Computing and Informatics, Vol. 40, 2021, 489–521, doi: 10.31577/cai_2021_3_489

BIG DATA STORAGE TOOLS USING NOSQL DATABASES AND THEIR APPLICATIONS IN VARIOUS DOMAINS: A SYSTEMATIC REVIEW

Amen Faridoon

University College Dublin, Dublin, Ireland
e-mail: amen.faridoon@ucdconnect.ie

Muhammad Imran

National Centre for Physics, Islamabad, Pakistan
&
CERN, Geneva, Switzerland
e-mail: muhammad.imran@cern.ch

Abstract. Over the past few years, data has been growing significantly due to the advent of new connected devices, availability of bandwidth, and the emergence of new applications which utilize cloud computing infrastructure in the data centers. This increased amount of data faces many problems in terms of storage, transmission, management, and processing, etc. Therefore, the term big data has gained significant attention from researchers in recent years. The rapidly growing quantity, velocity, and variety of data require more probable and logical tools for its storage. For this purpose, the industry is highly emphasizing the development of more viable tools for the storage of big data. The traditional big data storage tools are unsuccessful in storing an enormous amount of data. Hence, the structural modifications of management mechanisms of conventional storage systems such as SQL databases to NoSQL databases technology are necessary to cope up with drastically increasing requirements of big data storage. The primary objective of this paper is to concentrate exclusively on designing a road map for NoSQL big data storage technologies, evaluate current evidence, research progresses in NoSQL data storage systems and their applications in various domains. We conducted a systematic literature review (SLR) of various studies published in recent years. We propose a framework to classify selected articles on the basis of various factors such as motivations behind big data storage, NoSQL techniques used for storing big data, and significant ap-

plications of big data in different domains. Furthermore, we also discuss research issues and define an outline for future research in the big data storage domain for NoSQL databases.

Keywords: Big data, storage tools, NoSQL databases, systematic literature review

1 INTRODUCTION

Over the past few years, big data has gained significant attention of the researchers and industrial experts as world has faced challenges related to big data storage, transmission, management, processing, analysis, visualization, integration, architecture, security, quality and privacy. Big data is an abstract concept. People still have different opinions on the term "big data" but data scientists and experts explain it by five main characteristics which are called 5Vs [58]. These are volume, variety, velocity, veracity and value. The volume represents the quantity of data [45]. Many organizations already have tremendous quantity of archive data but they are not able to efficiently store and process it. Processing of a large amount of data is the main attraction of the big data analytics [36]. The variety refers to the format of the data [45]. The dataset format is divided into three main categories, i.e. structured, semi-structured and unstructured data. Various sources generate data in multiple formats, e.g. videos, audios, documents, comments, logs, tabular form and others. The velocity defines the increase in speed of data generation and processing [45]. Social media platforms and real time applications are some good examples that illustrate data generation with fast speed. The veracity covers the quality or accuracy of the data [5]. While dealing with the large quantity, velocity and diversity of the data, it is impossible that all of the data is 100 % correct rather there will be noisy data. The value is the very necessary aspect of the big data. The value basically states the worth of the data being extracted or extracting meaningful insight from the data. Having access to the big data is good but at the same time it would be useless if we are not able to turn it into the value. [5]

1.1 Challenges of Traditional Systems with Big Data

Traditional systems are not capable to store and process big data efficiently. This is because the traditional storage systems were not designed for such data. Society faces many problems with the traditional storage systems, some of them are

1. schema-on-write,
2. cost of storage,
3. cost of proprietary hardware,
4. complexity,

5. heterogeneous data,
6. causation and
7. bringing data to the programs.

Moreover, in order to reduce the fraction of the cost for traditional shared storage, we have to reduce the size of the data. As a result, after large volumes of the data thrown out causes reduction in the data to be analyzed which ultimately decreases accuracy and confidence of the results. Hence, most of the world has learned that the traditional storage systems are not upright for storing and analyzing huge amount of data.

1.2 Big Data Storage Technologies and Their Features

As the traditional storage systems fail in the era of the big data, the scientists have been looking up for modern systems to overcome this problem [53]. Big data storage is a storage infrastructure that is specifically designed for massive amount of data for storage, processing and retrieval. Big data storage tools are the basic driver for advance analysis and have the capacity to transmute the society. Some of the storage tools are:

1. BigTable,
2. HBase,
3. Cassandra,
4. MongoDB,
5. Neo4j, etc.

These tools meet the demand of storage and allow us to store heterogeneous data. These tools also have certain properties like scalability, access control, fault tolerance, failover recovery, real time query, SQL supported queries, distributed nature and much more [10]. Big data storage infrastructures provide the solutions for the problems that are faced in traditional systems. Solutions of the challenges discussed above in Section 1.1 are:

1. schema-on-read,
2. reducing the storage cost,
3. commodity hardware which overcomes the cost of proprietary hardware,
4. simplicity,
5. allow unstructured and semi-structured data to be stored and processed,
6. correlation and
7. bringing programs to the data.

1.3 Applications Using Big Data Storage Technologies

Many sources that include business applications, public web, social media, machine log data, transactions, sensor data and some real time applications generate big volume of the data continuously [13, 1]. Big data storage technologies cross the barriers and are used to handle the problems of big data. These technologies are not only important in IT-based sector but also have particular importance in non-IT-based sector like energy, health and finance etc. International data corporation (IDC) predicts that total amount of digital data created worldwide will grow from approximately 44 zettabytes in 2020 to 180 zettabytes in 2025 [55] due to the growing number of smart devices, sensors and availability of bandwidth [18]. Moreover, in one day, internet of things (IoT) and the use of connected devices generate huge amount of data. Approximately 500 million tweets and 294 billion emails are sent in a day. Facebook creates 4 petabytes of data. The 5 billion searches are made and 65 billion messages are sent on WhatsApp. The 4 petabytes of data are created from each connected car. In addition, weather channels receive 18 055 556 forecast requests. Venmo processed 51 892 peer-to-peer transactions. Uber riders take 45 788 trips and there are 600 new page edits to Wikipedia. These all happens in one day. The aggregate of space need to save one second video of a high quality is two thousand (2 000) times more than the space needed to save a plain text of page. Some of the predictions made by IDC regarding 2025 termed as IDC Data Age 2025 predictions are shown in Table 1.

Year	Numbers	Predictions	Organization
2025	>150 billion	More than 150 billion devices will be connected across the globe.	IDC Data Age 2025
2025	>6 billion	Consumers will interact with data every day.	IDC Data Age 2025
2025	90 zettabytes	In 2025 IoT devices will generate more than 90 zettabytes of data.	IDC Data Age 2025
2025	30 %	Comparing to 15 % in 2017 nearly 30 % of all data created will be real-time.	IDC Data Age 2025
2020	90 %	Large enterprises will generate revenue from data as a service.	IDC FutureScape: Worldwide IT Industry 2018 Predictions

Table 1. IDC Data Age 2025 predictions

1.4 Existing Issues of Big Data Storage Tools

Although NoSQL data tools have sorted out the issues of big data comprehensively but the velocity is the one area that still requires improvements. This is because of the ever increasing number of big data sources and heterogeneous data created from these sources [13]. Security and privacy are also the well-recognized challenges in the big data storage as the data is stored in the clusters of the data centers not in the users' personal devices. The compatibility and updating data are the requirements that big data storage tools must also fulfill [42]. Scalability is also a major challege which arises due to rapid change in the growth of data and this is handled by adding more storage to an existing node. Furthermore, the data consistency, single server storing of data and partitioning are the most imperative research challenges.

1.5 Our Contributions

Systematic literature review (SLR) uses systematic methods to collect secondary data, critically appraise research studies, and synthesize findings qualitatively or quantitatively. In this paper, we perform SLR on NoSQL big data storage tools and their applications in various domains. The primary objective of this SLR is to concentrate exclusively on designing a road map for NoSQL big data storage technologies, evaluate current evidences, research progresses in big data storage systems and their applications in various domains. We selected various publications from 2015 to 2020 on the basis of this SLR and classify these chosen articles on the basis of factors involved in the movement of SQL-to-NoSQL, frameworks used for storing massive data and significant applications in different industries. Furthermore, we also highlight research gaps and define outline for future research.

In summary, the main objectives of this SLR are following:

- To assist data scientist and researchers to understand and choose the storage framework that effectively suits to their requirements,
- To present the significance and applications of the NoSQL big data storage technologies to particular domain,
- To highlight issues or challenges of the NoSQL database systems,
- To discuss directions for future research in the big data storage.

This SLR entirely depends on framed questions, related studies, analyzing their findings, and gives a brief evidence through clear methodology. In this paper, we categorize the storage tools according to data model such as key-value, columnar, document oriented and graph stores. For evaluation, features or properties of the storage systems are being used. In this paper, we also highlight the use cases of these tools in different domains.

1.6 Needs for Conducting SLR

The need for an SLR entails to create a technology roadmap, analyzing current publications and the research progression in the NoSQL big data storage systems. It exclusively focuses on classification of storage technologies and their applications in different domains. To demonstrate that a similar review has not been already reported, we search the Compendex, IEEE Xplore, ACM, ScienceDirect, SpringerLink and Google Scholar digital libraries with the help of search string shown in Figure 1.

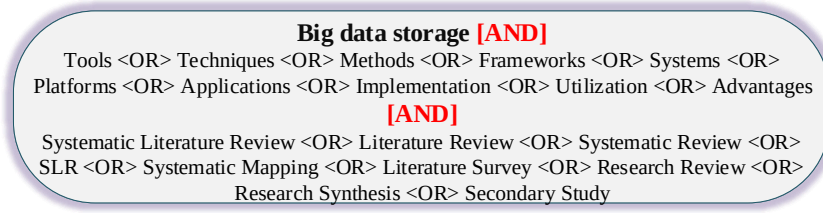


Figure 1. Search string for searching the similar SLRs

After searching the digital libraries of above mentioned databases by means of search string, we come across that none of the subsidiary studies was related to any of the research questions discussed in Section 2.

The remainder of the paper is organized as follows: Section 2 describes the investigation methodology, research questions and the extent to which we stretched our search for the studies. Section 3 provides answer to our research questions that we raised and focus on the results of our research methodology. Finally, we conclude in Section 4.

2 RESEARCH METHODOLOGY

SLR eliminates bias as compared to unstructured review process and also follows an extremely thorough and accurate sequence of steps to search literature. The primary aim of this SLR is to structure the existing literature in the domain of NoSQL big data storage systems and their applications. To conduct this SLR we adopted the guidelines presented in the study [8] with a three-step review process which includes planning, conducting and documenting, as shown in Figure 2. For the SLR to be effective – we obtain, analyze and document the outcome and evaluate our review protocols. The planning and conducting phases of the systematic literature review procedure are described below.

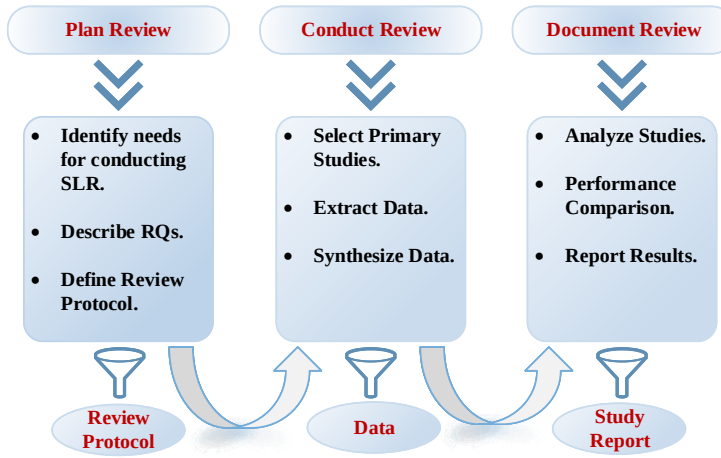


Figure 2. Outline of our research methodology

2.1 Review Plan

An essential element in conducting a systematic literature review is to establish a protocol for the study during the planning phase. Planning phase focuses the underlying principles for the recognition of the needs for a methodical review and results in a review protocol as follows.

Step 1: Characterization of review demands. We specify the research questions in Table 2 that provide the foundations for defining and assessing the review protocol. The research questions depend on our motivation to identify evidence about the distinct storage frameworks and their implementations used by the authors in the big data storage.

We describe the overall objectives and scope of our investigative study by showing them in Figure 3. We consider PICOC (population, intervention, comparison, outcome and context) criteria for this purpose [38].

Step 2: Analyze and specify review protocol. Based on the goals, we define the research questions and the scope of the review that lead us to design the search string for the extraction of the literature. Before the execution of the review protocol, we evaluated it by external experts. Therefore, we invited specialists/professionals to provide feedback who have capabilities for conducting and designing the SLRs in our area of investigation. Their assessment is reflected in the refined protocol.

	Research Questions	Objectives
RQ1	What are the main motivations behind big data storage?	We are interested to know what are main reasons that traditional storage systems failed in the era of big data.
RQ2	What are the existing NoSQL technologies and their key features used in big data storage?	The aim is to investigate and categorize the storage technologies according to database structure.
RQ3	What are the applications of NoSQL big data storage models in various domains?	The aim is to investigate the socio-economic influences of large scale storage systems of data.
RQ4	What are the current research problems and what should be the future research agenda in big data storage?	The main objective is to disclose the research flaws which need to be addressed and potential future directions in this area.

Table 2. Research questions mapped by objectives

Criteria	RQ1	RQ2	RQ3	RQ4
Population or problem	Practical motivation.	Storage technologies and their key features.	Application of storage technologies in various domain.	Research issues and future direction.
Intervention or Exposure	Internal/External validation; Extracting data and synthesis.			
Comparison	Perform comparison between applications of storage tools and evaluate the storage tools on the basic of their key features.			
Outcome	Categorize big data storage technologies. Grouping their applications. Hypotheses for future research.			
Context	A systematic investigation with an exclusive focus on storage technologies and their implementation in different areas.			

Figure 3. Define objective and scope of SLR via PICOC criteria

2.2 Review Conduction

The second phase of research methodology is the review conduction which is initiated by choosing the literature and its outcome is the extraction of data and synthesized information. This phase is further divided into three steps that are:

1. selection of primary research articles,
2. data extraction, and
3. synthesis of the results.

Step 1: Selection of primary research articles. By considering the research questions described in Table 2 and taking the guidelines from study [8] we developed a search string. After applying the search string to the digital libraries of five databases, i.e. IEEE Xplore, ScienceDirect, SpringerLink, ACM and Google Scholar,

we obtained 1 900 articles from 2015 to 2020. The search string and the results from various databases are shown in Figure 4.

		NAME	RESULTS
Storage	Migration <OR> Evaluation <OR> Switching <OR> Traditional Systems <OR> Transformation <OR> Modernization <OR> Relational Systems <OR> SQL-to-NoSQL **<OR>** Technologies <OR> Techniques <OR> Tools <OR> Platforms <OR> Frameworks <OR> Methods <OR> Approaches <OR> NoSQL Stores **<OR>** Applications <OR> Implementation <OR> Utilization <OR> Advantages <OR> Employment **<OR>** Issues <OR> Problems <OR> Difficulties <OR> Challenges **[AND]** Large Data <OR> Massive Data <OR> Macro Data <OR> Large Volume of Data <OR> Large Amount of Data <OR> Lots of Data	IEEE	552
		Springer	448
		Science Direct	138
Big Data		ACM	220
		Google Scholar	542
		Total	1900

Figure 4. Composition of search strings and search results

Initial Selection. This step comprises of inspecting the titles and abstracts of primary articles. In initial selection stage, we came across 136 papers. However further screening of these 136 papers was done against the inclusion/exclusion criteria which is presented in Table 3.

	Inclusion		Exclusion
I1	Scientific peer-reviewed papers are included.	E1	Studies that do not explicitly discuss storage technologies and their features.
I2	Studies that discuss technologies and their key features used in data storage.	E2	Studies that conduct survey on storage technologies.
I3	Studies that conduct experiment on storage technologies.	E3	Editorials, abstracts, courses and papers less than 5 pages.
I4	Studies that mention the issues or problems of big data storage technologies.	E4	Non-peer-reviewed articles.
		E5	Non-English manuscripts.
		E6	Thesis and book chapters.

Table 3. Inclusion/exclusion criteria

Final Selection. Final selection depends on our objectives and goals for big data storage areas which consist of storage systems and their key features, employment of storage systems in various domains, issues or challenges related to storage systems and reasons behind the failure of traditional storage systems for big data. After the completion of final selection process, we selected 33 studies.

We only considered articles for which full text is available and the studies having abstract only are discarded. We also exclude the publications related to text books, editorials, courses, non-peer-reviews, articles not written in English and the papers less than five pages.

Qualitative Assessment of Included Studies. After applying the above mentioned criteria, 33 research articles are filtered out from the database contained 136 papers. Then the screening of these 33 articles have been done through the quality assessment criteria presented in Figure 5. We categorize our quality assessment criteria into two portions that are "General characteristics for quality assessment" and "Specific characteristics for quality assessment". According to the general assessment criteria, we comprehensively reviewed the selected research papers, examined their purpose of the research, and observed that the study effectively described their proposed methodology according to their problem statement and presented results are inline with the contributions of the study. However, through the specific quality assessment criteria, we have to examine: do the research papers define the limitations of traditional storage systems over the dimensions of the big data?, do they clearly investigate the NoSQL big data storage technologies and their use cases in various domains?, and does the study mention the research problems of the storage systems?

	General characteristics for quality assessment	**Score**		
		Yes=2	Partially =1	No=0
G1	Is problem statement and research motivation clearly defined?			
G2	Are methodology of the research and its organization comprehensively described?			
G3	Is the research environment of the study clearly explained?			
G4	Are the presented results of the research inline with the contributions of the study?			
	Specific characteristics for quality assessment			
S1	Is the research successfully described the drawbacks of traditional data storage in the dimensions of the big data?			
S2	Are the research effectively investigate the existing big data storage technologies?			
S3	Are the study productively employed the big data storage systems in a particular domain?			
S4	Are the limitations of present research clearly indicated?			

Figure 5. Quality assessment criteria

3 DISCUSSION AND RESULTS

The Table 4 shows the detailed summary of selected 33 research articles. In this section, we present the discussion and findings on these selected papers on the basis of our research questions.

ID	Title	Year	Journal/ Conf.	RQs	Cs.
S1 [63]	Application and research of massive big data storage system based on HBase	2018	IEEE Xplore	RQ2	–
S2 [27]	Distributed storage system for electric power data based on HBase	2018	Big Data Mining and Analytics	RQ3	–
S3 [54]	A new method for time-series big data effective storage	2017	IEEE Access	RQ1, RQ2	12
S4 [28]	MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data	2015	IEEE Sensors Journal	RQ3	74
S5 [11]	An OAIS-based hospital information system on the cloud: Analysis of a NoSQL column-oriented approach	2017	IEEE Journal of Biomedical and Health Informatics	RQ3	16
S6 [62]	Big data storage and management in SaaS applications	2017	Journal of Communications and Information Networks	RQ1, RQ2	4
S7 [22]	Optimization strategy of Hadoop small file storage for big data in healthcare	2016	The Journal of Supercomputing	RQ2, RQ3	24
S8 [26]	An Approach to Security for Unstructured Big Data	2016	The Review of Socionetwork Strategies	RQ4	1
S9 [43]	Privacy preserving data publishing based on sensitivity in context of Big Data using Hive	2018	Journal of Big Data	RQ3, RQ4	–
S10 [59]	A Big Data platform for smart meter data analytics	2019	Computers in Industry	RQ3	2
S11 [16]	HB-File: An efficient and effective high-dimensional big data storage structure based on US-ELM	2017	Neurocomputing	RQ4	3
S12 [46]	Dynamic Preclusion of Encroachment in Hadoop Distributed File System	2015	Procedia Computer Science	RQ4	5
S13 [57]	A new reliability model in replication-based big data storage systems	2017	Journal of Parallel and Distributed Computing	RQ4	9
S14 [3]	Dynamic Merging based Small File Storage (DM-SFS) Architecture for Efficiently Storing Small Size Files in Hadoop	2018	Procedia Computer Science	RQ2, RQ4	1
S15 [35]	BigDimETL with NoSQL Database	2018	Procedia Computer Science	RQ2, RQ3	–
S16 [21]	Compression and Security in MongoDB without affecting Efficiency	2016	ICTCS	RQ2, RQ4	1

S17 [61]	RDBMS, NoSQL, Hadoop: A Performance-Based Empirical Analysis	2016	AMECSE	RQ1, RQ2, RQ3	5
S18 [20]	Performing OLAP over Graph Data: Query Language, Implementation, and a Case Study	2017	BIRTE	RQ2, RQ3	1
S19 [7]	Enabling Scientific Data Storage and Processing on Big-data Systems	2015	IEEE	RQ3	4
S20 [14]	Using the column oriented NoSQL model for implementing big data warehouses	2015	Semantic Scholar	RQ3	28
S21 [56]	Query-oriented Adaptive Indexing Technique for Smart Grid Big Data Analytics	2017	Journal of Signal Processing Systems	RQ3	2
S22 [60]	A Performance-improved and Storage-efficient Secondary Index for Big Data Processing	2017	2017 IEEE Inte. Conf. Smart Cloud	RQ3, RQ4	–
S23 [34]	A Versatile Event-Driven Data Model in HBase Database for Multi-Source Data of Power Grid	2016	2017 IEEE Inte. Conf. Smart Cloud	RQ3, RQ4	2
S24 [44]	A Framework for Migrating Relational Datasets to NoSQL	2015	Procedia Computer Science	RQ1	26
S25 [41]	Comparative Study of SQL and NoSQL Databases	2015	Inte. J. Adv. Res. Comp. Engi. Tech	RQ1, RQ2	9
S26 [49]	SQL Versus NoSQL Movement with Big Data Analytics	2016	Inte. J. Info. Tech. Comp.Sci	RQ1, RQ2	11
S27 [6]	Handling Big Data using NoSQL	2015	29th Inte. Conf. Adv. Info. Net. Appl.	RQ1, RQ2	37
S28 [51]	Aerospike: architecture of a real-time operational DBMS	2016	Proc.VLDB Endowment	RQ2	18
S29 [15]	A scalable generic transaction model scenario for distributed NoSQL databases	2015	Journal of Systems and Software	RQ2	12
S30 [52]	Analysis of various NoSql database	2015	ICGCIoT	RQ2	14
S31 [33]	Statistical analysis of tourist flow in tourist sports based on big data platform and DA-HKRVM algorithms	2020	Personal and Ubiquitous Computing	RQ2, RQ3	1
S32 [39]	Distributed Data Platform for Automotive Industry: A Robust Solution for Tackling Challenges of Big Data in Transportation Science	2019	ConTEL	RQ2, RQ3	1
S33 [30]	Multilevel Object Tracking in Wireless Multimedia Sensor Networks for Surveillance Applications Using Graph-Based Big Data	2019	IEEE Access	RQ2, RQ3	3

Table 4. List of selected studies

3.1 What Are the Main Motivations Behind Big Data Storage? (RQ1)

Our classification of research articles has allowed to counter our first research question (RQ1). We identify main reasons behind big data storage focused by the researchers in their studies. On the ground of literature, we note that relational databases have lost their popularity because of well-structured nature. The motivation and need behind using the big data storage technologies is that the relational, well-structured, well-schema [S3, S6, S17] databases could not develop as rapid as big data. The large volume of data comes with their own challenges [50] such as real-time processing, fast recovery, fault tolerance and complex structure of data is not fulfilling expectations and needs for heterogeneous data [S3, S17, S26]. In addition, the schema of relational and structured databases does not assist the recurrent changes. Some of the well-known network data repositories are Google, Twitter, Amazon and Facebook. There, management and dynamic scalability requirements exceed the capabilities of relational databases [40]. Thus, frequently growing and developing data needs a better and satisfactory solution. In order to fulfill the needs of big data, NoSQL databases have emerged which not only overcome the problems of relational datasets, rather the also have become mainly adopted frameworks for storing large scale data. Unlike the traditional databases, these frameworks have capacity and ability to deal with multiple users interacting with big data simultaneously. They also provide assurance for some distinctive characters over relational datasets like distributed scalability, availability, fault tolerance, consistency, data replication, parallel data processing, flexibility, multiple servers, non-relational distributed data models, efficient, high performance, cost saving and secondary indexing.

Based on data synthesis, we identify four primary factors which are scalability (32 %), availability (21 %), schema less (21 %) and data replication (16 %) of the studies [S3, S6, S17, S24, S25, S26], that clearly mentioned the SQL verses NoSQL movement, as shown in Figure 6. We also perform comparison between relational database systems and NoSQL big data storage models and this is shown in Table 5. This analysis clearly demonstrates that big data storage technologies have more worth than relational database systems.

3.2 What Are the Existing NoSQL Technologies and Their Key Features Used in Big Data Storage? (RQ2)

We answer RQ2 by considering the NoSQL storage technologies from selected research articles. The remaining part regarding this question involves remarkable features of these storage technologies. Big data storage tools are referred to as the storage tools that in measure particularly address the quantity, speed, or verity of challenges and do not drop under the heading of traditional storage systems. This does not specify that traditional storage systems do not solve these problems but other storage tools like NoSQL databases (column oriented store, Key-value store, document oriented store and graph oriented store) [S17, S25, S26] are often more methodical and less expansive. These technologies offer scalable storage solutions

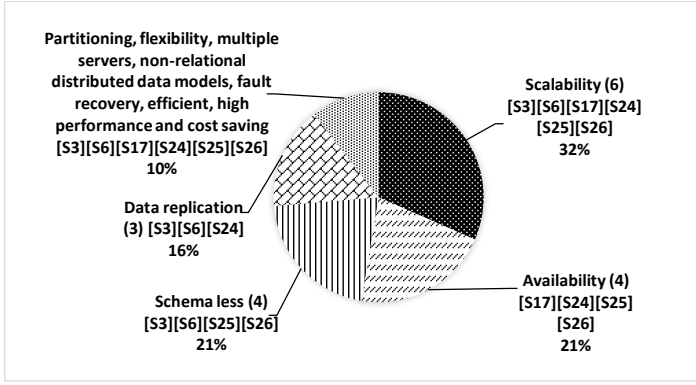


Figure 6. Percentage of studies with respect to various drivers behind big data storage

	Traditional Database Systems	NoSQL Storage Systems
Strengths	Store structured data. Vertical scalability. Extendable processing on a server. Particular schema. Particular DML (Data Manipulation Languages).	Store un-structured, semi-structured and structured data. Scalability. Extendable commodity servers. Parallel computing applications. High availability and fault tolerance. Reliability. Simultaneous accessibility and consistency.
Weaknesses	Bottleneck performance and delays in processing. With the growth of data, chances of occurring deadlock increases. Limited storage capacity. Difficulty in join operations for multidimensional data.	Due to scalability and performance not compliance with ACID (atomicity, consistency, isolation, durability).
Opportunities	Traditional data storage systems support complex queries. Built-in deployment. Atomicity in complex database transactions.	Simplicity in complex storage structures. Enhance response time for query.
Threats	With the dynamic growth of the data large volume for storage is required. Complex and Schema less data structures. Real-time processing and maintaining consistency for large number of storage servers are the main threats for relational storage systems.	Deployment of big data storage systems. Small files in large numbers is one more threat.

Table 5. SWOT analysis of traditional and big data storage architectures

for continuously growing big data with further improved data structures quality wise and support fault tolerance. After the keen survey of literature we categorize big data storage tools according to the classification of NoSQL data models. A brief summary and comparison of these tools are also presented in Table 6.

NoSQL databases. NoSQL database management systems are the most imperative family of big data storage technologies. When the storing and processing of large data is a primary requirement then the NoSQL is the preferred choice. NoSQL databases introduce data models from outside the relational world. These models are flexible in nature, provide horizontal scalability, schema-less and aim to manage large amount of data. NoSQL databases can be divided into four major categories based on their data model. [S25, S27, S30]

3.2.1 Key-Value Stores

This is the simplest NoSQL data store model. Key-value data models allow data to store in a schema-less way [S17, S25, S26, S27, S30]. Because of no schema, it is not compulsory that the data objects share the same structure. It may be completely structure or un-structure and can be assessed by a single key. In key-value store model, "Key" represents unique value to access row and value contain the information which corresponds to that key. Key-value model shows flexibility to add more records easily and to consume less memory storage because of different representations of values in records. Aerospike, Riak and Redis and many others are the examples of key-value database model.

Aerospike. Aerospike is the name of eponymous company that produces it. It is the open-source, in-memory, first flash-optimized key-value data storage software and is more suitable for storing real-time data[51]. It is written in C language and operates in three layers namely flash optimize data layer, distribution layer and cluster aware client layer. To ensure the consistency, the distribution layer is replicated across data centers. Whereas, client layer is responsible for managing communication in the server node and it is also used to track the cluster configuration in the database. The highlighted goals behind the development of Aerospike are to design a scalable and flexible framework for web applications and support reliability and consistency like a traditional database. [S27, S28, S29]

Riak. Riak [48] is also a NoSQL database storage system that provides high availability, fault tolerance, operational simplicity and scalability in a very low cast. In addition, it can be used to store data in memory, disk or both. It uses the term bucket and key for interaction with objects. The key is utilized to store the values whereas the bucket is for setting the bucket's properties. The Riak distributes the data throughout the cluster by computing the binary hash of each bucket/key pair and maps the calculated value to a location on an ordered ring of all such values. In multi-datacenters replication, one cluster acts as a primary

cluster and is responsible for handling requests from one or more secondary clusters. If the primary cluster goes down, then secondary cluster can take place of it. Moreover, the objective of Riak developers is to provide high availability of diverse data to applications. [S25, S26, S27, S30]

Redis. Original developer of the Redis is Salvatore Sanfilippo. Linux platform is selected for the development and testing of Redis [17]. It can provide powerful properties such as fast access to whole data resides in the memory, built-in persistency. Its distinctive feature is to support multiple datatypes. Thus, Redis is the most suitable option for heterogeneity in servers and application, and where in-memory data is the requirement. It is designed for efficiently supporting query operations and replication in a master-slave environment. [S17, S30]

3.2.2 Columnar Stores

Google's BigTable is the motivation behind the column family stores. The basic architecture of columnar store data model is rows and columns, and any number of key-value pairs can be stored within rows. Rows and columns both are split over multiple nodes to achieve scalability. Columns can be grouped to column families while rows are fragmented over nodes according to primary key. Columnar store systems provide efficient data compression and partition, and particularly perform well with aggregation queries such as sum, count, average, etc. The most important advantage of columnar store models is the scalability. They are well suited for parallel processing where data is distributed over thousands of clusters and are extremely fast in data loading. HBase, Cassandra, Hypertable, BigTable etc. are the examples of columnar stores. [S17, S25, S26, S27, S29, S30]

BigTable. BigTable is a compressed, high performance and proprietary data storage system developed by Google Inc [12]. The prominent properties of BigTable are flexibility, adoptability, reliability, high performance storage for a structured large-scaled data distributed over the commodity servers, and applicable storage and manage petabyte of data on thousands of machines. It is design for the distribution of highly scalable and structured data. BigTable is applicable to store large amount of structure data at google, web pages and many other google products. [S17, S26, S30]

HBase. HBase [29] is a column oriented, non-relational and distributed database model written in Java which is capable of managing structure and un-structure data. It was developed by Apache. The objective behind its designing is to handle big data storage needs in Apache project [19]. HBase runs on the top of Hadoop distributed file system (HDFS). It provides scalability, distribution, fault recovery and random read/write access to stored data. HBase architecture is composed of at least one master server responsible for management and assign regions to region servers and several slave servers to store data. The most prominent feature of HBase is support to read-intensive transactions. Motivation behind the development of HBase is to provide random, consistent and real-time

access to scalable BigTable with intensive read and write operations. LinkedIn is the use case of HBase. [S1, S17, S25, S26, S30, S31]

Hypertable. Hypertable [29] is an open source software inspired by the design of Google BigTable. It is written in C++ and runs on the top of distributed file system such as HDFS, GFS and CloudStore. It provides good support to consistency of stored data in terms of tables, and divides these tables to acquire scalability and distribution. The importance of Hypertable is that when the master becomes fail to respond for a brief time period and it has no effect on client data transfer. It is designed to provide parallel, scalable databases and better query performance for large size data. [S26, S30]

Cassandra. Initially, Cassandra was developed at Facebook to power the inbox search feature. Now it has become an Apache incubator project. Cassandra [23, 32] is a distributed, wide columnar store NoSQL database management system designed to handle large amount of data across many data centers or commodity servers. It has multiple features like scalability, instants storage, improved frequent read and write operation requests, achieve data consistency through periodic updates on replicating sites, and reliability achieves over large-scale systems [2]. However, the most prominent is high availability with no single point of failure, and fault tolerance and reduce latency which are achieved through clustering, partitioning and replication. [S6, S25, S26, S30]

3.2.3 Document Stores

Document oriented stores are one of the main categories of NoSQL database storage systems designed for storing, retrieving and managing document oriented information which is also called semi-structured data. Document stores are schema less and support secondary indexes. They are inherently a sub-class of key-value store but they support more complex data then key-value stores. In contrast with relational database, they store all information for a given object in a single instance in the database while relational databases store information in separate tables define by the programmer. MongoDB, SimpleDB, CouchDB and others are the examples of document oriented databases. [S3, S6, S16, S17, S25, S26, S27, S29, S30]

MongoDB. MongoDB [37] is an open-source, cross-platform document oriented database developed by Mongo Inc. MongoDB uses JSON like documents with the characteristics of MySQL. MongoDB stores documents in the form of binary representation known as BSON. When the primary server is failed, multiple replicas are considered to achieve the availability of data. Main features provided by MongoDB are adhoc queries (support field, range query and regular expression searches), indexing (primary and secondary indices are used to indexing the document), through the replica sets. Some of the MySQL properties acquired by MongoDB with slight modifications are high availability, load balancing, file storage, aggregation, dynamic updates etc. Aim of the MongoDB is

to provide relational data model facilities to document-oriented databases. [S3, S6, S16, S17, S25, S26, S27, S30, S32]

SimpleDB. SimpleDB is an open-source, distributed, document oriented database which is written in Erlang programming language. It is developed by Amazon Inc [47]. High availability, durability, data model flexibility and automatic indexing are most noticeable features of SimpleDB. Along with features, SimpleDB also have some limitations as compared to the consistency of other database management systems. SimpleDB provide eventual consistency also known as optimistic replication. To overcome the problem of eventual consistency, two new operations are introduced in 2010 that are conditional put and delete, and consistent read. Developers goal is to offer geographic replication for data availability and durability. It is used for complex queries, logs and online games. [S27, S29, S30]

CouchDB. CouchDB is an open source, NoSQL database software having a scalable architecture [4]. It is implemented in a concurrent oriented language Erlang. The main goals of developing CouchDB is to perform data operations and management on the web. CouchDB store any kind of data as documents, and each document has its own self-contained schema. Bi-directional replication and off-line operation were the two goals in the developer's mind at the time of designing the CouchDB. ACID properties of database, built for offline, document storage, eventual consistency, map/reduce views and indexes are the key features of CouchDB. [S17, S25, S26, S30]

3.2.4 Graph-Oriented Stores

Graph database stores are the part of the NoSQL databases, created to overcome the limitations of relational databases and are superlative choice to store data along with relations. The key concept behind the system is graph, that narrates the data items to a collection of nodes and edges. However, nodes represent the data items and edges represent the relationship between the nodes. Relationships between the data items allow stored data to linked together directly and most of the time data is retrieved with one operation. Graph search specific portion according to the execution of query, it does not search irrelevant data. Therefore, it improves the performance of the graph databases systems. Neo4j, InfiniteGraph, HyperGraphDB [31] and many more are the example of graph oriented stores. [S18, S25, S26, S27, S30]

Neo4j. Neo4j is an open source [9], graph database management system introduced by Neo4j, Inc. Neo4j is the effective replacement of relational databases. Scalability, concurrency, transaction load and read request loads are the highlighted properties of Neo4j system. It not only performs improvement on its older version but also competes other graph databases. With the help of buffering and without blocking, Neo4j supports to write-intensive transactions. [S18, S25, S26, S30, S33]

InfiniteGraph. Infinite [24] is a commercial, distributed graph database which is implemented in Java. InfiniteGraph is useful to find hidden relationships in highly connected big data sets. It can store growing data with some schema to further perform normalization and other presentation operations. To achieve scalability, InfiniteGraph implements graph model (Labeled directed multi-graph) technique. While, other key features of InfiniteGraph are concurrency, distribution, multi-threaded, cloud enabled, parallel query support, fully ACID, and having some schema. Easy traversal of complex relationships and provision of support for complex queries over high value data are the main goals for the development of InfiniteGraph. [S27, S30]

HyperGraphDB. HyperGraphDB [25] is an open source data storage mechanism designed for the knowledge management, artificial intelligence and semantic web projects. This graph database provides storage mechanism for random data and also support data mapping between host language and storage. By providing the customizable indexing feature, efficient graph traversal and data retrieval are achieved. However, for storing the graph information Key-value mechanism is used like nodes and edges of a graph are used as a key. In distinction to master-slave storage systems like HBase, Hypertable, Redis HyperGraphDB implements a peer-to-peer data distribution mechanism. [S26, S27, S30]

Summary of Storage Tools. The summary of storage tools is described in Table 6. The table highlights summary with respect to the preferred and non preferred areas, systems, vendors, licence, goals and applications of NoSQL databases for big data storage technologies.

The preferred area of key-value storage is user profile maintenance having no specific schema. It is also suitable for managing a large amount of small-sized data records of web applications like managing session information for online shopping and online games, etc. Moreover, searching for more attributes rather than one from records is the appropriate use case for key-value storage. However, frequent updates, query specific data values, and establish relationships of data values with each other are not suitable areas for key-value data models. Perform analysis to aggregate homogeneous data items is the most common application area for columnar stores. Furthermore, e-library, patient record management, customer data analysis, online attractive applications, write-intensive processing applications, and others are the use cases for column-oriented databases. Despite that, we should avoid column-store systems where the applications need complex queries. The most common applications of the document-oriented data model are maintaining social data, analyzing websites, content management, and e-commerce systems. However, this NoSQL model is not preferred where transactions with multiple operations are required. Recommendation systems, social networks, bioinformatics, pattern mining, and semantic web projects are the applications of the graph-oriented data model. Moreover, it is also preferred for location-based networks and real-time search. However, the use of such a store must be avoided where data cannot be modeled as a graph.

Data Model	Preferred Areas	Not Preferred Areas	Systems	Vendor	License	Goals	Applications
Key-value store	User profile maintenance having no specific schema. Section data for users. Shopping cart's data storage.	Need to be queried specific data value. Frequent updates. Establish relationships of data values with each other.	Aerospike	Aerospike, Inc.	Open Source	Designing a scalable and flexible framework for web applications. Support reliability and consistency like a traditional database.	Web applications
			Riak	Basho Technologies	Open Source	Objective of Riak is to provide high availability to applications.	Diverse data
			Redis	Salvatore Sanfilippo	Open Source	Redis is designed for master-slave environment to efficiently support query operations and replication.	Used for small structured data.
Wide-column	Blogging platforms are the use case of wide column. Counter-based and content management systems. Write intensive processing applications	Applications needed complex querying and has varying patterns queries. Avoid column stores systems where the database requirement is not established	HBase	Apache	Open Source	Motivation behind the development of HBase is to provide random, consistent and real-time access to scalable BigTable with read and write operations.	Latency tolerant applications, sparse and versioned data are the main areas of HBase. LinkedIn also use HBase.
			Cassandra	Apache	Open Source	Aim behind the development of Cassandra is to provide distributed, fault tolerance and highly available storage for data and improved access performance through replication and row distribution of data.	Online interactive applications like Facebook, twitter etc.
			Hypertable	Zvents	Open Source	Designed to provide parallel, high-performance, scalable databases, and better querying performance for large size data.	Store and maintain both structured and un-structured data
			BigTable	Google	Proprietary	Design for the distribution of highly scalable, structured data.	Used to store structure large volume data at google, web pages, and many google products.
Document oriented	Content management and e-commerce systems. Blogging and analytics platforms.	Applications needed complex search queries and transactions with multiples operations	MongoDB	MongoDB, Inc.	Open Source	Developed to provide fast and consistent data access from different applications across multiple interfaces. Another goal is to provide relational data models facilities to document oriented databases.	Real-time applications.
			SimpleDB	Amazon	Open Source	Offer geographic replication for data availability and durability.	Used for complex queries, logs and online games.
			CouchDB	Apache	Open Source	For web documents, make available a dynamic and self-contained schema.	Web applications and social data are the focusing areas.

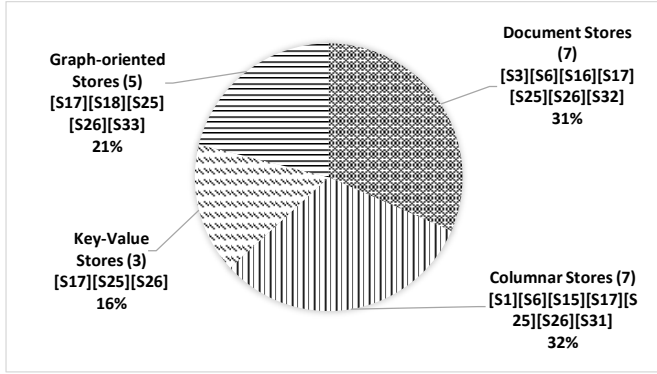
Graph stores	Graph based searches and IT operations are the use cases of graph oriented stores. Fraud detection. Social networks.	Use of such a store must be avoided where data cannot be modeled as a graph	Neo4j	Neo Technology	Open Source	To provide relation-like graph, data relationship manipulation and decision making.	Social networks and recommendations systems.
			Hyper GraphDB	Kobrix Software, Inc.	Open Source	Relational and object oriented data management and memory model for artificial intelligence and semantic web projects are the reasons behind HyperGraphDB.	Bioinformatics, pattern mining and semantic web projects are the applications of HyperGraphDB.
			Infinite Graph	Objectivity, Inc.	Commercial	Easy traversal of complex relationships and provide complex queries over high value data are the main goals.	Preference domains are Social and location-based networks, and real-time search.

Table 6. Summary of storage tools

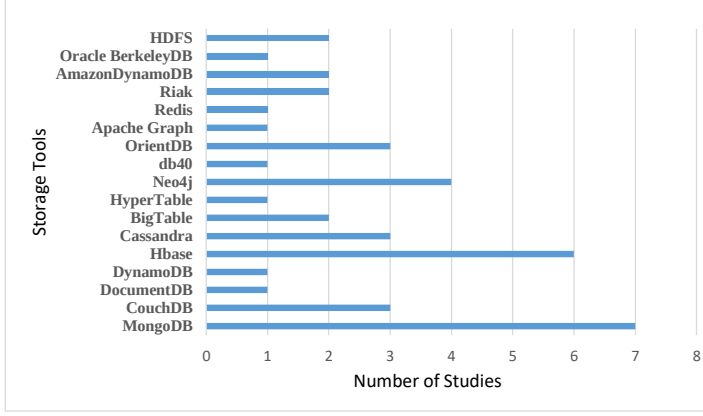
According to the selected studies, we conclude that document and columnar stores are the most frequently used NoSQL databases having the percentage of 31 % and 32 %, respectively, as shown in Figure 7 a). Whereas the graph-oriented and key-value stores are less used database in terms of percentage, i.e. 21 % and 16 %, as compared to document and columnar stores. Many of the researchers used MongoDB and CouchDB document stores with the number of 7 and 3, respectively, in their studies. In columnar stores HBase, Cassandra and BigTable storage tools gain the attention of the researchers for storing big data, as shown in Figure 7 b).

Most Repeated Features of NoSQL Models. Above mentioned big data storage technologies have storage structures to assist scalable resource configuration of big data. Most of the storage systems are developed to ensure availability, consistency, fault tolerance, flexibility, reliability and the durability in general. It can be deduced from Figure 8 that scalability, schema less, calculated performance, low cost, partitioning, data replication, accessibility and sharding are the most repeated features according to the selected studies.

Specific Features of NoSQL Models. NoSQL storage models have specific properties such as scalability, shared-nothing architecture, persistence, partitioning, in-memory, on disk or both memory and disk storage, and rigorous read and write. Figure 9 highlights the specific features of above mentioned NoSQL storage technologies. We can observe that all of the Key-value data models are in-memory, shared-nothing architecture, and scalable rather than the Redis. The Redis provides automatically data partitioning through horizontal scaling whereas, vertical scaling is difficult in it. The Aerospike does not provide a persistence feature while data is stored in memory, however we can persist the data by using persistence memory like disk or device storage. Databases included in the category of wide-column and document-oriented stores support maximum features presented in the



a) Number of studies mentioned NoSQL databases



b) Usage of storage tools in selected publications

Figure 7. NoSQL storage technologies

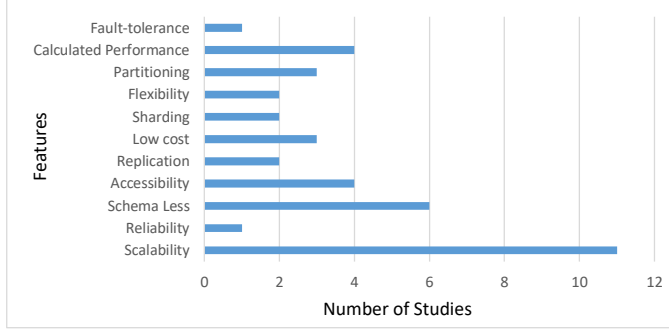


Figure 8. Most repeated features of storage tools

Figure 9. However, SimpleDB only allows persistence data and scalable systems. While, graph-oriented databases like Neo4j, HyperGraphDB, and InfiniteGraph do not back the partitioning and data persistence features.

Storage Systems	Memory Storage	Disk Storage	Intensive Read/Write	Persistence	Partitioning	Shared nothing Architecture	Scalability
Aerospike	✓	x	✓	x	✓	✓	✓
Riak	✓	✓	x	✓	x	✓	✓
Redis	✓	x	✓	✓	✓	✓	x
HBase	x	✓	x	✓	✓	x	✓
Cassandra.	✓	x	✓	x	✓	✓	✓
Hypertable	✓	✓	✓	✓	✓	✓	✓
BigTable	✓	✓	✓	✓	✓	✓	✓
MongoDB	✓	✓	✓	x	✓	✓	✓
SimpleDB	x	x	x	✓	x	x	✓
CouchDB	x	✓	✓	✓	✓	✓	✓
Neo4j	✓	✓	✓	✓	x	✓	✓
HyperGraphDB	✓	✓	-	x	✓	-	✓
InfiniteGraph	-	-	✓	x	✓	✓	✓

Figure 9. Specific features of storage tools

3.3 What Are the Applications of NoSQL Big Data Storage Models in Various Domains? (RQ3)

Big data storage technologies have very significant applications in various domains like real-time big data, time-series data, content management, customer 360 view, mobile applications, fraud detection and others. Many industries are now adopting big data storage technologies like NoSQL databases technologies for critical business applications. These technologies are gradually taking place of relational database technologies to acquire better features like scalability, flexibility, replication, partitioning etc. Some of the applications of big data storage technologies mentioned in under studied articles are discussed below.

Internet of Things (IoT). IoT (the internet of things) states that the control of automated intelligent and inter linked devices command over wide regions through sensors and other computing capabilities. The data produced by IoT is characterized by its continuous growth, unstructured and huge amount. Traditional database technologies are not capable enough to handle such a huge amount of IoT generated data and if you cannot store this heterogeneous data streaming in every second, you would not be able to accomplish any tasks on it. Thus big data storage technologies such as HBase, MongoDB, Cassandra etc. are normally based on distributed file system, database management and data processing technologies, have emerged as a fundamental technology to implement IoT generated data. Selected study [S4] uses MongoDB for storing the

multi-source IoT data sources such as RFID (Radio frequency identification), sensor and GPS (Global positioning systems). In addition, they also devise an effective shared key to maximizing the query speed and horizontally distribute data over data servers.

Healthcare. There is a huge amount of data associated with health sector and it has to be processed and stored. With the progress in health systems, they are continuously moving towards the effective digital solutions. The main objective behind this is to efficiently manage data resources and information associated with health processes. The study [S5] implements OAIS healthcare architecture based on NoSQL column-oriented Cassandra database management system and provides a way to handle such a big amount of HL7 clinical documents in a scalable manner. Moreover authors conduct case study for finding the blood glucose level and assembled results are stored in OAIS system to monitor health condition of patients and to halt deaths. In study [S7], the aim of the authors is to devise a method for the short files storage of genomic and clinical data that will help researchers to execute analytics in healthcare. The given method incorporates the small files of data block and after merging stores these big files so that to reduce the data blocks of HDFS.

Decision Making. We have experienced an immense amount of data on the web. This is because of speedy technological advances with the accessibility of smart devices and social networks like Instagram, Twitter, Facebook, etc. These social sites enable us to make effective decisions. The authors in study [S15] perform ETL (Extract-Transform-Load) operations with HBase to store tweets by using join algorithms. Results highlight the ETL operation execute well with join operation to make effective business decisions. Similarly, the authors in study [S20] perform decision queries on star schema benchmark (SSB) data warehouse and considered HBase columnar NoSQL database for storing purpose.

Electric Power Data. One of the big data storage application is managing enormous electric power data. Electric power systems consist of billions of devices nowadays. These devices generate hundred and thousand of records in a single day. For ensuring the security and maintenance of these power systems, huge amount of data from large number of data sources required to be properly processed and inspected so that a rapid decision could be made in real time. According to study [S2], authors proposed a system through which electric power data can be stored effectively by using HBase. Proposed system is used to monitors a status, and also to perform data migration and fragmentation. The proposed system of study [S10] is capable for storing, quering, visualizing and analyzing large scale smart grid power data sets. Results obtained from this are compared with IBM, MongoDB, Google Cloud and AmpLab provides comparatively easy platform to handle such a big electric power data, with ability of decision making. In studies [S21, S22, S23] researchers devised a unique method to store enormous amount of data by gaining the advantage of HBase. The proposed [S21, S22] systems not only increase the query process but also prevent

space for storage. While authors in study [S23] proposed a data model in which join operation is integrated by using virtual column family.

The number and percentage of selected studies with respect for various domains using NoSQL big data storage models is shown in Figure 10.

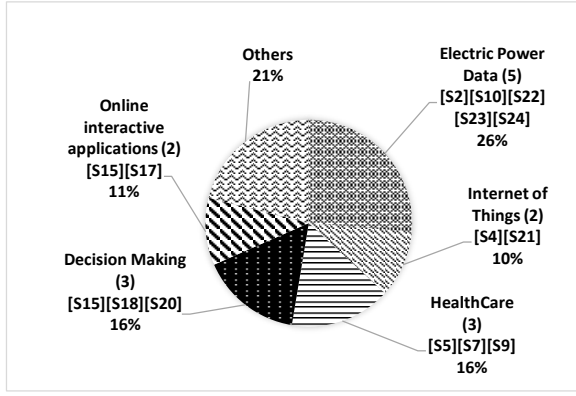


Figure 10. Number of studies contain big data storage applications

3.4 What Are Existing Research Issues and What Should Be the Future Research Agenda in Big Data Storage? (RQ4)

There are many advanced data storage technologies which help us a lot in storing big data but they are not yet perfect platforms. Undoubtedly, the new data storage technologies offer a lot of benefits over conventional data storage technologies, but these technologies are not better enough. There is no ideal platform to be used as a better storage solution. Information extracted from the selected articles help us to answer our research question four (RQ4). Here is a brief look at the existing research issues and distinctions in NoSQL big data storage technologies.

3.4.1 Future Research Challenges

Security. Top level existing research issues include providing security to a stored data. Security of a data is considering as a big challenge on any platform. Big data storage technologies are facing a fairly handsome list of issues regarding security. Many security issues will likely be solved as data storage technologies continue to grow. But for now, security is a distinction in big data storage technologies. Authors in study [S8] provide security suit which contains various algorithms. Providing security to unstructured data, authors take data from Wikipedia and Google search API by taking various data types into account and their sensitivity level. Providing security to Hadoop complex distributed

file system is a challenging task. The main focus of study [S12] is to provide a security model to ensure the variety of secure data operations like insertion, deletion and replication of data over clusters. Likewise, for storing files of small size in HDFS, authors in study [S14] implement the encryption technique known as Twofish to ensure the security of content present in files.

Advantages of MongoDB are that it provides replication, schema-less, supports indexing and many more but it also has some limitations related to security of a data stored in it. So authors in study [S16] played a part in resolving the problem by introducing a middleware encryption before storing the data into database.

Issues relating to security in big data storage systems need to be further investigated into with respect to different tools.

Read Performance. Authors in study [S22, S23] highlighted the issues (join operation, effective indexing and random read) related to HBase. To overcome the limitation of indexing schema authors in study [S22] implement secondary indexing technique to speed up query processing and save the huge storage space. A virtual column family is introduced to resolve the problems of random read and join operation in HBase [S23].

Similar issues relating to performance during fetching data from big data storage systems with respect to different bid data storage tools need to be evaluated.

Data Management. With the escalation of big data, the related data storage industries emphasize more on data management instead of computational management. In recent time, managing a data is a big task. Many data storage technologies have been proposed which help us a lot in storing a growing data and processing resources. However, still more efficient technologies are required for data acquisition, processing, pre-processing, storage and management of big data. The continuing development on big data management focuses mainly on bringing effective solutions that support big data efficiently. Management of growing volume of data is also very significant in this regard beside processing, pre-processing and storage. The main concerns of ongoing development include methods of data clustering, replication and indexing for effective storage exploitation and data retrieving.

Data Consistency. Data consistency is considered as a design goal for big data storage technologies. In distributed systems, consistency and availability have greater impact on each other and one of them is compromised. Data consistency remains a basic task for big data storage technologies. Like, NoSQL databases do not perform ACID transaction, a technique used for ensuring data consistency. In general, data consistency is a major issue in big data storage that needs to be addressed.

Scalability. The term scalability refers to handle and support increasing volume of data in such a way that a prominent optimization in the storage resources

is possible. Scalability is considered as one of the significant design goal for data storage technologies. Existing technologies have better scalability standards over traditional data storage technologies but in many respects, scalability is still a challenge. For instance, some NoSQL databases are not better enough at automating the process of sharding (spreading a database across multiple nodes). Other databases like SQL are also facing the same type of problem.

Single Server Storing of Data. Storing a big data under a single server is not a better decision while considering a nature of data. It is wise to configure a cluster of multiple hardware elements as the distributed storage system.

Frequent Data Update and Schema Change. With the rapidly growing volume of data, the need for increasing the update rate for data is also very high. Changes made in schema is also very communal in case of unstructured data. However, existing storage technologies are better in scalability but requirement to be efficient in data updates and schema is still under consideration.

Partitioning Method. Maintaining acceptable performance in growing size of the database become more complex. So the partitioning is the method to manage busy and large amount of data. Two types of partitioning are offered by the data models that are horizontal and vertical partitioning applied on data based on access patterns. However, during the execution access patterns might be wrong. Thus existing data models identified for big data storage solutions show that partitioning is a critical research challenge.

4 CONCLUSION

Big data is an abstract concept. Experts categorized big data by 5Vs referred to as volume, variety, velocity, veracity and value. As the data is growing continuously and rapidly, so this increased quantity, speed and diverse nature of data require more reliable and logical tools for its storage. The main objective of this survey is to refocus, probe and analyze the futuristic NoSQL big data storage models. We conducted SLR by selecting 33 publications from year 2015 to 2020 on the basis of our defined criteria. The primary and major objective of this SLR is to reconcentrate on the storage tools, mentioned the applications and spot the challenges of storage systems. We categorize our selected publications on the basis of four major questions.

The main concern of our first research question is to highlight the factors that are scalability, availability, schema less, data replication etc. involved in the migration of traditional tools to big data storage systems. According to the selected studies, most of the researchers frequently used document and columnar store 31 % and 32 %, respectively, NoSQL databases. The results clearly show that 14 out of 33

publications mostly used MongoDB, HBase, CouchDB, Cassandra and Neo4J storage tools. Scalability, schema less, calculated performance, partitioning, low cost and accessibility are among the most repeated features of storage tools. Big data storage tools play a consequential role in many fields. But the results gather from our selected publications tells us that smart power grid, healthcare, decision-making, online interactive applications and internet of things are the major domains where their applications are extensively used.

We have recognized that there has been an extraordinary research work done over the years by researchers. However, there are many flaws that still need to be fixed in terms of security, privacy, read performance, data management, data consistency, scalability, single server data storage, frequent update and data partitioning.

REFERENCES

[1] Abouzeid, A.—Bajda-Pawlikowski, K.—Abadi, D.—Silberschatz, A.—Rasin, A.: HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. Proceedings of the VLDB Endowment, Vol. 2, 2009, No. 1, pp. 922–933, doi: 10.14778/1687627.1687731.

[2] Abramova, V.—Bernardino, J.: NoSQL Databases: MongoDB vs Cassandra. Proceedings of the International C* Conference on Computer Science and Software Engineering (C3S2E '13), ACM, 2013, pp. 14–22, doi: 10.1145/2494444.2494447.

[3] Ahad, M. A.—Biswas, R.: Dynamic Merging Based Small File Storage (DM-SFS) Architecture for Efficiently Storing Small Size Files in Hadoop. Procedia Computer Science, Vol. 132, 2018, pp. 1626–1635, doi: 10.1016/j.procs.2018.05.128.

[4] Anderson, J. C.—Lehnardt, J.—Slater, N.: CouchDB: The Definitive Guide: Time to Relax. O'Reilly Media, Inc., 2010.

[5] Ishwarappa—Anuradha, J.: A Brief Introduction on Big Data 5VS Characteristics and Hadoop Technology. Procedia Computer Science, Vol. 48, 2015, pp. 319–324, doi: 10.1016/j.procs.2015.04.188.

[6] Bhogal, J.—Choksi, I.: Handling Big Data Using NoSQL. 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, 2015, pp. 393–398, doi: 10.1109/waina.2015.19.

[7] Biookaghazadeh, S.—Xu, Y.—Zhou, S.—Zhao, M.: Enabling Scientific Data Storage and Processing on Big-Data Systems. 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 1978–1984, doi: 10.1109/BigData.2015.7363978.

[8] Brereton, P.—Kitchenham, B. A.—Budgen, D.—Turner, M.—Khalil, M.: Lessons from Applying the Systematic Literature Review Process within the Software Engineering Domain. Journal of Systems and Software, Vol. 80, 2007, No. 4, pp. 571–583, doi: 10.1016/j.jss.2006.07.009.

[9] Buerli, M.: The Current State of Graph Databases. Department of Computer Science, California Polytechnic State University, San Luis Obispo, December 2012.

[10] Cattell, R.: Scalable SQL and NoSQL Data Stores. ACM SIGMOD Record, Vol. 39, 2011, No. 4, pp. 12–27, doi: 10.1145/1978915.1978919.

[11] Celesti, A.—Fazio, M.—Romano, A.—Bramanti, A.—Bramanti, P.—Villari, M.: An OAIS-Based Hospital Information System on the Cloud: Analysis of a NoSQL Column-Oriented Approach. IEEE Journal of Biomedical and Health Informatics, Vol. 22, 2018, No. 3, pp. 912–918, doi: 10.1109/jbhi.2017.2681126.

[12] Chang, F.—Dean, J.—Ghemawat, S.—Hsieh, W. C.—Wallach, D. A.—Burrows, M.—Chandra, T.—Fikes, A.—Gruber, R. E.: Bigtable: A Distributed Storage System for Structured Data. ACM Transactions on Computer Systems, Vol. 26, 2008, No. 2, Art. No. 4, 26 pp., doi: 10.1145/1365815.1365816.

[13] Chen, C. L. P.—Zhang, C.-Y.: Data-Intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data. Information Sciences, Vol. 275, 2014, pp. 314–347, doi: 10.1016/j.ins.2014.01.015.

[14] Dehdouh, K.—Bentayeb, F.—Boussaid, O.—Kabachi, N.: Using the Column Oriented NoSQL Model for Implementing Big Data Warehouses. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), 2015, pp. 469–475.

[15] Dharavath, R.—Kumar, C.: A Scalable Generic Transaction Model Scenario for Distributed NoSQL Databases. Journal of Systems and Software, Vol. 101, 2015, pp. 43–58, doi: 10.1016/j.jss.2014.11.037.

[16] Ding, L.—Liu, Y.—Han, B.—Zhang, S.—Song, B.: HB-File: An Efficient and Effective High-Dimensional Big Data Storage Structure Based on US-ELM. Neurocomputing, Vol. 261, 2017, pp. 184–192, doi: 10.1016/j.neucom.2016.06.080.

[17] Excoffier, L.—Lischer, H. E. L.: Arlequin Suite Ver 3.5: A New Series of Programs to Perform Population Genetics Analyses under Linux and Windows. Molecular Ecology Resources, Vol. 10, 2010, No. 3, pp. 564–567, doi: 10.1111/j.1755-0998.2010.02847.x.

[18] Gantz, J.—Reinsel, D.: The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. IDC iView: IDC Analyze the Future, December 2012, pp. 1–16.

[19] George, L.: HBase: The Definitive Guide: Random Access to Your Planet-Size Data. O'Reilly Media, Inc., 2011.

[20] Gómez, L.—Kuijpers, B.—Vaisman, A.: Performing OLAP over Graph Data: Query Language, Implementation, and a Case Study. Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics (BIRTE '17), ACM, 2017, Art. No. 6, doi: 10.1145/3129292.3129293.

[21] Hasija, H.—Kumar, D.: Compression and Security in MongoDB Without Affecting Efficiency. Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16), ACM, 2016, Art. No. 96, doi: 10.1145/2905055.2905155.

[22] He, H.—Du, Z.—Zhang, W.—Chen, A.: Optimization Strategy of Hadoop Small File Storage for Big Data in Healthcare. The Journal of Supercomputing, Vol. 72, 2016, No. 10, pp. 3696–3707, doi: 10.1007/s11227-015-1462-4.

[23] Hewitt, E.: Cassandra: The Definitive Guide. O'Reilly Media, Inc., 2010.

[24] InfiniteGraph. Infinitegraph – Distributed Graph Database. 2014.

[25] Iordanov, B.: HyperGraphDB: A Generalized Graph Database. In: Shen, H. T. et al. (Eds.): Web-Age Information Management (WAIM 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6185, 2010, pp. 25–36, doi: 10.1007/978-3-642-16720-1_3.

[26] Islam, M. E.—Islam, M. R.—Ali, A. B. M. S.: An Approach to Security for Unstructured Big Data. The Review of Socionetwork Strategies, Vol. 10, 2016, No. 2, pp. 105–123, doi: 10.1007/s12626-016-0067-6.

[27] Jin, J.—Song, A.—Gong, H.—Xue, Y.—Du, M.—Dong, F.—Luo, J.: Distributed Storage System for Electric Power Data Based on HBase. Big Data Mining and Analytics, Vol. 1, 2018, No. 4, pp. 324–334, doi: 10.26599/BDMA.2018.9020026.

[28] Kang, Y.-S.—Park, I.-H.—Rhee, J.—Lee, Y.-H.: MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data. IEEE Sensors Journal, Vol. 16, 2016, No. 2, pp. 485–497, doi: 10.1109/jsen.2015.2483499.

[29] Khetrapal, A.—Ganesh, V.: HBase and Hypertable for Large Scale Distributed Storage Systems. Department of Computer Science, Purdue University, 2006.

[30] Küçükkeçeci, C.—Yazici, A.: Multilevel Object Tracking in Wireless Multimedia Sensor Networks for Surveillance Applications Using Graph-Based Big Data. IEEE Access, Vol. 7, 2019, pp. 67818–67832, doi: 10.1109/access.2019.2918765.

[31] Kaliyar, R. K.: Graph Databases: A Survey. International Conference on Computing, Communication and Automation, IEEE, 2015, pp. 785–790, doi: 10.1109/ccaa.2015.7148480.

[32] Lakshman, A.—Malik, P.: Cassandra: A Decentralized Structured Storage System. ACM SIGOPS Operating Systems Review, Vol. 44, 2010, No. 2, pp. 35–40, doi: 10.1145/1773912.1773922.

[33] Li, D.—Deng, L.—Cai, Z.: Statistical Analysis of Tourist Flow in Tourist Spots Based on Big Data Platform and DA-HKRVM Algorithms. Personal and Ubiquitous Computing, Vol. 24, 2020, No. 1, pp. 87–101, doi: 10.1007/s00779-019-01341-x.

[34] Liu, B.—Zhu, Y.—Wang, C.—Chen, Y.—Huang, T.—Shi, W.—Li, M.—Mao, Y.: A Versatile Event-Driven Data Model in HBase Database for Multi-Source Data of Power Grid. 2016 IEEE International Conference on Smart Cloud (SmartCloud), IEEE, 2016, pp. 208–213, doi: 10.1109/smartcloud.2016.28.

[35] Mallek, H.—Ghozzi, F.—Teste, O.—Gargouri, F.: BigDimETL with NoSQL Database. Procedia Computer Science, Vol. 126, 2018, pp. 798–807, doi: 10.1016/j.procs.2018.08.014.

[36] Manyika, J. et al.: Big Data: The Next Frontier for Innovation, Competition, and Productivity. Report, McKinsey Global Institute, 2011.

[37] MongoDB. MongoDB Architecture Guide (White Paper), 2015.

[38] Petticrew, M.—Roberts, H.: Systematic Reviews in the Social Sciences: A Practical Guide. John Wiley and Sons, 2008, doi: 10.1002/9780470754887.

[39] Pevec, D.—Vdovic, H.—Gace, I.—Sabolic, M.—Babic, J.—Podobnik, V.: Distributed Data Platform for Automotive Industry: A Robust Solution for Tackling Big Challenges of Big Data in Transportation Science. 2019 15th International Conference on Telecommunications (ConTEL), IEEE, 2019, pp. 1–8, doi: 10.1109/contel.2019.8848542.

[40] Pokorny, J.: NoSQL Databases: A Step to Database Scalability in Web Environment. International Journal of Web Information Systems, Vol. 9, 2013, No. 1, pp. 69–82, doi: 10.1108/17440081311316398.

[41] Pore, S. S.—Pawar, S. B.: Comparative Study of SQL and NoSQL Databases. International Journal of Advanced Research in Computer Engineering and Technology, Vol. 4, 2015, No. 5, pp. 1747–1753.

[42] Putnik, G.—Sluga, A.—ElMaraghy, H.—Teti, R.—Koren, Y.—Tolio, T.—Hon, B.: Scalability in Manufacturing Systems Design and Operation: State-of-the-Art and Future Developments Roadmap. CIRP Annals, Vol. 62, 2013, No. 2, pp. 751–774, doi: 10.1016/j.cirp.2013.05.002.

[43] Rao, P. S.—Satyanarayana, S.: Privacy Preserving Data Publishing Based on Sensitivity in Context of Big Data Using Hive. Journal of Big Data, Vol. 5, 2018, No. 1, Art. No. 20, doi: 10.1186/s40537-018-0130-y.

[44] Rocha, L.—Vale, F.—Cirilo, E.—Barbosa, D.—Mourão, F.: A Framework for Migrating Relational Datasets to NoSQL. Procedia Computer Science, Vol. 51, 2015, pp. 2593–2602, doi: 10.1016/j.procs.2015.05.367.

[45] Russom, P.: Big Data Analytics. TDWI Best Practices Report, Fourth Quarter, Vol. 19, 2011, No. 4, pp. 1–34.

[46] Saranya, S.—Sarumathi, M.—Swathi, B.—Victer Paul, P.—Sampath Kumar, S.—Vengattaraman, T.: Dynamic Preclusion of Encroachment in Hadoop Distributed File System. Procedia Computer Science, Vol. 50, 2015, pp. 531–536, doi: 10.1016/j.procs.2015.04.027.

[47] Sciore, E.: SimpleDB: A Simple Java-Based Multiuser Syst for Teaching Database Internals. ACM SIGCSE Bulletin, Vol. 39, 2007, No. 1, pp. 561–565, doi: 10.1145/1227504.1227498.

[48] Sheehy, J.: Riak 0.10 Is Full of Great Stuff. 2010.

[49] Venkatraman, S.—Fahd, K.—Kaspi, S.—Venkatraman, R.: SQL Versus NoSQL Movement with Big Data Analytics. International Journal of Information Technology and Computer Science, Vol. 8, 2016, No. 12, pp. 59–66, doi: 10.5815/ijitcs.2016.12.07.

[50] Skoulis, I.—Vassiliadis, P.—Zarras, A. V.: Growing Up with Stability: How Open-Source Relational Databases Evolve. Information Systems, Vol. 53, 2015, pp. 363–385, doi: 10.1016/j.is.2015.03.009.

[51] Srinivasan, V.—Bulkowski, B.—Chu, W.-L.—Sayyaparaju, S.—Gooding, A.—Iyer, R.—Shinde, A.—Lopatic, T.: Aerospike: Architecture of a Real-Time Operational DBMS. Proceedings of the VLDB Endowment, Vol. 9, 2016, No. 13, pp. 1389–1400, doi: 10.14778/3007263.3007276.

[52] Srivastava, P. P.—Goyal, S.—Kumar, A.: Analysis of Various NoSQL Database. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2015, pp. 539–544, doi: 10.1109/icgciot.2015.7380523.

[53] Subramaniyaswamy, V.—Vijayakumar, V.—Logesh, R.—Indragandhi, V.: Unstructured Data Analysis on Big Data Using Map Reduce. Procedia Computer Science, Vol. 50, 2015, pp. 456–465, doi: 10.1016/j.procs.2015.04.015.

[54] Tahmassebpour, M.: A New Method for Time-Series Big Data Effective Storage. IEEE Access, Vol. 5, 2017, pp. 10694–10699, doi: 10.1109/access.2017.2708080.

[55] Tulchinsky, I.: The Age of Prediction. WorldQuant Journal, 2017.

[56] Wang, C.—Zhu, Y.—Ma, Y.—Qiu, M.—Liu, B.—Hou, J.—Shen, Y.—Shi, W.: A Query-Oriented Adaptive Indexing Technique for Smart Grid Big Data Analytics. Journal of Signal Processing Systems, Vol. 90, 2018, No. 8-9, pp. 1091–1103, doi: 10.1007/s11265-017-1269-z.

[57] Wang, J.—Wu, H.—Wang, R.: A New Reliability Model in Replication-Based Big Data Storage Systems. Journal of Parallel and Distributed Computing, Vol. 108, 2017, pp. 14–27, doi: 10.1016/j.jpdc.2017.02.001.

[58] White, M.: Digital Workplaces: Vision and Reality. Business Information Review, Vol. 29, 2012, No. 4, pp. 205–214, doi: 10.1177/0266382112470412.

[59] Wilcox, T.—Jin, N.—Flach, P.—Thumim, J.: A Big Data Platform for Smart Meter Data Analytics. Computers in Industry, Vol. 105, 2019, pp. 250–259, doi: 10.1016/j.compind.2018.12.010.

[60] Wu, H.—Zhu, Y.—Wang, C.—Hou, J.—Li, M.—Xue, Q.—Mao, K.: A Performance-Improved and Storage-Efficient Secondary Index for Big Data Processing. 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017, pp. 161–167, doi: 10.1109/smartcloud.2017.32.

[61] Yassien, A. W.—Desouky, A. F.: RDBMS, NoSQL, Hadoop: A Performance-Based Empirical Analysis. Proceedings of the 2nd Africa and Middle East Conference on Software Engineering (AMECSE'16), ACM, 2016, pp. 52–59, doi: 10.1145/2944165.2944174.

[62] Zheng, X.—Fu, M.—Chugh, M.: Big Data Storage and Management in SaaS Applications. Journal of Communications and Information Networks, Vol. 2, 2017, No. 3, pp. 18–29, doi: 10.1007/s41650-017-0031-9.

[63] Pan, Z.—Zhao, L.: Application and Research of Massive Big Data Storage System Based on HBase. 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 2018, pp. 219–223, doi: 10.1109/icccbda.2018.8386515.

Amen Faridoon is currently Ph.D. student in computer science at the University College Dublin in Ireland. She received her Bachelor's degree in computer science from the Govt. Girls Post Graduate College No. 1, Abbottabad, Pakistan in 2018. She then worked as Research Assistant at National Centre for Physics, Islamabad, Pakistan in 2019. Her research interests include machine learning, big data, data mining and cloud computing.

Muhammad Imran received his Ph.D. degree in electronic engineering from the Dublin City University, Ireland, in 2017. He is currently working in CMS Offline Computing Group at CERN, Geneva, Switzerland since October 2019. In addition, he holds a permanent position as Senior Scientific Officer in the National Centre for Physics, Pakistan since July 2008. His research interests include cloud computing, cluster computing, big data, data science, software engineering, SDN, and optical networks.