Mining Developer Questions about Major NoSQL Databases

Saiful Islam, Khalid Hasan, Rifat Shahriyar
Bangladesh University of Engineering and Technology (BUET)
Dhaka, Bangladesh
saifulislamt073@gmail.com, 72.khalidhasan@gmail.com, rifat@cse.buet.ac.bd

ABSTRACT

NoSQL databases are quickly becoming more and more popular among developers. While RDBMS is still the most popular, the NoSQL camp is closing in the gap. To bridge this gap, we aim to carry out the first empirical software engineering research on NoSQL databases using Stack Overflow posts. Being one of the leading question-answering sites available, Stack Overflow has become a helpful resource in numerous software engineering research. In this paper, we chose five NoSQL databases, MongoDB, Cassandra, Redis, Neo4j, and HBase, based on their popularity and the increasing number of posts on Stack Overflow. We extracted the relevant questions and investigated different challenges and issues faced by the developers of NoSQL databases and the various domains the NoSQL databases are used by mining questions asked on Stack Overflow. We sorted the issues by popularity and difficulty metrics and observed the different nature of difficulty and popularity. We found that connection issues and integration are the most common difficult issues the developer of NoSQL databases faced. We also found that Cassandra, HBase, and Neo4j are very popular among Java developers, MongoDB and Redis are very popular among node.js developers, and Cassandra and HBase are very popular for big data systems. Our findings will help better understand the challenges, requirements, and specific applications of the NoSQL databases.

Keywords

NoSQL Database, Stack Overflow, LDA

1. INTRODUCTION

With the exploitation of the Internet and cloud computing, modern databases have to save and utilize data effectively. They need to satisfy the demand for high performance when reading and writing. So, the traditional relational databases are facing many new challenges. Especially in large scale and high-concurrency applications, such as search engines and SNS, using the relational database to store and query dynamic user data has appeared inadequate. NoSQL databases were created to counter these shortcomings.

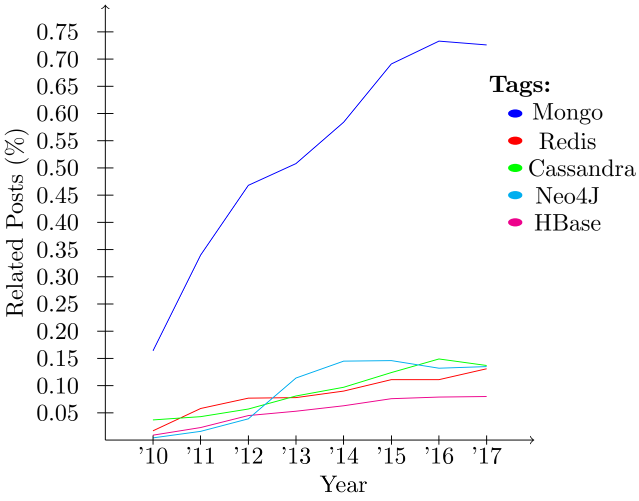

Fig. 1. Percentage of related posts per year in Stack Overflow

NoSQL refers to the next generation of non-relational databases. Starting from the demands to have more flexible, scalable, and less ACID-restricted databases [15] rather than traditional Relational databases, NoSQL databases fit for storing unstructured data and heavy read-write workloads [17]. Developers from Internet giants and tech start-ups have created a slew of NoSQL databases designed to evade SQL-based relational software's schema rigidity. However, their progeny has some issues that can complicate efforts to deploy and manage enterprise NoSQL architecture to support business intelligence initiatives and big data applications. According to Guy Harrison [12], the NoSQL database's promise has generated a lot of enthusiasm, but there are many obstacles to overcome before they can appeal to mainstream enterprises. One of a few of the top challenges is maturity. Most NoSQL alternatives are in pre-production versions with many key features yet to be implemented. Another top challenge is the availability of expertise. Almost every NoSQL developer is in a learning mode. This situation will be addressed naturally over time, but for now, it's far easier to find experienced RDBMS programmers or administrators than a NoSQL expert .

The increasing demand for NoSQL is noticeable in Stack Overflow from Figure 1. Over the last decade, the NoSQL related posts have only been increased in number. So, we decided to mine questions that are on NoSQL and study what the developers are asking. Our findings will improve the overall quality of NoSQL databases. For our research, we extracted posts from Stack Overflow (SO), of which 1,54,83,377 were questions. Then we differentiated the questions on NoSQL. Then we looked into the questions to find the issues that NoSQL developers face.

Stack Overflow has been being used in the research area due to the availability of high-quality useful data. It is specifically designed to optimize the experience of asking and answering programming specific questions and is targeted explicitly at hobbyist and professional programmers. Stack Overflow is also specially designed to ask and answer objectively provable questions. It actively discourages opinion-based questions. In 2016 and 2017, 39 major academic papers, and in 2018, about 16 papers were published using Stack Exchange Data [1]. These publications mainly focused on finding out trending issues or the question/answer culture on Stack Overflow. But no such work has been done on NoSQL.

In our research, we have picked five NoSQL databases, MongoDB, Cassandra, Redis, Neo4j, and HBase, based on their popularity and the increasing number of posts. We extracted bodies of all the questions and used them to train our topic model on Mallet. After extracting the particular topic of each question, we researched to find answers to our research questions.

—**RQ1:** What are the issues users of these NoSQL databases face?

—**RQ2:** Which are the most difficult topics?

—**RQ3:** What Are the common issues the 5 NoSQL Databases share? What are the unique issues of each database?

—**RQ4:** What domains use NoSQL databases most?

From our experiment, we found 8 issues for Cassandra, 7 issues for HBase, 8 issues for MongoDB, 8 issues for Neo4j, and 7 issues for Redis. We also ranked the issues according to their popularity and difficulty. We also find that Connection issues, Structure (Questions on tables, designs, query system, data structure, etc.), Integration (Using the NoSQL databases with popular frameworks and programming languages) are the common issues among all databases. We separated issues that are unique only to each database. We then looked into where these databases are being used. We found that they are very popular with web or mobile development.

Our research can be useful to various parties. All frameworks or programming languages use some drivers or tools to connect to NoSQL databases. Developers of those technologies can use our research result to improve their technology or support further. Developers can now choose what database to use for their purpose as they can measure all the pros and cons of each database.

2. BACKGROUND AND RELATED WORKS

In this section, some background information about NoSQL databases and the previous works that have motivated our research are mentioned.

2.1 Background

In this research, the top 5 NoSQL databases are investigated. Let us briefly introduce them:

Cassandra. Cassandra is a distributed, open-source NoSQL database designed to handle large amounts of data, providing high availability with no single point of failure [2].

HBase. HBase is an open-source, non-relational, distributed database modeled after Google's Bigtable [10] and written in Java. HBase began as a project out of a need to process massive amounts of data for natural-language search [3].

MongoDB. MongoDB is a document-oriented database program. Classified as a NoSQL database program, MongoDB stores data in flexible, JSON-like documents [4].

Neo4j. Neo4j is a graph database management system. Neo4j is the most popular graph database according to DB Engines ranking and the 22nd most popular database overall [5].

Redis. Redis is an open-source, in-memory data structure project implementing a distributed, in-memory key-value database with optional durability [6].

2.2 Related Works

We looked into some papers that used Stack Overflow data to analyze the trends, issues among developers of various technologies such as web development, mobile development, frameworks, etc. NoSQL has been on the rise recently, and no such research was done for NoSQL development. So, we were motivated to use Stack Overflow resources to find the issues and trends of NoSQL developers.

Bajaj et al. studied some common challenges, misconceptions among web developers [7]. They conducted a mixed-method analysis of the data from the Stack Overflow data dump [11]. Rosen et al. researched the most common issues encountered by mobile developers. They used LDA based topic modeling on posts related to mobile development and researched into what mobile developers are asking about [16]. Barua et al. also used LDA to extract what the developers are talking about [8].

3. METHODOLOGY

In this section, we present various stages of our research methodology.

3.1 Data Extraction

Stack Exchange releases a data dump of all its publicly available content roughly every three months via archive.org. It also makes that information query-able over the Internet at the Stack Exchange Data Explorer [11]. These data dumps are offered in XML form. Data dump contains information on posts with a list of answers, upvotes, downvotes, accepted answers, comments, users with user reputation, rating, etc. For our research, we downloaded the latest data dump that was available at that time. Our data contained data from September 2008 to March 2018. There were several XML files, namely posts.xml, users.xml, tags.xml, etc. In our research, we only needed posts.xml as our interest lies in the Stack Overflow posts. We wrote a script to extract the contents of the posts.xml file to a table in the MySQL database for convenience.

3.2 Data Processing

After we ran the script, we had a table of 3,96,46,923 entries. All of them were 'posts,' which included questions, answers, and comments. There were 1,54,83,377 questions, but we only needed the questions that served our purpose. So, we ran a query to extract questions that contained tags of the names of our 5 databases. The Table 1 shows the number of posts of each of the databases: Then, we extracted the question body from these entries. Question bodies

Table 1. Number of Posts of five NoSQL databases

NoSQL Database	Number of Posts
MongoDB	86807
Cassandra	14893
HBase	8926
Neo4j	15799
Redis	13842

are stored in HTML format. So, we got rid of the unwanted tags, code fractions, etc. We then separate the five types of posts related to our five databases and put them in separate folders. Next, we apply LDA to the question bodies.

3.3 Topic Modeling using LDA

Topic modeling is the task of identifying topics that best describes a set of documents. These topics will only emerge during the topic modeling process. One popular topic modeling technique is Latent Dirichlet Allocation (LDA) [9]. Each document or question body contains a mixture of different topics. So, it is easy to discover the theme of a document [16].

We use Mallet to perform topic modeling. Mallet is a Java-based package for topic modeling [14].

At first, we import our question bodies saved as text files with the mallet 'import' command. It combines all the text files into a mallet file. Then we use the mallet file to perform topic modeling by using the 'train-topics' command. This command has a parameter called 'num_topics'. We run the command with num_topics value of 50. This command opens the mallet file and trains the mallet to find 50 topics. It outputs every word in the corpus of materials and its topic into a compressed file. It also outputs a text document showing the top keywords for each topic and outputs a text file indicating the breakdown, by percentage, of each topic within each original text file imported.

We then analyzed the words for each topic. Some of the word sets of those topics are very similar in meaning. So, we merged word sets that were similar and named the topics manually. We then noticed some of the topics had the same names or scopes. So, we merged them as well. We followed the same procedure for each of the five sets of data.

4. RESEARCH FINDINGS

In this section, we present our research findings and answer the questions we formulated earlier.

4.1 RQ1: What are the issues users of these databases face?

To answer the first question, LDA was applied on the five datasets. Then the topics were named manually so that it can be understood clearly. Here are the topics of the five databases that were found.

4.1.1 Cassandra. In Table 2, the topics found in the Cassandra posts are shown. Each topic was organized according to views per question. The topic that gets more views can be said to be more popular. So, Table 2 is a ranking by popularity.

According to calculation, data manipulation and data security take the top place in popularity. Let's call this topic data. 'Installation' represents questions on installation issues the users face. 'Datastax' is a distributed cloud database built on Cassandra. 'Errors and exceptions' means the exceptions that Cassandra may face. 'Import' topic covers all questions regarding the import or export process of various data in Cassandra. 'Structure' means questions related to data structures, table design, query system, etc., 'Connection Issues' encompasses Cassandra connection issues. 'Integration' means questions about using Cassandra with frameworks and languages.

Table 2. Popularity rank of Cassandra topics

Topic	Ques	View	View/Ques
Data	1360	2437293	1792
Installation	87	130462	1499
Datastax	88	123000	1397
Errors & exceptions	1342	1550532	1155
Import	252	279356	1108
Structure	7036	7115628	1011
Connection Issues	2963	2910585	982
Integration	1765	1366599	774

Table 3. Popularity rank of Hbase topics

Topic	Ques	View	View/Ques
Connection Issues	711	1028547	1446
Installation	40	52559	1313
Structure	2189	2376483	1086
Integration	4890	5053804	1033
Import	559	526809	942
Errors & Exceptions	316	252366	798
Disk space issues	221	145246	657

Table 4. Popularity rank of MongoDB topics

Topic	Ques	View	View/Ques
Comparative discussion	6144	11349377	1847
Installation	2394	4173120	1743
Connection Issues	10695	18535793	1733
Data Distribution	6102	9447623	1548
Integration	14919	17062280	1663
Features	27755	32103646	1156
Structure	15964	17791577	1114
Errors & Exceptions	2834	2152073	759

4.1.2 Hbase. Table 3 is the topics of the hbase related posts according to their popularity.
Disk space is a quite major topic of HBase posts. HBase is used with big data quite often. Thus, it was picked as a topic. Other topics mean the same as discussed previously on Cassandra's issues.

4.1.3 MongoDB. Table 4 shows the topics of mongodb posts.
'Comparative discussion' topic represents posts that query comparative discussion between MongoDB and other NoSQL databases. 'Data Distribution' includes questions about data partitioning, sharding, and other distribution methods. The 'Features' topic is related to MongoDB features. 'Geospatial features', 'time data' are something unique to MongoDB.

4.1.4 Neo4j. Table 5 shows the Neo4j topic found.

Neo4j is a graph-based database. 'Graph Platform' are posts about the definition and identity in the Neo4j database. Indexing is a process of keeping redundant copies of some data in the database for making searches of related data more efficient. As the data is stored,

Table 5. Popularity rank of Neo4j topics

Topic	Ques	View	View/Ques
Comparison	1034	1147949	1110
Connection Issues	1357	1127771	831
Graph Platform	167	115503	691
Integration	3397	1695091	498
Features	756	352600	466
Structure	8544	3908886	457
Import	331	150009	453
Errors & Exceptions	213	85119	399

Table 6. Popularity rank of Redis topics

Topic	Ques	View	View/Ques
Connection issues	835	1899530	2274
Installation	293	550476	1878
features	1262	2361976	1871
Structure	5028	7620718	1515
Integration	5906	7963251	1348
Social networking	401	453357	1130
Testing	117	112714	963

it is connected with the DB size topic. Both these topics are then included in the ‘Features’ topic.

4.1.5 Redis. Table 6 shows the redis questions topics.

Redis was found most being used in social media. These questions are about ‘Social networking’ topics. Redis is often used with software unit testing. ‘Testing’ topic includes all those posts that include software testing.

RQ1: What are the issues users of these databases face? Users face data manipulation and data security related issues most in using Cassandra. Again, in using Hbase, disk space related issue is an unique topic to consider and it is used with Big data quite often. For using MongoDB, users usually tend to picture a comparative discussion with other NoSQL databases. Moreover, graph platform related questions are common for Neo4j database practitioner. Also, social networking turns out to be an important topic for Redis database users.

4.2 RQ2: Which are the most difficult topics?

Here presented is the difficulty ranking of all the topics from the five databases. Finding difficult issues will help the NoSQL developers to focus on the most relevant topics.

The difficulty is defined as the time needed for a question to have an accepted answer. Many answers may be posted on a question, but only a few are accepted. The mean and median time needed for a question to get an accepted answer was calculated. The percentage of questions that find an accepted answer and the average number of answers (that are not accepted but answers nonetheless) that each topic has was also calculated.

Previous research shows that most comments, answers, and activities happen within the first hour of the questions [13]. We determine the difficulty level of the topics by the median time taken for an accepted answer as the mean is likely to be heavily distorted by latency responses [16]. There are also a lot of questions that are answered. But the number of questions answered is not a good measure to find the difficulty of a certain topic. A question can be unanswered for many reasons, such as trivial queries, bad composition, etc. According to calculations, the overall percentage of questions to receive an accepted answer is 53.44%.

Table 7. Difficulty of Cassandra topics

Topic	Mean(Minutes)	Median(Minutes)
Data	20863	313
Installation	29233	305
Datastax	10213	313
Errors & Exceptions	19050	332
Import	17473	253
Structure	10718	280
Connection Issues	19294	325
Integration	15177	589

Table 8. % of questions with accepted answers and average number of answers for Cassandra

Topic	% Accepted	Avg Answers
Data	45	1.47
Installation	41	1.15
Datastax	30	1.27
Errors & Exceptions	42	1.21
Import	40	1.04
Structure	47	1.17
Connection Issues	40	1.16
Integration	40	1.07

The percentage of questions with accepted answers depends not only on the difficulty of the question posted but also on the writing style, content of the questions. Often poorly written questions don’t get an answer. That’s why only the median times were considered when calculating difficulty.

4.2.1 Difficulty of Cassandra Topics. Table 7 lists the Cassandra topics along with the mean, the median time it takes for questions in a topic to get at least one ‘accepted’ answer. Table 8 displays the percentage of Cassandra questions that receive at least an accepted answer and also the average number of answers that each question received. The topics are arranged according to their popularity as defined in RQ1.

Table 7 shows that ‘Integration’ has the maximum median among all the topics. Also, from Table 8, we notice that ‘Integration’ has the least percentage of accepted answers. So ‘Integration’ is the most difficult topic, followed by Errors and Exception, Connection Issues, Data, Datastax, Installation, and structure. It takes 253 to 589 minutes to get an accepted answer for a Cassandra related question. Table 8 shows that the percentage of finding accepted answers varies from 40% to 47%, and there is about one answer on average to each question. ‘Data’ related topics have the highest percentage of accepted answers with 45% lower than the average value. That can be explained by the lack of expertise in NoSQL technologies.

4.2.2 Difficulty of Hbase Topics. In Table 9, the mean and median time before an accepted answer is posted. ‘Connection Issue’ is the topic with the highest median time before an accepted answer. So, ‘connection issues’ is the most difficult topic in HBase. Other difficult topics in order are: Disk space issues, Integration, Errors and exceptions, Import, Structure. So HBase can improve its documentation on connection issues so that the developers using it can make their lives easier.

Table 9. Difficulty of Hbase topics

Topic	Mean(minutes)	Median(minutes)
Connection Issues	19140	1392
Installation	44908	460
Structure	18189	608
Integration	26797	809
Import	12815	782
Errors& Exceptions	18674	801
Disk space issues	28871	1195

Table 10. % of questions with accepted answers and average number of answers for Hbase

Topic	% Accepted	Avg Answers
Connection Issues	32. 9	1. 2
Installation	22. 5	1. 125
Structure	39. 7	1. 12
Integration	39. 5	1. 16
Import	33. 1	1. 09
Errors& Exceptions	39. 2	1. 01
Disk space issues	37. 1	1. 045

Table 11. Difficulty of MongoDB topics

Topic	Mean(minutes)	Median(minutes)
Comparative discussion	16433	113
Installation	16673	171
Connection Issues	15658	88
Data Distribution	17364	132
Integration	16361	154
Features	10416	70
Structure	9873	70
Errors & Exceptions	7168	73

Table 10 shows the percentage of questions that received an accepted answer. The percentage varies from 22.5% to 39.7%. These low percentages can be interpreted in some ways. One way is that there are very few experts on HBase, or there are few questions. Indeed, there are only 8,926 questions on HBase in the past 8 years. So that explains the low percentage of accepted answers.

4.2.3 Difficulty of MongoDB Topics. Table 11 shows all the topics arranged according to their popularity along with the required mean and median time for an accepted answer. MongoDB topics take a median time of range 70 minutes to 171 minutes to get an accepted answer. Installation takes the most time with a mean time of 171 minutes. Also, according to Table 12, 'Installation' has the second-lowest percentage. So, 'Installation' is the most difficult topic. The ranking of difficulty is Integration, Data distribution, Comparative discussion, Connection issues, Errors and exceptions, Features, and Structure.

This shows that MongoDB has a major issue with its installation process. Thorough documentation and customer support can be of good use regarding this issue.

Table 12 displays the percentage of questions that received an accepted answer and the average number of answers per question. The percentage varies from 40.6% to 52.71%. Structure topic has the highest percentage out of all topics. The questions received at least 1 answer per question.

The data for MongoDB are close to the average value, and also, the median time taken for answers is less than other databases. It is proof that MongoDB is way more popular and more used than the other four databases that were selected.

Table 12. % of questions with accepted answers and average number of answers for MongoDB

Topic	% Accepted	Avg Answers
Comparative discussion	48.1	1. 39
Installation	41.29	1. 21
Connection Issues	44.39	1. 19
Data Distribution	43.67	1. 13
Integration	49.6	1. 15
Features	51.28	1. 17
Structure	52.71	1. 15
Errors & Exceptions	40.6	1. 04

Table 13. Difficulty of Neo4j topics

Topic	Mean(minutes)	Median(minutes)
Comparison	17438	208
Connection Issues	11124	310
Graph platform	7101	373
Features	11224	174
Integration	13595	321
Structure	4786	122
Import	4317	111
Errors & Exception	11849	1105

Table 14. % of questions with accepted answers and average number of answers for Neo4j

Topic	% Accepted	Avg. Answers
Comparison	47. 96	1. 43
Connection Issues	44. 87	1. 11
Graph platform	33. 5	1. 02
Features	48. 8	1. 15
Integration	47. 65	1. 1
Structure	58. 4	1. 22
Import	50. 4	1. 26
Errors & Exception	47. 8	1. 14

4.2.4 Difficulty of Neo4j Topics. Table 13 lists all the mean and median time required for an accepted answer. Median times range from 111 minutes to 1105 minutes. 'Errors and exception' takes 1105 minutes to receive an accepted answer. Errors and exceptions are the most difficult topics. The difficulty rank is as follows- Errors and exceptions, Graph platform, Integration, Connection issues, Structure, Import. The mean time ranges from 4317 minutes to 17438 minutes.

Table 14 represents the percentage of questions with accepted answers. The percentage varies from 33.5% to 58.4%. Each topic has at least 1 answer, whether it was accepted or not. For Neo4j, the percentage is closer to the average value. Questions find answers with higher percentages, which is conducive for beginners.

4.2.5 Difficulty of Redis Topics. According to Table 15 the Redis topics take from 165 to 344 minutes to achieve an accurate answer. Question of topic 'Features' takes the most time with 344 minutes making it the most difficult topic. Difficulty ranking is as follows - Integration, Connection issues, Social networking, Testing, Structure, Installation.

Features deal with Redis snapshots. Redis does a complete copy of what's in memory at some points in time. When you lose power

Table 15. Difficulty of Redis topics

Topic	Mean(minutes)	Median(minutes)
Connection issues	23411	266
Installation	17278	165
Features	28486	344
Structure	19297	178
Integration	18675	283
Social networking	5661	203
Testing	24777	183

Table 16. % of questions with accepted answers and average number of answers for Redis

Topic	% Accepted	Avg Answers
Connection issues	43	1.18
Installation	41.3	1.17
Features	43.6	1.19
Structure	51	1.22
Integration	44.76	1.11
Social networking	48.12	1.16
Testing	48.7	1.13

between two snapshots, you will lose the data from the last snapshot and the crash. It proves to be the most difficult topic for Redis.

Table 16 shows the percentage of questions with accepted answers and the average number of answers per question. The percentage ranges from 41.3% to 51%. The structure is the most answered topic in Redis, while 'Connection issues' is the least answered topic. Additionally, each topic has at least one answer per question.

RQ2: Which are the most difficult topics?
Integration is the most difficult topic Cassandra users face, followed by errors and exceptions. Furthermore, connection and disk-space related issues are prevalent in using Hbase. Again, the installation-related issue is one of the major concerns for Mongodb, followed by integration, and data distribution, though it is the most popular NoSQL database. For Neo4j users, errors and exceptions are the most difficult topics to face, along with graph platform, integration, etc. Also, for using Redis, users consider features (i.e., features deal with Redis snapshots') related topics being the most difficult, followed by integration, connection issues, etc.

4.3 RQ3: What Are the common issues the 5 NoSQL Databases share? What are the unique issues of each database?

This question looks into the common problems among users and unique problems in each of the databases. From previous tables, it is easily perceived that Connection issues, Structure and Integration are common issues among all. We now briefly discuss all these common issues:

Connection Issues. Connecting to a database may result in various exceptions, errors, and challenges. Moreover, there are various tools and providers for connection to a database. So it is no surprise that connection issues are common among all the databases.

Structure. Every database system has its own data structures, table design, query system, etc. This topic deals with those issues. So, it is rightly a common topic among all databases.

Integration. Integration is the topic which includes posts regarding using the database as a back-end to various popular frameworks and also using popular languages in database operations. All five of the databases have questions associated with this issue.

Now we discuss topics that are unique to each of our five databases.

Topics unique to Cassandra. 'Datastax' is only a Cassandra topic. Cassandra uses DataStax as its data platform. So, a fair amount of posts in Cassandra are related to DataStax.

Topics unique to Hbase. HBase has two unique topics - Domain Classes and Disk space. Domain classes topic deals with the usage of domain objects. Disk space means issues regarding memory spaces. It means HBase can improve both these issues to face the expectation of their developers.

Topics unique to MongoDB. MongoDB has Data Distribution as its unique topic. MongoDB fashions sharding, locks, etc., to facilitate a distributed system. It shows that MongoDB is a primary choice for distributed systems.

Topics unique to Neo4j. Neo4j has a Graph platform as its unique topic. Neo4j is a graph-based database. In that way, it is unique to other databases.

Topics unique to Redis. Redis has Social networking and Testing, which is unique to it alone. Redis is widely used in social media. Also, testing is done via Redis quite often.

RQ3: What are the common issues amongst databases?
Questions relating to database connection, data structure and integration are common for all five of the databases.

4.4 RQ4: What Domains use NoSQL databases most?

This question queries which domains work most with the NoSQL databases. To solve this problem, the tags associated with the posts were extracted. Usually, there are multiple tags in a question, and each tag is allowed to be added only once. So, for this question, the tags relevant to the platforms were considered. Then the number of times each of those tags was added with the questions was counted, and then we the tags with the most counts were listed.

4.4.1 Cassandra. Table 17 ranks the languages and web frameworks involved with Cassandra. From the table, it is seen that Cassandra is heavily used with Java. Python and Scala are also prominently used. Cassandra is quite popular with Scala users.

Another significant tag here is 'Big data'. Cassandra is a favorite for back-end development applications, analytics etc. So, Cassandra is a good choice among NoSQL databases when working with Big data.

4.4.2 Hbase. Table 18 is showing the ranking of different tags in HBase related posts. Table 18 shows Java on top of the list, and Java has outnumbered all other platforms. So HBase is mostly used with Java and Java related platforms.

Big data is also near the top. So, HBase is also becoming popular for Big data like Cassandra. Here, the android framework is a tag that we overlooked in previous NoSQL databases. So HBase is the

Table 17. Frequency of tags in Cassandra questions

Tags	Count	Tags	Count
Java	1840	Python	572
Scala	525	PHP	257
Node.js	249	C#	210
Spring	197	Big data	179
Spring-data	133	Spring-boot	131

Table 18. Frequency of tags in Hbase questions

Tags	Count	Tags	Count
Java	1454	Big data	236
Python	219	scala	218
Android	204	Node.js	203
C#	157	PHP	153
Spring	125	Spring-data	90
Javascript	73		

Table 19. Frequency of tags in MongoDB questions

Tags	Count	Tags	Count
Node.js	13300	Javascript	8375
Java	6299	Python	4905
PHP	4787	Ruby-on-rails	3763
C#	3507	Ruby	2178
Spring	2096	Angular js	1430
Scala	1065	Django	1025

most popular choice for NoSQL database among android developers.

4.4.3 MongoDB. Table 19 is the ranking of platforms with respect to the highest number in MongoDB related posts.

As can be seen, Node.js is most used with MongoDB. Also, MongoDB has the highest number of questions. So, MongoDB is more popular among Node.js developers. JavaScript, Java, Python, PHP all have substantial contributions to MongoDB posts. Even all other platforms related questions have been discussed in a good number with respect to all other NoSQL databases. MongoDB is the top choice among all the other NoSQL databases.

4.4.4 Neo4j. Tags ranked by their presence in Neo4j are shown in Table 20. Java outnumbered other web frameworks to be in the top position. That means the number of Java related questions is higher than other platforms in Neo4j tagged posts. So, Neo4j is used most with Java, and for this reason, Java related questions are much higher in number. There are also significant posts on Spring, Node.js, Python, JavaScript, etc.

Neo4j has regular uses in social networking, fraud detection, real-time recommendation systems, knowledge graph, network, and IT operations. So, it is very predictable to see the tags that mostly consist of frameworks and languages.

4.4.5 Redis. Table 21 shows top associated tags from Redis tagged posts. It can be seen from the table that people working with Node.js have asked the most questions about Redis. It implies that Redis is most used with Node.js. People from Python, PHP, Django, Spring, and Laravel backgrounds contribute to the posts but in much lower percentages than Node.js. There is also a tiny amount of posts regarding Go and C. That shows Redis is less used in those platforms.

Table 20. Frequency of tags in Neo4j questions

Tags	Count	Tags	Count
Java	1932	Spring	509
Python	499	Spring-data	281
C#	278	Node.js	233
Ruby-on-rails	219	PHP	210
Javascript	175	Ruby	122
Scala	113		

Table 21. Frequency of tags in Redis questions

Tags	Count	Tags	Count
Node.js	1987	Python	1056
PHP	879	Ruby-on-rails	818
Java	805	C#	614
Ruby	539	Javascript	494
Django	423	Spring	371
Laravel	279	Spring-boot	224
Spring-data	108	Go	102
C	93		

Redis is mostly used by people from a web development background and also object-oriented programming language experts, most notably with C# for caching purposes.

> ***RQ4: What Domains use NoSQL databases most?***
> Java is the most used platform for almost every NoSQL database. All popular frameworks like Django, Spring, Node.js are used with all the databases described in some capacities or other. Popular languages like Python, Ruby, PHP, C# all are quite popular for using NoSQL databases. It indicates that NoSQL databases have vast usage with web applications. HBase and Cassandra are popular choices for Big data. HBase is also the most popular among all these databases for android development. MongoDB and Cassandra databases have been used more, but other databases are also becoming popular for different languages and web frameworks.

5. FUTURE WORK AND CONCLUSION

In our paper, we looked into the issues that NoSQL developers face from the data provided by Stack Overflow. We analyzed the questions to get our desired results.

Our results have produced the topics of questions in the five databases, the difficulty ranking of those topics, the common topics or issues among the NoSQL databases, and also where the databases are used most. We have figured out that Connection issues, Integration and Structure are the common issues. That proves that all NoSQL databases are used for web or mobile development. We have also found that most NoSQL databases are mostly used with Java, Node.js. Cassandra and HBase are the primary choices for Big data systems.

By comparing the results of our five databases, we have observed that MongoDB is by far the more popular than the other databases. The number of questions is huge for MongoDB, the topics for MongoDB are less difficult, and the percentage of questions to receive an accepted answer is close to the average percentage. Also, MongoDB has more applications in fields such as web development, mobile development, etc.

Our result can be beneficial to make more NoSQL databases easier to use. Documentations for difficult topics can be prepared, or researchers can dive into these topics to improve database usability. Tool providers can use our results as a review of their product and improve its future versions. Managers can make a decision based on their requirements.

6. REFERENCES

[1] Academic papers using Stack Exchange data. `https://meta.stackexchange.com/questions/134495/academic-papers-using-stack-exchange-data`. Accessed: 2020-12-19.

[2] Apache Cassandra. `https://cassandra.apache.org/`. Accessed: 2020-12-19.

[3] Apache HBase. `https://hbase.apache.org/`. Accessed: 2020-12-19.

[4] The most popular database for modern apps, MongoDB. `https://www.mongodb.com/`. Accessed: 2020-12-19.

[5] Neo4j graph platform, the leader in graph databases. `https://neo4j.com/`. Accessed: 2020-12-19.

[6] Redis. `https://redis.io/`. Accessed: 2020-12-19.

[7] Kartik Bajaj, Karthik Pattabiraman, and Ali Mesbah. Mining questions asked by web developers. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 112–121, New York, NY, USA, 2014. ACM.

[8] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3):619–654, Jun 2014.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan 2003.

[10] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Michael Burrows, Tushar Chandra, Andrew Fikes, and Robert Gruber. Bigtable: A distributed storage system for structured data (awarded best paper!). In Brian N. Bershad and Jeffrey C. Mogul, editors, *7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA*, pages 205–218. USENIX Association, 2006.

[11] Stack Exchange. Data dump. `https://archive.org/details/documentation-dump.7z`, march 2019.

[12] Guy Harrison. *Next Generation Databases: NoSQL and Big Data*. Apress, Berkely, CA, USA, 1st edition, 2015.

[13] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design lessons from the fastest Q&A site in the west. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2857–2866, New York, NY, USA, 2011. ACM.

[14] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. `http://mallet.cs.umass.edu`, 2002.

[15] Kuldeep Singh Renu Kanwar, Prakriti Trivedi. NoSQL, a solution for distributed database management system. *International Journal of Computer Applications*, 67(2), 4 2013.

[16] Christoffer Rosen and Emad Shihab. What are mobile developers asking about? a large scale study using stack overflow. *Empirical Software Engineering*, 21(3):1192–1223, Jun 2016.

[17] Hailing Zhang, Yang Wang, and Junhui Han. Middleware design for integrating relational database and NoSQL based on data dictionary. *International Journal of Computer Applications*, 12 2011.