Computing* (pp. 99–106). Springer Singapore. https://doi.org/10.1007/978-981-13-2348-5_8

Sun, Z., Han, L., Huang, W., Wang, X., Zeng, X., Wang, M., & Yan, H. (2015). Recommender systems based on social networks. *Journal of Systems and Software*, *99*, 109–119.

Syed, D., Zainab, A., Ghrayeb, A., Refaat, S. S., Abu-Rub, H., & Bouhali, O. (2021). Smart Grid Big Data Analytics: Survey of Technologies, Techniques, and Applications. *IEEE Access*, *9*, 59564–59585. https://doi.org/10.1109/ACCESS.2020.3041178

Tang, L., Li, J., Du, H., Li, L., Wu, J., & Wang, S. (2022). Big Data in Forecasting Research: A Literature Review. *Big Data Research*, *27*, 100289. https://doi.org/10.1016/j.bdr.2021.100289

Verma, S., Kawamoto, Y., Fadlullah, Z. M., Nishiyama, H., & Kato, N. (2017). A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues. In *IEEE Communications Surveys and Tutorials* (Vol. 19, Issue 3, pp. 1457–1477). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/COMST.2017.2694469

Vohra, D. (2016). Practical Hadoop Ecosystem. In *Practical Hadoop Ecosystem*. Apress. https://doi.org/10.1007/978-1-4842-2199-0

Xie, L., Zhou, W., & Li, Y. (2016). Application of improved recommendation system based on spark platform in big data analysis. *Cybernetics and Information Technologies*, *16*(Specialissue6), 245–255. https://doi.org/10.1515/cait-2016-0092

Yang, J., Wang, H., Lv, Z., Wei, W., Song, H., Erol-Kantarci, M., Kantarci, B., & He, S. (2017). Multimedia recommendation and transmission system based on cloud platform. *Future Generation Computer Systems*, *70*, 94–103. https://doi.org/10.1016/j.future.2016.06.015

Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B., & Vasilakos, A. v. (2016). Big data: From beginning to future. In *International Journal of Information Management* (Vol. 36, Issue 6, pp. 1231–1247). Elsevier Ltd. https://doi.org/10.1016/j.ijinfomgt.2016.07.009

Zheng, Z., Wang, P., Liu, J., & Sun, S. (2015). Real-time big data processing framework: Challenges and solutions. *Applied Mathematics and Information Sciences*, *9*(6), 3169–3190. https://doi.org/10.12785/amis/090646