ISSN 2219-9365

https://doi.org/10.31891/2219-9365-2023-76-1
UDC 004.9

SHYKHMAT Anton
Lviv Polytechnic National University
https://orcid.org/0000-0003-1732-7408
e-mail: anton.o.shykhmat@lpnu.ua
VERES Zenoviy
Lviv Polytechnic National University
https://orcid.org/0000-0002-2312-2575
e-mail: zenovii.y.veres@lpnu.ua

SELECTION OF DATABASES TO STORE GEOSPATIAL-TEMPORAL DATA

The proliferation of geospatial-temporal data, driven by the widespread adoption of sensor platforms and the Internet of Things, has escalated the demand for effective data management solutions. In this context, GeoMesa, an open-source toolkit designed to enable comprehensive geospatial querying and analytics in distributed computing systems, plays a pivotal role. GeoMesa seamlessly integrates geospatial-temporal indexing capabilities with databases like Accumulo, HBase, Google Bigtable, and Cassandra, facilitating the storage and management of extensive geospatial datasets. This article addresses the critical need to benchmark and compare the performance of Accumulo and Cassandra when employed as underlying data stores for GeoMesa. By conducting performance tests, we aim to provide valuable insights into the relative strengths and weaknesses of these database systems, thereby aiding decision-makers in selecting the most suitable solution for their specific application requirements. The evaluation includes an in-depth analysis of performance metrics, such as throughput and latency, as well as consideration of system parameters, query density, and data access distribution. It was identified that Accumulo outperforms Cassandra almost in all areas – read latency and resource usage under heavy load and write latency under any load. In turn, Cassandra has lower read latency under low load and CPU usage under heavy load.
Keywords: IoT, geospatial-temporal data, Cassandra, Accumulo, GeoMesa

ШИХМАТ Антон, ВЕРЕС Зеновій
Національний університет «Львівська політехніка»

ВИБІР БАЗ ДАНИХ ДЛЯ ЗБЕРІГАННЯ ГЕОПРОСТОРОВИХ ТА ЧАСОВИХ ДАНИХ

Активне використання геопросторових та часових даних, яке спричинене широким поширенням Інтернету речей, підвищило актуальність ефективного управління даними. У цьому контексті ключову роль відіграє GeoMesa, відкрите програмне забезпечення, що призначене для забезпечення геопросторового аналізу в розподільних обчислювальних системах. GeoMesa інтегрується з такими базами даних, як Accumulo, HBase, Google Bigtable та Cassandra, сприяючи зберіганню та управлінню геопросторовими наборами даних. У цій статті порівнюється продуктивність Accumulo та Cassandra, коли вони використовуються як основні сховища даних для GeoMesa. Проведення тестів продуктивності допомагає надати цінні відомості про переваги та недоліки цих систем баз даних, що допоможуть розробникам програмного забезпечення вибрати найбільш оптимальне рішення для конкретних вимог застосування. Оцінка включає аналіз показників продуктивності, таких як пропускна спроможність та затримка, з врахуванням параметрів системи, частоти запитів та доступу до даних. В результаті дослідження встановлено, що Accumulo має кращу продуктивність ніж Cassandra практично в усіх аспектах - швидкість читання та використання ресурсів при великому навантаженні та швидкість запису при будь-якому навантаженні. У свою чергу, Cassandra має більшу швидкість читання при низькому навантаженні та використовує менше процесорного часу при великому навантаженні.
Ключові слова: геопросторові та часові дані, Cassandra, Accumulo, GeoMesa

Formulation of the problem

Geospatial data analysis is crucial for agriculture vehicles predictive maintenance. It provides valuable insights into the tractor's operational context. Leveraging this data, along with predictive analytics and remote monitoring, enables more effective maintenance strategies to be developed. Such solutions reduce operational costs, enhance the reliability of tractor fleets, and ultimately improve productivity in agriculture. The information about weather conditions, climate or air quality could be obtained from meteorological stations, weather satellites, and environmental agencies. Weather conditions have a significant impact on vehicles and their performance. For example, Extreme heat can lead to engine overheating, rain and snow make fields muddy, which increases traction challenges, dust and sand are extremely aggressive for air filters and radiators. Process of combining such data from various spatially and temporally referenced datasets into a consistent unified representation is called fusion. It poses a challenge since the volume of data and count of sensors is growing continuously. Decreased cost of sensor platforms gave rise to the concept of the "Internet of Things.". These IoT devices generate substantial amounts of geographically tagged data. The increase in data volume has led to the development of new computational methods. Various distributed databases and computing platforms have been developed, each with its unique set of trade-offs, and softening constraints of traditional relational database management solutions. However, handling geospatial-temporal data in a distributed mode presents distinctive challenges and considerations.

Analysis of Recent Research and Publications

Existing research [2, 3] compares features of data storages. The determination of the most effective data storage solution for an application involves assessing the performance and scalability of different systems tailored to the specific needs of the application. Benchmarking serves as a valuable method of evaluation, examining systems based on performance metrics such as throughput and latency, system parameters including the number of CPU cores, amount of RAM, and disk space, as well as workload parameters like query density and data access distribution. Additionally, benchmarks offer insights into performance bottlenecks under various workloads [4].

GeoMesa is an open-source toolkit designed to empower extensive geospatial querying and analytical tasks across distributed computing systems. It incorporates geospatial-temporal indexing capabilities to work seamlessly with databases such as Accumulo, HBase, Google Bigtable, and Cassandra, facilitating the storage and management of extensive datasets containing points, lines, and polygons [1]. Existing research provides benchmarks of Accumulo [5] and Cassandra [6] databases without integration with GeoMesa and in different testing environments.

The goal of this article is to benchmark and compare the performance of Accumulo and Cassandra when they are used as underlying data storages for GeoMesa and in identical testing environment conditions to avoid any inconsistencies that can affect the results.

Presenting main material

Performance tests were divided into two groups – writes and reads. Each test thread tried to simulate a real user, so there were random pauses between the execution of tests. 1,5,10,20,30,40 and 50 users that access the database concurrently were simulated. Each test was run for 15 minutes from which around 12-13 minutes were taken to calculate results.

1. For our testing, we generated a dataset of telemetry data, that is produced by agriculture vehicles for predictive maintenance purposes. The following test scenarios for read queries were evaluated:
2. For model X machines return the last known location
3. For radius X return a list of machines with make model year
4. For X field boundaries return the machineID that has executed on this field in the past year
5. For this machineID return the harvest data
6. For this machineID return the chassis data over the past year
7. For machines of model X return a list of machines with their diagnostic trouble code events over the past month
8. For this machineID and this diagnostic trouble code event time return the location of the machine

What machines have experienced overheating events this year

A Cassandra cluster of 6 nodes with different IP addresses was deployed on AWS EC2 instances with 16 CPUs and 64 Gb RAM each. All the nodes are connected in a cluster by installing Cassandra and GeoMesa in all of them and configuring them. Ubuntu 20.04 was used as an operating system. Each Cassandra node used 2 disks: one for OS and second for Cassandra data. An additional 2 CPUs and 7 Gb RAM virtual machine with Ubuntu 20.04 was deployed to be used as a jump host to interact with Cassandra cluster. Figure 1 represents Cassandra cluster deployment infrastructure.

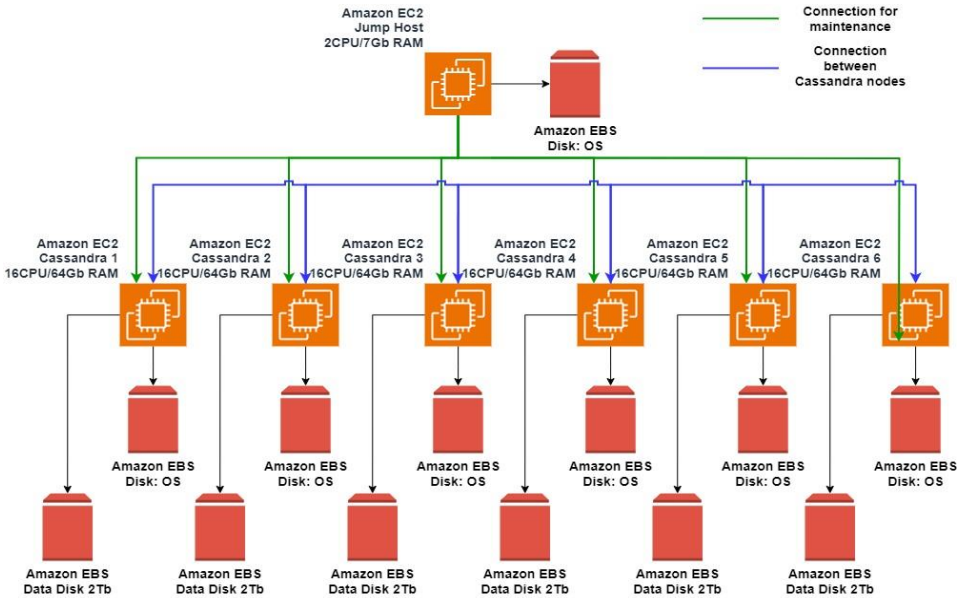

Fig. 1. Cassandra cluster deployment

ISSN 2219-9365

An Accumulo cluster of 7 nodes with different IP addresses was also deployed on separate AWS EC2 instances – 1 master node and 6 slave nodes. The master node required fewer computation resources, so 4 CPUs and 16 Gb RAM virtual machine with CentOS 7.5 was used for it. Each slave node was deployed on virtual machines with 16 CPUs, 64 Gb RAM, and CentOS 7.5. The following services were installed on the master node: HDFS Name node, HDFS Secondary Name node, Zookeeper, YARN Resource Manager, YARN Registry DNS, YARN Timeline Service 1.5, YARN Timeline Service 2.0, Accumulo Master, Accumulo Garbage Collector, Accumulo Tracer, Accumulo Monitoring. Each slave node contained the following services: HDFS Data node, Zookeeper Client, Monitoring Client, Accumulo Table Server, GeoMesa. An additional virtual machine was used to install Ambari – web service to deploy and manage Hadoop cluster. Figure 2 represents Cassandra cluster deployment infrastructure.

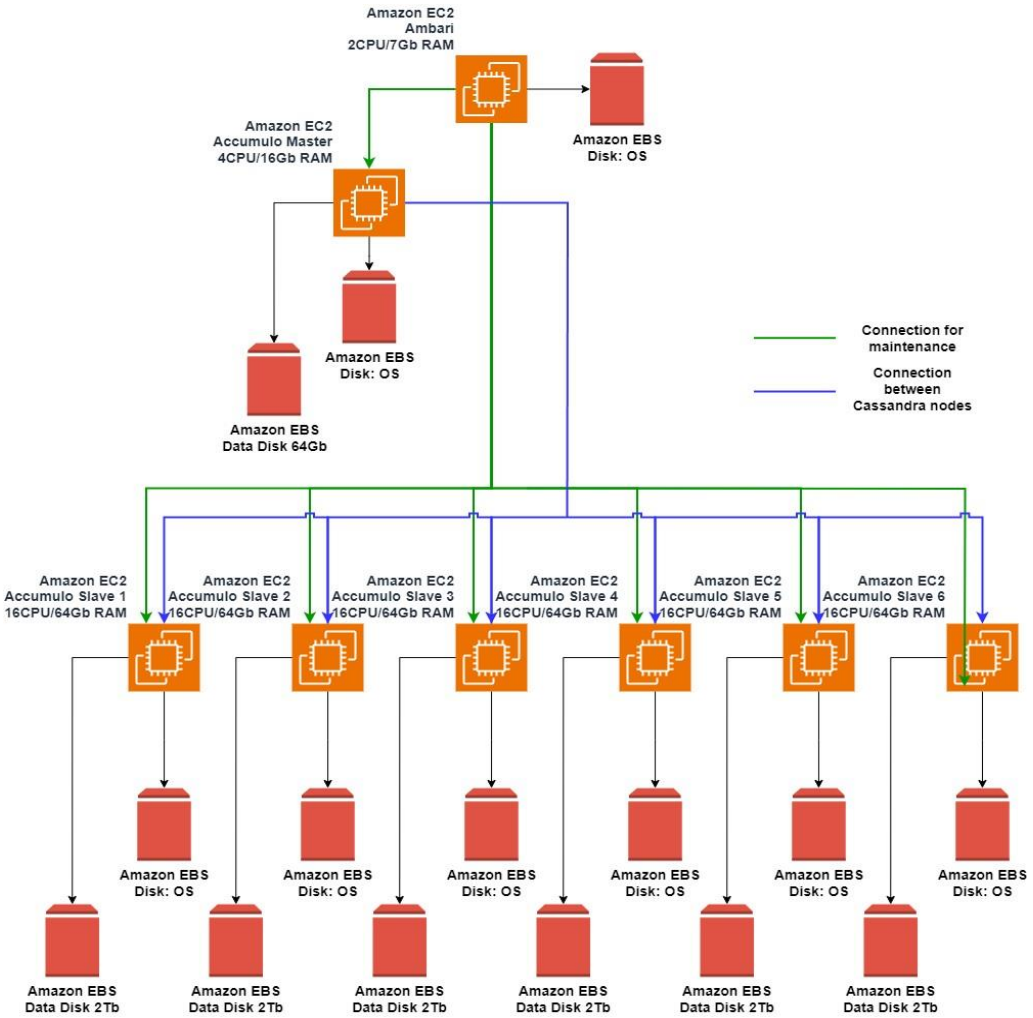

Fig. 2. Accumulo cluster deployment

Performance tests were executed from 2 Amazon EC2 instances with Ubuntu 20.04 using jMeter 5.6, Grafana, Prometheus, and Pandas were used to analyze the results. Before each test, the environment was heated up, to ensure that any initialization processes, caching mechanisms, or resource allocation were settled and consistent and to eliminate the impact of cold starts, focusing on the steady-state performance. Each test was run for 15 minutes from which around 12-13 minutes were taken to calculate results.

Results from tests for 1, 5, and 10 users querying GeoMesa at once favor Cassandra as DB, queries took 20-40ms in average. Times for GeoMesa on Accumulo in the same situation were 50-90ms on average, 120-280% longer than on Cassandra. When tests reached 20 users or more, the situation changed for GeoMesa on Cassandra, the average time to send a query and get a result increased to 155ms with 20 users and even 875ms average with 50 users. On the other hand, GeoMesa on Accumulo, queries took 125ms on average with 50 users. From a user perspective, GeoMesa with Accumulo was faster by 80% on average than Cassandra.

ISSN 2219-9365

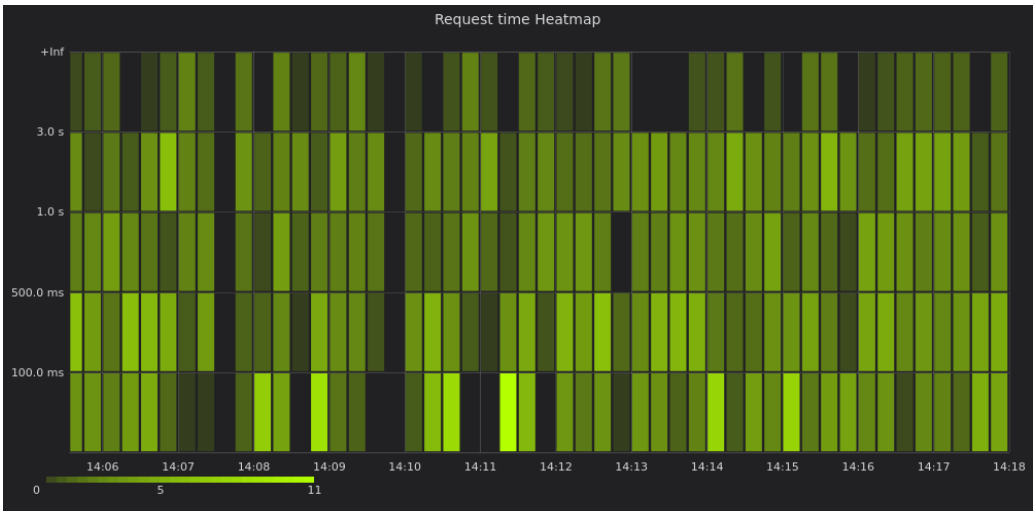

Fig. 3. GeoMesa/Cassandra [50users] reads time buckets

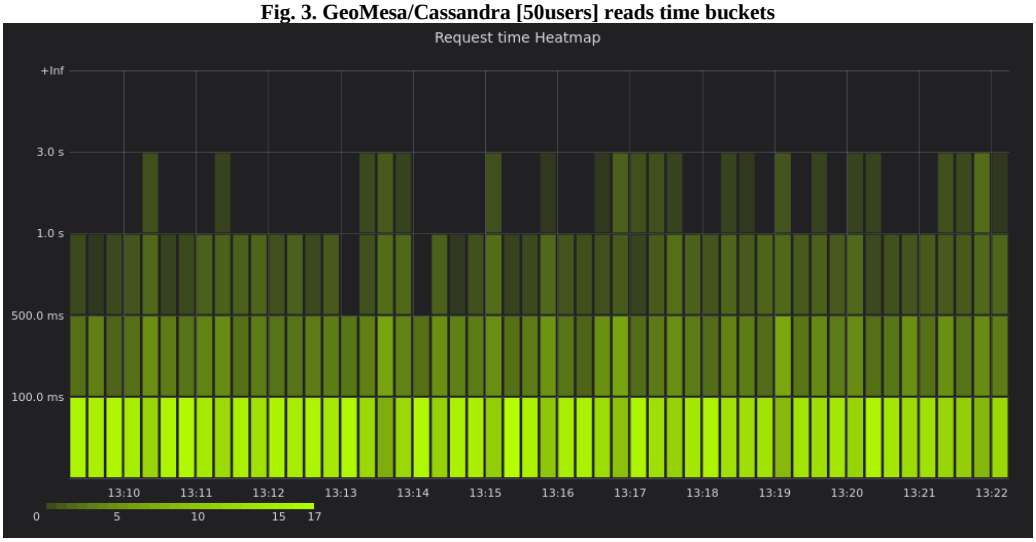

Fig. 4. GeoMesa/Accumulo [50users] reads time buckets

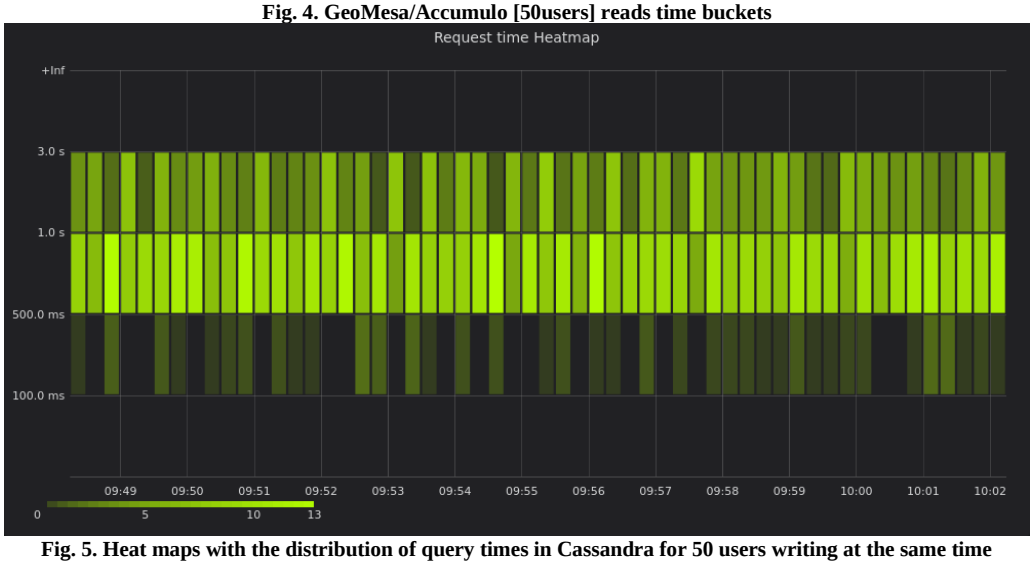

Fig. 5. Heat maps with the distribution of query times in Cassandra for 50 users writing at the same time

For write queries in all tests GeoMesa on Accumulo performed better than GeoMesa on Cassandra by 10-20% with a load of 1 to 50 users. With quicker responses came better throughput where for 50 users at once GeoMesa on Cassandra was able to ingest 22k requests per second on average while GeoMesa on Accumulo 28k requests per second on average.

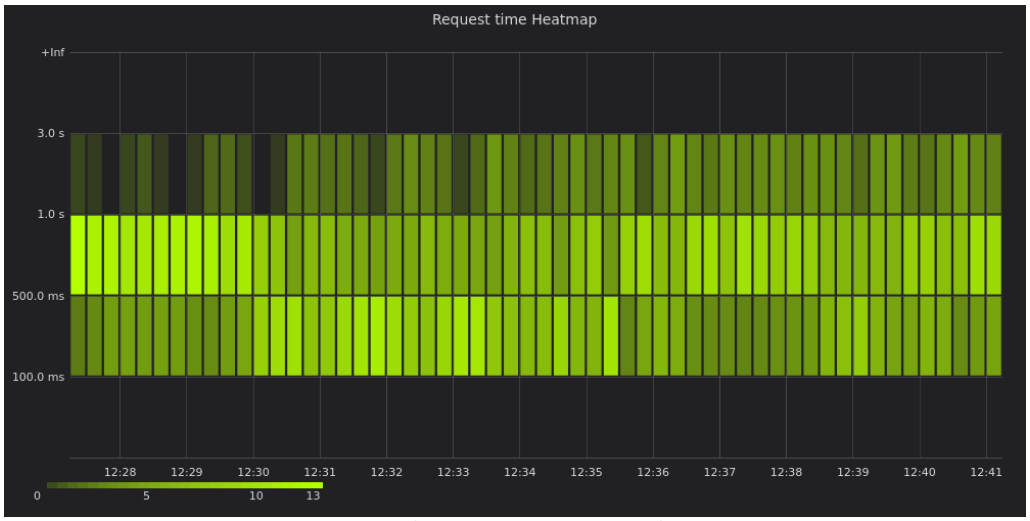

Fig. 6. Heat maps with the distribution of query times in Accumulo for 50 users writing at the same time

Detailed information about resource usage for read and write operations made by different numbers of concurrent users is present in Table 1

Table 1.

Average Cassandra and Accumulo resource usage

	1 user	**5 users**	**10 users**	**50 users**
Cassandra average write query time (ms)	417	432	460	1054
Accumulo average write query time (ms)	301	398	446	810
Cassandra average CPU load (1m)	0.113	0.368	0.647	2.597
Accumulo average CPU load (1m)	0.341	0.881	1.393	4.35
Cassandra average network usage reads IN (Mbps)	19	92	178	844
Accumulo average network usage reads IN (Mbps)	1	4	8	34
Cassandra average network usage reads OUT (Mbps)	40	200	395	1876
Accumulo average network usage reads OUT (Mbps)	14	70	135	620
Cassandra average network usage writes IN (Mbps)	10	50	102	371
Accumulo average network usage writes IN (Mbps)	9	39	76	320
Cassandra average network usage writes OUT (Mbps)	10	43	84	299
Accumulo average network usage writes OUT (Mbps)	6	27	52	220

Conclusions

The executed performance tests for read and write operations prove comparative advantages of two GeoMesa underlying database systems Cassandra and Accumulo. Those systems could be employed to store telemetry data produced by agricultural vehicles for predictive maintenance purposes. Results proves selection of Cassandra when small count of users (1, 5, and 10) are concurrently accessing the GeoMesa to read the data. However, as the number of users accessing the system increased to 20 or more, Cassandra experienced a substantial degradation in performance, while Accumulo maintained a more consistent performance and as a result demonstrated a significant 80% improvement in query response times on average compared to Cassandra under heavy load. For write queries across all user loads (1 to 50 users), Accumulo outperformed Cassandra. Furthermore, Accumulo demonstrated efficiency by requiring less network bandwidth for both read and write operations. However, it's important to note that GeoMesa on Accumulo placed a heavier load on the CPU throughout the tests.

Ultimately, the choice between Cassandra and Accumulo depends on the specific use case and priorities of the system. Cassandra performs admirably with lower user loads for read operations, but its performance deteriorates under heavy concurrent access. Accumulo consistently maintains a good level of performance for both read and write operations, making it a suitable choice for scenarios where high concurrency and reliability are crucial. Accumulo imposes increased CPU load comparing to Cassandra.

The results of these performance tests provide valuable insights for decision-makers, enabling them to make an informed choice based on the unique requirements of their applications and the trade-offs between query response times, write throughput, and resource utilization.

References

1. Fox, A., Eichelberger, C., Hughes, J., and Lyon, S., Spatio-temporal indexing in non-relational distributed databases. Spatio-temporal indexing in non-relational distributed databases. 2013. pp. 291-299. doi: 10.1109/BigData.2013.6691586.
2. Oussous, Ahmed & Benjelloun, Fatima-Zahra & Ait Lahcen, Ayoub & Belfkih, Samir. NoSQL databases for big data. International Journal of Big Data Intelligence Volume 4. 2017. pp. 171-185. doi: 10.1504/IJBDI.2017.085537.

3. Byali, Ramesh. Cassandra is a Better Option for Handling Big Data in a No-SQL Database. International Journal of Research Publication and Reviews Volume 3. 2022. pp. 880-883. doi: 10.55248/gengpi.2022.3.9.27
4. Kim, Suneuy & Kanwar, Yuvraj. GeoYCSB: A Benchmark Framework for the Performance and Scalability Evaluation of NoSQL Databases for Geospatial Workloads. 2019. pp. 3666-3675. doi: 10.1109/BigData47090.2019.9005570.
5. R, Dr & R, Chaitra & P, Archana & M, Bhagyalakshmi. Performance Benchmarking and Metrics Evaluation of Apache Accumulo. International Journal of Advanced Research in Science, Communication and Technology. 2022. pp. 241-245. doi: 10.48175/IJARSCT-5332.
6. Vyas, Kena & Jat, PM. Study of Consistency and Performance Trade-Off in Cassandra. 2022. pp. 61-77. doi: 10.5121/csit.2022.121907.