International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Suitability Of Influxdb Database For Iot Applications

Mohammad Nasar, Mohammad Abu Kausar

Abstract**:** ***Large amounts of data are generated every moment by connected objects creating Internet of Things (IoT). IoT isn't about things; it's about the data those things create and collect. Organizations rely on this data to provide better user experiences, to make smarter business decisions, and ultimately fuel their growth. However, none of this is possible without a reliable database that is able to handle the massive amounts of data generated by IoT devices. Relational databases are known for being flexible, easy to work with, and mature but they aren't particularly known for is scale, which prompted the creation of NoSQL databases. Another thing to note is that IoT data is time-series in nature. In this paper we are discussed and compare about top five time-series database like InfluxDB, Kdb+, Graphite, Prometheus and RRDtool.***

Index Terms**:** ***Internet of Things, NoSQL databases, InfluxDB, Kdb+, Graphite, Prometheus, RRDtool.***

I. INTRODUCTION

The word IoT (Internet of Things) usually relates to the situations where connectivity of network and computational ability expanded to sensors, objects, and everyday usable stuffs usually not considered computer systems, enabling devices to produce, transfer and put away information with less human interference. [16].

The fundamental concept of IoT is the occurrence of a multitude of item or items surrounding it, for instance Radio-Frequency Identification (RFID) tags, actuators, sensors, mobile phones, and so forth that can communicate to achieve some general goal [8]. The number of growing "Things" nearby us indicates IoT's effects on our daily lives. In 2019, 14.2 billion "Things" were connected as said by Gartner, an American research and consulting firm with a figure of 25 billion by 2021 [1]. These "things" produce a big data quantity. IoT applications therefore it requires a more flexible database schema for high availability and flexibility to readily scale up to satisfy growing requirements.

Relational databases can deal with immense volume of information, however they have real disadvantages, for example, scaling up is substantially very difficult than just including new hardware, the expense related with scaling is exceptionally increasing [23]. So as to discuss these restrictions, NoSQL databases are progressively utilized with the reason to deal with tremendous amount of data, and give auto scaling, superior execution, and high accessibility [24]. Though, there are numerous NoSQL databases, and each database has its very own execution qualities, strategy of data distribution, Query language, etc. This will make the Superior database option for more difficult applications. Hence, look into the characteristics of NoSQL databases and specifications for IoT data management are essential for selecting the appropriate database for IoT applications.

II. INTERNET OF THINGS (IOT)

The word Internet of Things (IoT) relates to a large and diverse network of physical and virtual elements integrated in sensors, software, electronics, and connectivity to allow objects to attain higher value and service by exchanging information over the Internet with other linked objects. [14]. Things ' in terms of IoT can be a man with a heart rate monitor implant, a farm pet with a biosensor transmitter, a field operating robot assisting in any kind of disaster, or any other device that can be allocated an IP address is able to transfer information and interact within the current Internet infrastructure [15].

The Internet of Things importance is not that more and more devices, individuals and systems are being' linked' to each other. It is through advance and innovative apps that the data produced from these items are shared, analyzed and processed, implementing entirely fresh analytical methods and considerably modified timeframes. The Internet of Things will drive Big Data in real-time, offering more data from many distinct sources, and allowing us to obtain entirely fresh views on the surrounding environment.

The world of data and apps in the Internet of Things needs considerably more flexibility, agility and scalability. A restricted amount of fundamental needs drive these demands.

Gartner forecast that 14.2 billion linked objects are equipped in 2019 [1], reaching a total of 25 billion by 2021 and facilitating the information used to evaluate, predict, handle as well as create smart choices independently. The Internet of Things introduces extremely new difficulties, particularly with regard to database management systems (DBMS), such as incorporating tons of enormous information into real-time, processing activities as they flow, and addressing data security. An instance might be IoT-based ambient humidity sensors or air conditioner sensor installed in different smart cities, which in just a few minutes generate large quantity of data on the humidity and temperature of the live atmosphere. In this regard, we could see that technology is already benefiting from multiple architectural types of IoT in several industries, such as: smart city, transportation, intelligent health, assisted living, e governance, e-education, logistics, retail,

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Mohammad Nasar, Department of Computing and Informatics, Mazoon College, Muscat, Sultanate of Oman.
Mohammad Abu Kausar, Department of Information Systems, University of Nizwa, Nizwa, Sultanate of Oman.

agriculture, industrial production, automation, and other business management. [2, 3, 4].

The IoT seeks to be using smart technologies by linking stuff to achieve anything anywhere at any moment. The idea of IoT came into being in late 90's and Kevin Ashton launched the word Internet of Things in 1999[5]. IoT basically enables an autonomous yet safe link for the communication of real-world working devices. [6]. by automating daily tasks [7], the IoT decreases physical job. The objects linked to the Internet are quickly increasing. The smart phones incorporate multiple sensors and actuators that senses the information, calculates the data and transmits the precious data collected using the Internet. [8]. By using this network with different appliances that incorporate the sensors, we would be able to develop multiple new apps that will impress the world [9].

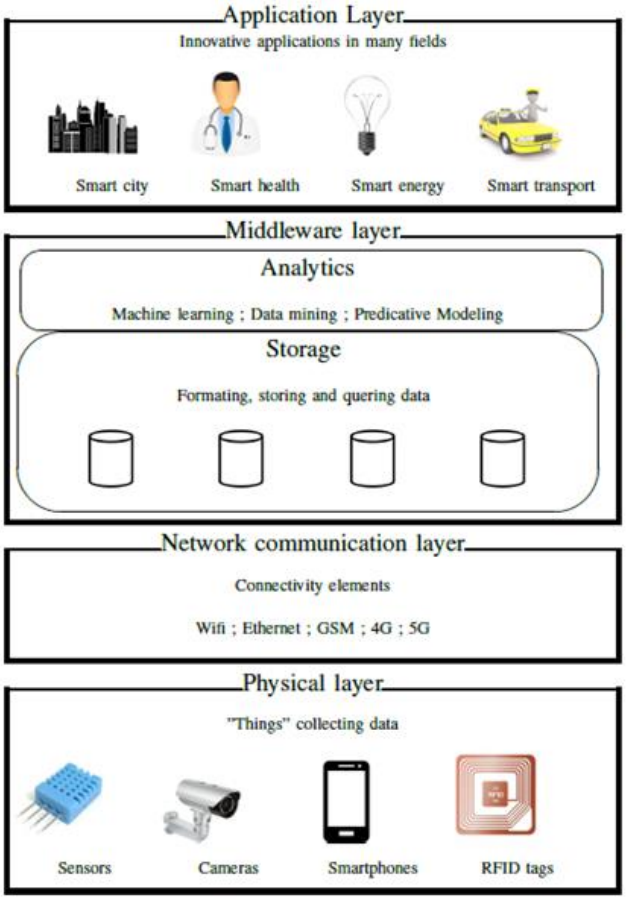

Figure 1: Internet of Things architecture

In this context, IoT is composed of four layers, as shown in Fig 1:

- The physical layer: containing " Things " that generate data
- The communication layer: playing the bridge role between the"Things" , as well as between the physical layer and the middleware layer by transmitting data.
- The middleware layer: storing data generated by" Things" and analyzing them in order to extract useful information's.
- The application layer: taking advantages of the extracted information by creating smart applications.

In this work, we are interested in the middleware layer, especially in data management. In the next section, we present data management approaches in some IoT applications.

III. NOSQL DATABASES

To manage IoT data efficiently, finding the appropriate type of database is very crucial. But it might be challenging task to choose an optimal database for IoT applications, as the IoT environment is often not the same. There are several aspects to take into account when selecting a database for IoT apps. The most significant of those are scalability, capacity to manage large quantities of data at appropriate speeds, versatile schema, and convenience with a variety of analytical instruments, safety and expenses.

Answering to the Internet of Things with Relational Database Management Systems (RDBMS) is an alternative but it provides a restricting aspect which in time becomes a major barrier to the implementation of the complete possibilities feasible from all information types. The Internet of Things databases involve the flexibility of the NoSQL strategy, enabling various kinds of data to be stored as discussed previously, but more effectively, the agility and efficiency to cater the fundamental data models to new and evolving company demands and apps. [10]

In the 60's, database systems started gaining ground. Various kinds were created, each by using its information representation scheme. Initially set as topographical databases depending on linked lists, later transformed into related databases with attachments, triggers, features, stored procedures and object-oriented features. NoSQL appeared in the early 2000s and has become a common trend. Today's most frequently used database formulations are based on the relational model using SQL as its query language.

Conventional RDBMS systems may continue to play a vital role throughout the IoT development, extremely structured data sets are produced and processed from a massive amount of business IT systems and maintained in a comparatively isolated way. Databases will involve different levels of flexibility, scalability and agility when it comes to managing more heterogeneous information generated by millions and millions of sensors, gateways, and devices each with its own data structures and potentially connected and integrated over many years.

A. Database Requirements for IoT

IoT's advantages rely strongly on the large volumes of data created by apps. This introduces a number of problems for the application's database management system, most importantly in terms of scalability and the capacity to ingest data quickly.

Data sent from devices are often very diverse; therefore the ability to manage rapidly evolving data also helps to store IoT data [11]. There are several parameters influencing how often these demands can be handled by a database. Scalability is primarily limited by the sharding and replication techniques used for partitioning and distributing data in a cluster, the volume of data produced.

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The overall throughput of a database system relies on several aspects like scalability, consistency requirements, and data model wealth, index utilization, fundamental systems (e.g. storage engine and programming language) and hardware. It would be essential that the database needed to power an IoT request is designed with these variables in mind to meet the criteria without compromising.

B. NoSQL for IoT

As discussed in Section II (IoT), Scalability and the capacity to quickly consume data are the main database requirements for IoT apps. NoSQL systems are ideal for IoT since they are designed with significant horizontal scalability. Some other popular characteristic of NoSQL databases is the effective use of data storage in memory, which would be highly useful for writing throughput and latency. RDBMS systems appear to be incomplete because they were not intended to process the quantity of data or the rate produced by the data [11, 12]. A schema may be configured easily by introducing a new field to a database or adding a unique column group to a table, making it easy to manage information from IoT apps that change quickly [11]

NoSQL database solutions, however, are becoming more common as huge quantities of swiftly increasing unstructured data are stored, exceeding the efficiency and scalability limitations of rigid relational databases. That raised the question whether it came to the dawn of the relational model. On the one side, relational databases, following the laws of normalization, use normality forms on the concept of information divided into field records and tables. On the other side, for the normality and re-design of scalable facilities, NoSQL databases escape continues to deliver a solid performance alternative.

IV. RAISING A NEW GENERATION ON NOSQL

Volume matters in the Internet of Things, of course. No article on the topic is comprehensive without mentioning "50 billion devices" or any other dreamy predicted.

However for me, more important number is a million issues. When the device counts balloon, NoSQL-savvy developers ' ranks must expand to meet the growing demand for apps. As per the projections by VisionMobile, today there are only 300,000 developers of the Internet of Things, but this amount will explode by 2020:

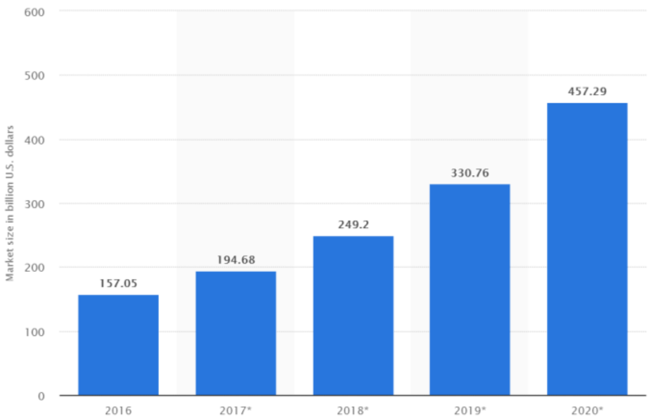

Figure 2: Size of the IoT market worldwide from 2016 to 2020 (in billion U.S. dollars) [Source: https://www.statista.com/statistics/764051/iot-market-size-worldwide/]

The complete Internet of Things (IoT) connected devices installed is expected to reach 75.44 billion globally by 2025, a five-fold increase in ten years. The IoT, supported by the omnipresent Communication technology, is the next important step in providing the commitment of the Internet to make the world a connected place.

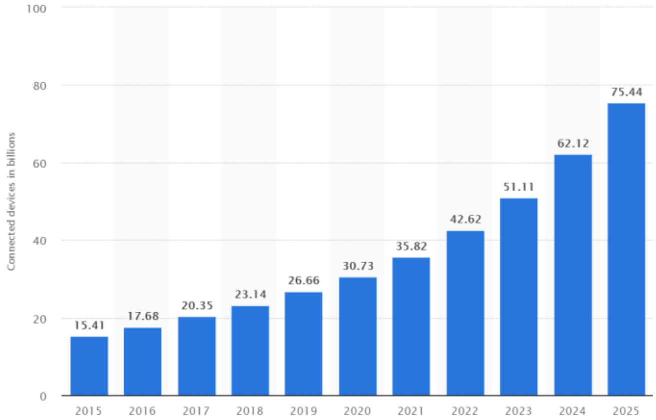

Figure 3: Internet of Things (IoT) connected devices installed worldwide from 2015 to 2025 (in billions) [Source: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/]

V. TIME SERIES DATABASE

A time series database (TSDB) is a time-stamped or time series data optimized database. Data from time series are merely measures or occurrences which are controlled, down sampled and indexed over period. This may include server statistics, tracking of system performance, network information, sensor information, occurrences, clicks, market trades, and many other analytics data types.

A time series database is specific designed to handle time-stamped metrics and occurrences or measurements. A TSDB is designed over time to measure change. Characteristics that allow time series data very distinct from other information workloads are multiple records information lifecycle management, summary, and wide range data scans [17].

A. Time Series in NoSQL Database

As one of the fundamental principles, NoSQL declared a universal data model to be rejected. The data model has to satisfy the handling techniques needed. The second fundamental principle is the absence of specific instruments (layers) for programming access. The Data Access API is component of NoSQL schemes and provides some of the key components for the underlying data model choice. We will address top 5 time series NoSQL database schemes in next section.

B. Top Time Series Databases

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Several TSDBs and their rankings is shown in Figure 4 of Time Series DBMS ' DB-Engines Ranking. DB-Engines[13] is an integrated website that ranks search engine-based databases, social media references, amount of work offers, and frequency of technical debate. As of June 2019, Influx DB is ranked number one on this list. Figure 2 also demonstrates the chronological modifications in such databases but it can be seen that, in June 2018, InfluxDB also was the top TSDB and retained its rank.

Rank			DBMS	Database Model	Score		
Jun 2019	May 2019	Jun 2018			Jun 2019	May 2019	Jun 2018
1.	1.	1.	InfluxDB	Time Series	17.98	-0.09	+6.65
2.	2.	2.	Kdb+	Time Series, Multi-model	5.80	+0.21	+2.79
3.	3.	↑ 4.	Graphite	Time Series	3.33	+0.10	+0.95
4.	4.	↑ 6.	Prometheus	Time Series	3.32	+0.21	+2.06
5.	5.	↓ 3.	RRDtool	Time Series	2.67	-0.23	0.00
6.	6.	↓ 5.	OpenTSDB	Time Series	2.24	-0.23	+0.68
7.	7.	7.	Druid	Multi-model	1.78	+0.09	+0.65
8.	8.	↑ 17.	TimescaleDB	Time Series, Multi-model	1.11	-0.05	+1.06
9.	9.	↓ 8.	KairosDB	Time Series	0.50	-0.04	+0.09
10.	10.	↓ 9.	eXtremeDB	Multi-model	0.41	+0.03	+0.13

Figure 4: DB-Engines Ranking of Time Series DBMS [Source: https://db-engines.com/en/ranking/time+series+dbms]

Now we are going to discuss about above top five NoSql Time series database in details.

1. **Influx DB**: Influx DB is an open source distributed database of time series created in 2013 by InfluxData. It is developed using Go programming and is dependent on LevelDB, a database of key value. In conjunction from front-end, clients are supplied with an HTTP api and libraries to work together with the database. The primary benefit of InfluxDB has been its ability to aggregate values without any manual interference in moment buckets on - the-fly. Software such as Grafana, a strong front-end tool that provides visualization characteristics for time series data, can access InfluxDB [19]. InfluxDB will have no internal incompatibilities and SQL is used to query a data structure that includes measurements, series and points. Each point comprises of diverse pairs of key-value named fieldset and timestamp. Values may be integers of 64-bit, 64-bit floating points, Booleans and strings. Points are indexed according to their tagset and moment. InfluxDB uses TCP, HTTP, and UDP to store information.

Architecture of InfluxDB

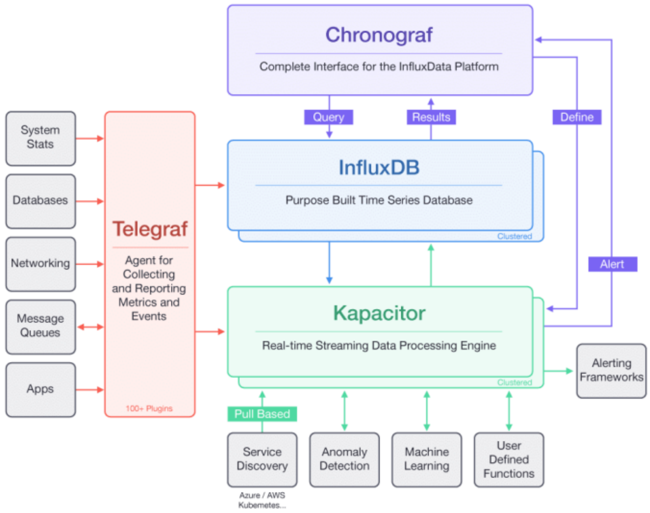

Figure 5: Architecture of InfluxDB [Source: https://www.influxdata.com/time-series-platform/].

In this context, Architecture of InfluxDB composed of four layers, as shown in Fig 5

Telegraf: Telegraf is the server-driven open source agent for gathering and reporting metrics from InfluxData.

InfluxDB: InfluxDB is the open source time series database of InfluxData intended to manage heavy loads of write and query.

Chronograf: Chronograf is the open source web application of InfluxData for tracking infrastructure, alert management, and visualization of information and management of databases.

Telegraf: Telegraf is the server-driven open source agent for gathering and reporting metrics from InfluxData.

InfluxDB: InfluxDB is the open source time series database of InfluxData intended to manage heavy loads of write and query.

Chronograf: Chronograf is the open source web application of InfluxData for tracking infrastructure, alert management, and

visualization of information and management of databases.

Kapacitor: Kapacitor is the open source information handling framework for InfluxData, which makes alerts simple to generate, operate ETL tasks, and identify anomalies. Kapacitor is the open source information handling framework for InfluxData, which makes alerts simple to generate, operate ETL tasks, and identify anomalies.

Features:

- High quality custom information storage published explicitly of information from time series. The TSM motor enables elevated intake velocity and consolidation of information
- Written completely in Go. It publishes without internal dependencies into a single binary.
- Easy, high-performance HTTP APIs writes and query.
- Help plugins for many other data intake protocols like Graphite, Collectd and OpenTSDB.
- SQL-like expressive query language customized to query

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

aggregated information readily.

- Tags enable the indexing of sequence for quick and effective queries.
- Retention of stale information effectively ends automatically.
- Steady queries calculate aggregate information instantly to create more effective frequent queries.

2. **KDB+:** Kdb+ database developed in2003, kdb+ is a commercial column-based relational time-series database with in-memory abilities. The column-based database architecture is primarily useful when performing aggregate queries like, sum, count, avg etc. Backing up this database, is its Q query language. According to its developers [18], the primary design objectives of Q are efficiency, speed and expressiveness.

Features:

- Column-oriented: The tables in kdb+ are stored as columns and apply an operation to an entire column vector. This is especially useful when you have billions of records and want to aggregate based on columns which is a general case in financial data analysis.
- In Memory database: One of the important features that give speed to kdb+ is it's in memory storage and manipulation of the data. Since the tables are stored in memory and data manipulation is performed in memory with Q, there is no need for stored procedure language. In memory storage makes it really fast but it requires a lot of RAM which is no longer a major problem as servers with high RAM is now inexpensive.
- Ordered lists: In kdb+ database, all the rows are stored in a specific order as ordered lists are the foundation of all data structures. This makes processing the large amount of data very fast and efficient.

3. Graphite:

Graphite [20] is a graphing scheme that is extremely configurable in real time. As a client, you write an application that gathers numerical time series data you were concerned in graphing and bring it to the processing backend of Graphite, carbon, which stores the data in the specific database of Graphite. The information can then be viewed via the internet interfaces of the graphite.

Features:

- Decrease significantly programming time with our easy-to-use, drag-and-drop Crimson 3 software
- Graphite's sleek bezel enables clients can save actual property panels or enclosures bringing in actual dollar savings.
- Rugged building offers a broad variety of temperature and shock and vibration requirements for deployment in hostile settings
- Graphite can simply accept prompt inputs from different devices using modular capabilities or to provide complete PID control.

4. Prometheus:

Prometheus [21] is an open-source toolkit that was initially constructed at Sound Cloud to track and notify systems. After its founding in 2012, Prometheus has been embraced by many businesses and Organizations as well as a very active group of designers and customers are involved in the project. It became a project of open source in its own right and is maintained separately of any business.

Features:

- A multi-dimensional information model in which time series information is characterized by metric name and key / value size;
- Independent individual server devices without any reliance on distributed storage;
- Data gathering over HTTP through a pull model;
- Time series information driven through an intermediate portal to other information locations and shops;
- Targets found through service discovery or static configuration;

RRDTool:

RRDtool (round-robin database tool) is intended to manage information from time series such as bandwidth utilization, temperature, or CPU load. The data is collected in a database based on a circular buffer, so the footprint of the system storage stays continuous over time.

It also involves tools that were initially designed to obtain round-robin information in a graphical format. For many programming languages, bindings exist, e.g. Perl, Ruby, Python, Tcl, PHP, Lua. There is a complete autonomous application of Java called rrd4j.

Features:

- Round-Robin Database tool (RRD tool) handles time-series data like temperatures, CPU load, network bandwidth etc.
- RRDtool is open source tool and it can be integrated into Shell scripts, Python, Ruby, Perl, Lua or Tcl applications easily
- Another important feature of RRDtool is its ability to create graphs.

Comparison between Top 5 Time Series NoSQL databases is given on Table 1:

VI. POPULARITY

TSDBs are obviously responsible for managing time series data, but with the advent of the Internet of Things (IoT), their popularity appears to have increased. IoT is a connectivity network of physical devices / objects that allows information to be exchanged and collected. These techniques generate big amounts of information that are generally time-stamped, and with the increasing popularity of IoT, the popularity of TSDBs has risen much more, so they can be used to store information from sensors and devices in this domain effectively. Other popular uses of TSDBs include tracking DevOps and analyzing real-time information. These days, numerous big businesses such as Facebook, eBay etc. use TSDBs instead of relational databases for information

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

surveillance purposes in particular. Based on the graph in Figure 1a it is shown that over the previous couple of years the popularity of TSDBs has increased quickly. The popularity of TSDBs improved by 30 percent from July 2018 to May 2019, which is twice as much as that of Multivalue database management systems, which is second in the list. TSDBs ' popularity continues to grow until now. In Figure 1bwe also visualize the rise in IoT popularity beginning in 2015 to offer an understanding of the concurrent growth of the two areas.

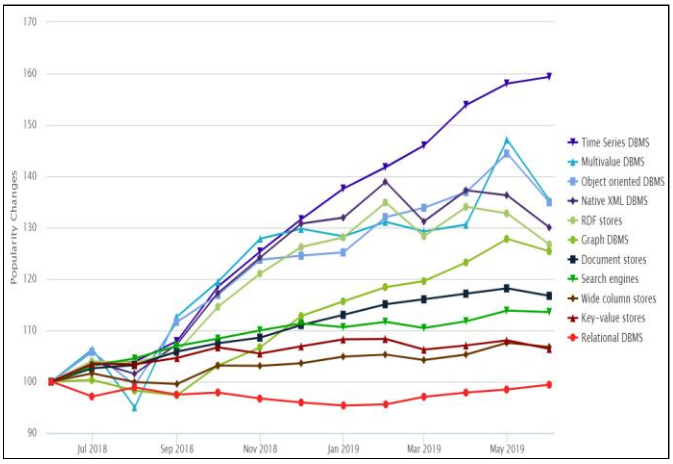

Figure 6: DB-Engines Ranking, Popularity Trend [Source: https://db-engines.com/en/ranking_categories]

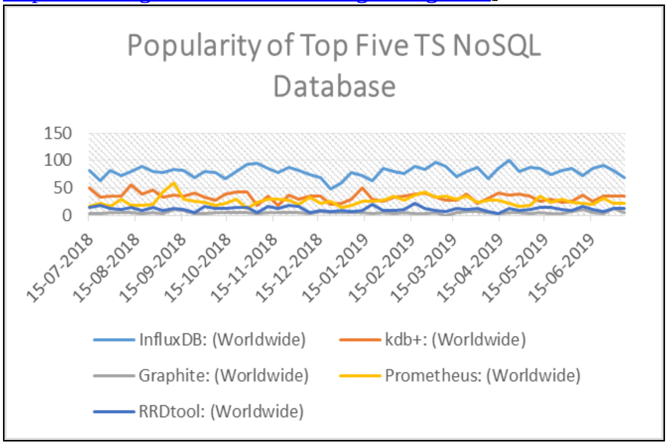

Figure 7: Popularity Of Top Five Time Series Nosql Database.

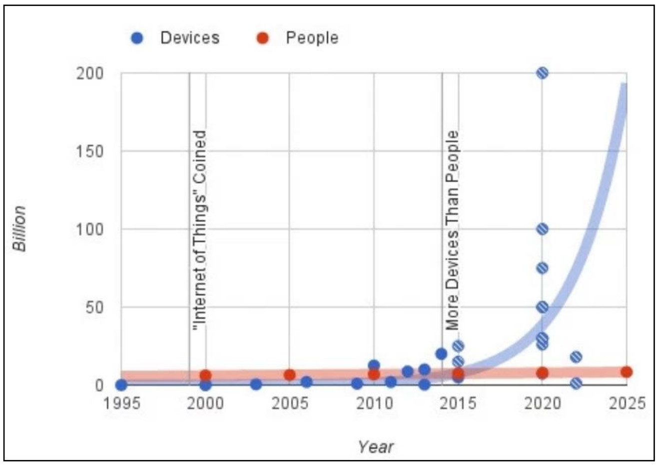

Figure 8: IoT trendline, Brookings [Source:https://www.brookings.edu/blog/techtank/2015/06/09/sketching-out-the-internet-of-things-trendline/]

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table 1: Comparison between Top 5 Time Series NoSQL databases

Name	InfluxDB	Kdb+	Graphite	Prometheus	RRDtool
Narrative	This database is used to store time series data, metrics and events	High performance database for Time Series	This database is use to graphing tool for time series data and data logging	monitoring system and Open-source Time Series database	round-robin database for graphing tool for time series data and data logging
database model (primary)	Time Series database	Time Series database	Time Series database	Time Series database	Time Series database
Initial release	2013	2000	2006	2015	1999
License	Open Source	commercial	Open Source	Open Source	Open Source
Language used to develop	Go	Q	Python	Go	C
Server operating systems	Linux, OS X	Linux,OS X,Solaris, Windows	Linux,Unix	Linux,Windo ws	HP-UX, Linux
SQL	SQL-like query language	SQL-like query language (q)	No	No	No
APIs	HTTP API JSON	HTTP API, Jupyter, Kafka, WebSocket, JDBC, ODBC	HTTP API Sockets	RESTful HTTP/JSON API	in-process shared library, Pipes
supported programmin g languages	Lisp ,Perl, PHP, Python, R, Ruby, .Net, Erlang, Go,Java, JavaScript, JavaScript (Node.js)	MatLab, Perl, PHP, Python, R, Scala, C, C#, C++, Go, Java, JavaScript, Lua	Python, (Node.js), JavaScript	JavaScript, .Net, C++, Go, Haskell, Java (Node.js)	C,C#,Java,JavaS cript (Node.js),Lua,P erl,PHP,Python, Ruby
Method of Partitioning	Sharding	horizontal partitioning	None	Sharding	None
Method of Replication	selectable replication factor	Master-slave replication	None	Yes	None
MapReduce	No	No	No	No	No
Concurrency	Yes	Yes	Yes	None	Yes
In-memory capabilities	yes	Yes		No	Yes
User concepts	simple management of rights via user accounts	management of rights via user accounts	No	No	No

VII. CONCLUSION

NoSQL databases are becoming progressively significant part of the database that can offer real benefits. The purpose of this article was to investigate how different NoSQL database can effectively handle a heterogeneous and large amount of Internet-of-Things data, in order to meet the increasing demands on both load and performance. In this article, we first give a data requirements for IoT applications. We propose then a comparative study of top five of the most popular Time series database of NoSQL in accordance with this requirement. We discuss about popularity of Time Series database from July 2018 to May 2019. We also discuss about current Popularity of Top five Time Series NoSQL database at last we compare the top five NoSQL databases with many different features.

REFERENCES

1. https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends
2. Gubbia, J., Buyya, R., Marusica, B.S., Palaniswamia, M., 2013. Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput Syst. 29 (7), 1645–1660.
3. Miorandi, D., Sicari, S., Chlamtac, I., 2012. Internet of things: vision, applications and research challenges. Ad Hoc Netw. 10 (7), 1497–1516.
4. Giusto, D., Iera, A., Morabito, G., Atzori, L. (Eds.), 2010. The Internet of Things. Springer.
5. Ashton, Kevin (2009), "That 'internet of things' thing." RFID journal 22(7):97-114.
6. T. Fan and Y. Chen. (2010) "A Scheme of Data Management in the Internet of Things," 2nd IEEE International Conference on Network Infrastructure and Digital Content.
7. Y. Huang and G. Li. (2010) "A Semantic Analysis for Internet of Things," International Conference on Intelligent Computation Technology and Automation (ICICTA).

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

8. Q. Zhou and J. Zhang. (2011) "Research Prospect of Internet of Things Geography," 19th International Conference on Geoinformatics.
9. J. Li, Z. Huang, and X. Wang. (2011) "Countermeasure Research about Developing Internet of Things Economy," International Conference on E -Business and E -Government (ICEE).
10. Why NoSQL databases are needed for the Internet of Things
11. MongoDB. Internet of Things. url: https : / / www . mongodb .com / use - cases/internet-of-things (visited on 04/12/2017).
12. Dan Sullivan, James Sulllivan. Find the IoT database that best fits your enterprise's needs (Source: https://internetofthingsagenda.techtarget.com/feature/Find-the-IoT-database-that-best-fits-your-enterprises-needs)
13. DB-Engines. Db-engines ranking, 2019. (Source: https://db-engines.com/en/ranking/time+series+dbms)
14. Adrian McEwen & Hakim Cassimally (2013): Designing the Internet of Internet, first edition. John Wiley and Sons, West Sussex, PO19 8SQ, United Kingdom.
15. John A. Stankovic (2014): Research Directions for the Internet of Things. IEEE Internet of Things Journal 1(1), p. 3–9, doi:10.1109/jiot.2014.2312291.
16. Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. The Internet Society (ISOC), 1-50.
17. https://www.influxdata.com/time-series-database/
18. Kx Systems. The 21st Century Time-series Database. Tech. rep. Kx Systems, 2010
19. https://kx.com/discover/#time-series-database
20. https://graphite.readthedocs.io/
21. https://prometheus.io
22. Chooruang, K., & Meekul, K. (2018, November). Design of an IoT Energy Monitoring System. In 2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE) (pp. 1-4). IEEE.
23. Franck Michel. Integrating heterogeneous data sources in the Web of data. PhD thesis, Universit´e Cˆote d'Azur, 2017.
24. Gunasekaran Manogaran, Daphne Lopez, Chandu Thota, Kaja M Abbas, Saumyadipta Pyne, and Revathi Sundarasekar. Big data analytics in healthcare internet of things. In Innovative healthcare systems for the 21st century, pages 263–284. Springer, 2017.

AUTHORS PROFILE

Dr. Mohammad Nasar is working as Assistant Professor in Department of computing and Informatics, Mazoon College, Muscat, Sultanate of oman. He is having more than ten years of experience in Teaching and Research. He has published several research papers in different international Journals of repute.

Dr. Mohammad Abu Kausar is working as Assistant Professor in Department of Information Systems at University of Nizwa, Sultanate of Oman. He is having more than nine years of teaching, research and Software Development experience. He has published several research papers in various International journals of repute.

Retrieval Number: J92250881019/19©BEIESP
DOI: 10.35940/ijitee.J9225.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)