Analysis of the Effectiveness of NoSQL Solutions for Big Data Processing

Andrey Vlasov, Georgy Biryukov and Pavel Repnikov

Bauman Moscow State Technical University, 5/1 Baumanskaya 2-ya st., Moscow, 105005, Russia

Abstract

The paper discusses a visual technique for the cluster and criterial analysis of the performance indicators of NoSQL solutions based on a property space model implemented in the form of a visual cobweb model. An approach to assessing the performance indicators of data management systems is presented. The analysis of the development trend of promising data storage architectures and position of the NoSQL solutions is carried out. The main significant factors for the construction of the efficiency matrix are formalized. The types of the NoSQL solutions, their main advantages and disadvantages are analyzed, and recommendations for their use are given. The effectiveness of the NoSQL solutions depending on their type is estimated. As a result of the analysis, it is shown that it is advisable to use the NoSQL solutions when processing a large amount of semi-structured and unstructured data (Big Data) in a distributed system. The proposed method for assessing the efficiency based on the property space makes it possible to evaluate the considered solutions according to a set of criteria: volume, complexity, clustering, encapsulation, interface, and CAP indicators.

Keywords

NoSQL, big data, efficiency, criteria, digital transformation, industry 4.0, clustering

1. Introduction

In the context of digital transformation of industry, the effectiveness of the activities of an organization largely depends on the effectiveness of the data management system. With the further introduction of the Industry 4.0 paradigm and cyber-physical systems, changes in the approach to organizing data processing may be a key to ensure the competitive advantages of the organization [1]. The concepts of Industrial Internet of Things (Industrial IoT) and SmartFactory are an integral part of the Industry 4.0 paradigm, which implies a further increase in computing resources at each level of digital transformation of industry [2–4]. A more comprehensive use of data in the "data-driven" approach in decision-making not only for the implementation of direct control but also for solving long-term problems [5, 6] is also associated with this paradigm. Two main architecture models are used in the Industry 4.0 paradigm: RAMI 4.0 (Reference Architectural Model Industry 4.0) developed by the Industry 4.0 working groups and IIRA (Industrial internet reference architecture) developed by the Industrial Internet Consortium [5, 6]. RAMI 4.0 and IIRA are similar in structure and share a common goal of providing hardware and software convergence. The Industrial IoT in the Industry 4.0 paradigm defines the presence of a large number of interconnections between data acquisition devices and devices implementing control [5, 6].

Currently, data management technology is developing under the pressure of the "cloud computing" paradigm, which involves the use of a large number of processors and machines working in parallel to solve big data processing problems [7-11]. This paradigm leads to the idea of building data centers by combining a large number of low-cost storage methods instead of fewer high-performance servers.

SibDATA 2021: The 2nd Siberian Scientific Workshop on Data Analysis Technologies with Applications 2021, June 25, 2021, Krasnoyarsk, Russia
EMAIL: vlasov.a.i@ymservices.ru (A. Vlasov)
ORCID: 0000-0001-5581-4982 (A. Vlasov); 0000-0003-2088-6809 (G. Biryukov); 0000-0003-0626-1263 (P. Repnikov)

The main problems of modern data storage systems are analyzed [12]. One of the main problems is the discrepancy between an object and relational models (Object-relational impedancemismatch), where the impedancemismatch (voltage mismatch) term is borrowed from electronics [12]. This discrepancy lies in the difficulties of using relational DBMS in software systems created utilizing object-oriented design (OOD). Another problem is the differences in data types [12]. One of the main obstacles to displaying the two data models is the type system mismatch. The relational model strictly prohibits the use of pointers and scalar types, and the semantics of their operators also cause problems. Still another problem is the differences in concurrent access and transaction models. The smallest unit of work in a database – a transaction – is a much larger operation than any OOD operation. The final problem to be mentioned is the problem of mapping, which has four aspects: the mapping of structures, constraints, operations, and databases [13-17].

Recently, NoSQL data storage systems have become more widespread. They do not use a relational model. Most implementations of the NoSQL solutions are distributed as Open-source [18]. One of the common properties of the NoSQL solutions is the focus on data aggregation [19-23]. The NoSQL solutions are distributed non-relational databases designed for storing large amounts of data and their massively parallel processing on a large number of typical servers [23]. Standard web applications are very flexible; they contain texts, comments, images, videos, source code, etc. Therefore, the underlying databases of such applications must also be flexible [16]. The NoSQL systems have demonstrated the ability to store and index arbitrarily large datasets while providing a large number of concurrent user queries [23].

How to evaluate the effectiveness of a NoSQL solution for a specific application? This question is not unambiguous and requires either a significant amount of experimental research or the involvement of expert groups. The effectiveness of the NoSQL solution will be understood as a comprehensive indicator which provides a generalized evaluation of the solution. When evaluating the effectiveness, attention should be paid to the coverage of the groups of criteria by the NoSQL functionality: Volume – Complexity – Clustering – Encapsulation – Interfaces. To systematize them, the visual technique of cluster and criterial analysis of performance indicators based on the property space model implemented in the form of a visual cobweb model is used [25].

2. Literature review

The development trends of promising storage architectures and position of the NoSQL solutions are analyzed. Data storage systems can be divided into two large groups: relational and non-relational [12, 13, 17]. The latter are actively developing, and their types will be discussed in more detail.

Key-Value Storage (KVS) storage systems are designed according to the principle of storing key-value pairs [22]. For the database itself, the content of the value is opaque, i.e. working with the values is organized only with the help of keys, and the values themselves are not visible. The examples of implementation are DynamoDB from Amazon, Voldemort (project-voldemort.com), Redis (redis.io), Riak (docs.riak.com).

Document-Oriented Storage (DOS) originates from the IBM's LotusNotes solutions [26]. They are based on document storages representing the structure of a tree. This scheme is focused on storing aggregated data. They are represented in the BSON (Binary JavaScript Object Notation) format, similar to JSON (JavaScript Object Notation). These systems are the most intuitive, with CouchDB (couchdb.apache.org) and MongoDB (mongodb.com) being the examples.

Column-Oriented Storage (COS) storage systems [27] are storage systems similar to column-oriented relational databases, but they have their peculiarities. Data Model – Row Key – Column Family – Column – Value. The examples include [13, 15, 16]: BigTable from Google, Hbase (hbase.apache.org), Cassandra (cassandra.apache.org), Hypertable (hypertable.org). HBase and Hypertable are a kind of “superstructure” over Hadoop. Hadoop, in turn, shares the logic of GFS – Google file system.

Graph systems (Graph Storage – GS) [28] model complex data fairly well and allow translating complex data into storage. The data model in this case is a collection of nodes, edges, and their attributes. The examples include: Neo4j (neo4j.com), AllegroGraph (allegrograph.com), GraphDB (graphdb.ontotext.com).

To ensure the integrity (consistency) of data, most classic database systems are transaction-based. The set of transactional parameters is called ACID (Atomicity-Consistency-Isolation-Reliability) [19-21]. However, meeting the ACID requirements presents scaling issues.

The high availability requirements of modern systems, known as the CAP-theorem (Consistency, Availably, Partition Tolerance), generate contradictions in distributed systems [22, 23, 29, 30]. The CAP theorem postulates that only two of three different aspects of horizontal scaling can be achieved completely simultaneously. The CAP theorem is a concept stating that for a distributed storage system it is impossible to achieve the properties of consistency, availability, and partition tolerance at the same time. Consistency means that a request for the same data in different nodes gives the same answer. Availability and partition tolerance means that any access to a system node will guarantee a response. Partition tolerance means that for any set of failures of nodes in the network, except for the entire network, a correct response to the request will be received. An accessibility violation is a situation when the node to which the request is being sent can wait indefinitely. A fragmentation violation means that a certain set of requests passing between sections of the network may not receive a response, but the nodes will be available. The CAP metrics are one of the basic metrics for evaluating the NoSQL solutions.

Many systems which support CA (Consistency – Availability) include relational DBMSs. The AP (Availably – Partition Tolerance) set includes key-value systems Dynamo (aws.amazon.com/ru/dynamodb), Voldemort, document-oriented systems CouchDB, Riak, and column-oriented Cassandra. The CP set (Consistency – Partition Tolerance) includes: key-value – BerkeleyDB, MemcacheDB (memcached.org), Redis; document-oriented MongoDB, Terrastore (dbdb.io/db/terrastore); column-oriented BigTable (cloud.google.com/bigtable), Hypertable, HBase.

Many NoSQL databases primarily reduce the consistency requirements to achieve better accessibility and separation, leaning towards the BASE model (base availability, flexible state, and final consistency) [23, 24].

3. Methods

Various approaches are used to analyze the efficiency of data processing solutions [25, 30]. The current work uses the method of comparative criteria analysis of expert data forming a space of properties, which is visualized using a visual spider web model [25]. The results obtained from the expert group using the criteria groups by means of ranking are tabulated and displayed on the spider-web visual model. The value of the complex coefficient is plotted on each branch of this model. The overall estimate is based on the effective inner area of the spider web polygon.

The first group of criteria includes *volumetric* requirements (in bytes). The second group includes criteria which determine the *complexity* of the data (the analysis uses the "simplicity" indicator to ensure the uniformity of the indicator impact on the overall efficiency). Data complexity is a concept in computer science and the theory of algorithms that denotes the function of the dependence of the amount of work performed by some algorithm on the size of the input data. Recently, the term "web 3.0" has been increasingly used, when a semantic link appears between the user-generated content, and the data stored in the form of semantic structures, for example, in the form of GiantGlobalGraph, a giant global graph actively used by social networks [12, 17, 18]. The third group includes the level of *clustering*. Data clustering is the automatic division of elements of a set into groups, depending on their semantic proximity. Recently, there has been a clear tendency to store data in different places, breaking it down logically or physically. Another group of criteria evaluates *encapsulation*. The object-oriented architecture of programs implies the presence of encapsulated objects, whose presentation is hidden. With this data implementation, some properties of objects should not be accessible outside the object. Object-relational mapping forces the entire content of the object to be exposed to the interaction with interfaces. Thus, the mapping breaks encapsulation. Still another group of criteria, according to the object-oriented paradigm, for providing and delimiting access to the interiors of the object, are *special interfaces*. The relational model also does not support inheritance and polymorphism, which makes it even more difficult to display objects. By evaluating a specific storage solution by the criteria groups (volume, complexity, clustering, encapsulation, special interfaces, and CAP level), an informed decision can be made about its effectiveness. All the main

criteria for evaluating the effectiveness can be summarized in eight groups (Table 1). The CA, CP, AP characteristics are determined by the CAP theorem and assigned by the respondents [29].

Table 1
Matrix of the criteria efficiency

Criteria	Name			
	KVS (Key-Value Storage)	DOS (Document-Oriented Storage)	COS (Column-Oriented Storage)	GS (Graph Storage)
Volume	6	9	8	7
Simplicity	8	6	4	5
Clustering	8	6	7	9
Encapsulation	6	7	8	8
Interfaces	7	5	4	6
CA	4	3	3	4
CP	5	8	6	9
AP	8	6	7	7

Each criterion is an average value set by a group of experts on a 10-point scale, which is plotted on the corresponding axis of the property space (represented as a visual spider web model) [25]. The best solution matches the solution with the maximum property space coverage area.

4. Results and discussion

The property space is constructed in the form of a visual cobweb model and used to evaluate the effectiveness of the main types of No-SQL solutions (in general) in comparison with the relational approach (Figure 1).

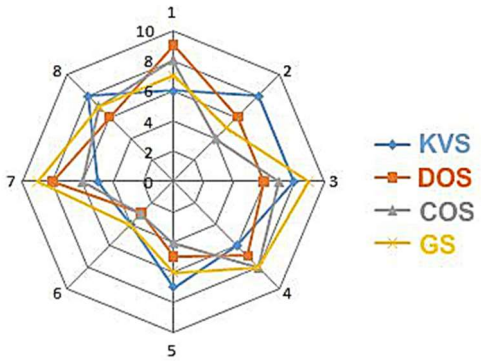

Figure 1: Property space of the No-SQL solutions is constructed in the form of a visual cobweb model (1 - Volume; 2 - Simplicity; 3 - Clustering; 4 - Encapsulation; 5 - Interfaces; 6 - CA; 7 - CP; 8 - AP).

To obtain expert assessments, the data obtained from the respondents as part of a free survey on profile sites and in profile groups of social networks in the Russian-speaking segment of the Internet were used. The experts are purposefully selected among highly qualified experienced specialists in the field of study, while the respondents are chosen randomly. When the experts are involved, the consistency of the experts is estimated according to a certain methodology and the results are ranked. The population shown in Table 1 is the average estimate of a random sample, which is a representative part of the general population of expert assessments from the respondents. The confidence probability ("accuracy") is 90%, and the confidence interval ("error") is 6%. The general population (total respondents) is 1150, and the required sample size is 128 respondents. The representativeness of the sample is ensured by the size and randomness of the selection of

respondents. The sample obtained by averaging the data from the sample of 128 respondents is representative of this study.

As one can see from Figure 1, the KVS (Key-Value Storage) model is fairly simple. Its performance is greatly increased due to caching mechanisms whichoperate based on mappings. According to the CAP theorem, this model shows good results in terms of AP, but loses in terms of the data consistency. The KVS model does not support the atomicity of transactions; the increase in the amount of the processed data necessitates maintaining the uniqueness of keys at the level of the applications themselves. The KVS model is preferable for storing images, creating specialized file systems, scalable Big Data systems, Internet of Things (IoT) systems, including industrial ones (Industrial IoT, IIoT).

The DOS (Document-Oriented Storage) model is semantically more complex, it includes metadata associated with the stored content, which allows one to make content-based queries. The data and relationships are not stored in tables but are a collection of independent documents. This model is well suited for various hierarchical structures, catalogs, CMS, etc.

The COS (Column-Oriented Storage) model assumes that data is stored in cells, grouped into columns rather than in rows. This is one of the most complex models in terms of its organization. But the use of the column storage enables fast search/access and data aggregation. This model provides the presence of timestamps, which allows it to be used for organizing counters, registering, and processing events related to time: analytics systems, IoT/IIoT applications, content management systems, etc.

The GS (Graph Storage) model uses a flexible graphical representation. This solution is focused on presenting a set of information with complex reciprocal links. Their area of application lies in communication-oriented tasks: social networks, navigation systems, various road maps, network topologies, etc.

As a result, the efficiency of the NoSQL solutions can be estimated at 65% for the KVS type, at 62% for the DOS type, at 58% for the COS type, at 68% for the GS type. Close efficiency values reflect the functional similarity of the ranking criteria for the types of the NoSQL solutions. When analyzing specific DBMSs, the difference in the indicators is more significant. In this paper, the application of the property space method is illustrated by an example of a generalized, complex assessment of the main types of NoSQL DBMS. This technique can be widely used in assessing the effectiveness of the specific No-SQL solutions.

5. Conclusion

It can be stated that, in general, NoSQL solutions provide relatively inexpensive, highly scalable storage for both large volumes and small data packages. They can be used for machine-to-machine communication (search and data exchange). A separate area of their application is analytics for semi-structured or hybrid data. Most NoSQL solutions are open source, which makes them preferable over the conventional commercial databases.

The analysis shows that it is impossible to achieve an effective solution for all the criteria at once. At the same time, losing in one thing can be compensated by other advantages. The trend in which it is difficult to single out a particular solution is called “The era of polyglot persistence”. It implies that different data stores must be used for different needs. Among the main advantages of non-relational DBMSs in comparison with the classical ones are linear scalability (an increase in the number of cluster nodes which increases the overall system performance), flexibility (full-text search can be implemented with partially structured data), convergence of information representations, high availability (replication, fault tolerance, dividing an array of information across different network nodes), productivity growth (due to the type of the solutions), functional completeness (built-in data manipulation languages (DML), API, interfaces, processing of complex, multivalued data types).

Any new solution has well-publicized advantages and unknown disadvantages, and any classic solution has forgotten advantages and many disadvantages identified as a result of exploitation. The NoSQL solutions are at the beginning of their development, but the limited capacity of the built-in DML, complexity in the implementation of full ACID requirements for transactions, inconsistency of the requirements of the CAP model (consistency, availability, resistance to separation), and BASE

model (basic availability, flexible state and final consistency), platform dependence of the application to a specific DBMS due to the specific of DML and the applied data model should already be noted.

The following postulate remains valid: a specific task requires a specific solution. The classic SQL solutions are focused on processing strongly typed information of a relatively small volume. When processing a large amount of semi-structured and unstructured data (Big Data) in a distributed system, it is advisable to use NoSQL solutions. Further work will focus on the analysis of the specific NoSQL solutions for the implementation of specific applications. The proposed method for assessing the property space will make this choice more valid and reasoned.

6. Acknowledgements

Some project results were obtained with the Ministry of Science and Higher Education's financial support for project No. 0705-2020-0041, “Fundamental research of methods of the digital transformation of the component base of micro-and nanosystems”.

7. References

[1] V. Akberdina, A. Kalinina, A.Vlasov, Transformation stages of the Russian industrial complex in the context of economy digitization, Problems and Perspectives in Management 16(4) (2018) 201–211.

[2] S. Wang, J. Wan, D. Li, C. Zhang, Implementing smart factory of Industry 4.0: Outlook, International Journal of Distributed Sensor Networks (2016) 3159805.

[3] V. A. Shakhnov, A. E. Kurnosenko, A. A. Demin, A. I. Vlasov, Industry 4.0 visual tools for digital twin system design, Advances in Intelligent Systems and Computing 1295, (2020) 864–875.

[4] V. A. Shakhnov, A. E. Kurnosenko, Modelling of the digital electronics manufacturing in the context of Industry 4.0 concept, in Proceedings of the 1st International Scientific and Practical Conference Digital Transformation of the Industry: Trends, Management, Strategies, 2019, pp. 585–594.

[5] H. P. Breivold, K. Sandström, Internet of Things for Industrial Automation – Challenges and Technical Solutions, in: IEEE International Conference on Data Science and Data Intensive Systems, 2015, pp. 532–539.

[6] Q. Zhao, Presents the Technology, Protocols, and New Innovations in Industrial Internet of Things (IIoT), Internet of Things for Industry 4.0. EAI, Springer Innovations in Communication and Computing, Springer, Cham, 2020.

[7] R. K. Bathla, Shallu, G. Suseendran, Research analysis of BIG DATA and cloud computing with emerging impact of testing, International Journal of Engineering and Technology (UAE) 7(3), 27 (2018) 239–243.

[8] Z. Zhou, L. Zhao, Cloud computing model for BIG DATA processing and performance optimization of multimedia communication, Computer Communications 160 (2020) 326-332.

[9] C. Yang, K. Liu, F. Hu, Q. Huang, Z. Li, BIG DATA and CLOUD COMPUTING: innovation opportunities and challenges, International Journal of Digital Earth 10(1) (2017) 13–53.

[10] A. A. Prudius, A. A. Karpunin, A. I. V lasov, Analysis of machine learning methods to improve efficiency of BIG DATA processing in Industry 4.0, in: Journal of Physics: Conference Series, volume 1333(3), 2019 p.032065.

[11] A. I. Vlasov, K. A. Muraviev, A. A. Prudius, D. A. Uzenkov, Load balancing in BIG DATA processing systems, International Review of Automatic Control 12(1) (2019) 42–47.

[12] B. A. Novikov, M. Y. Levin, Comparative analysis of SQL and NOSQL DBMS performance, Computer Tools in Education 4 (2017) 48–63.

[13] D. Barberis, Modern SQL and NOSQL database technologies for the atlas experiment. Compilation, in: CEUR Workshop Proceedings. 26. Ser. “Selected Papers of the 26th International Symposium on Nuclear Electronics and Computing (NEC 2017), 2017, pp. 15–22.

[14] G. Dodonov, B. B. Chumak, Dynamics of the transition from a relational model to NOSQL solutions,. Scientific Almanac 3-2(41) (2018) 24–29.

[15] L. Y. Krstic, M. S. Krstic, Testing the capabilities of the NOSQL database using the database benchmark tool, Military Technical Bulletin 66(3) (2018) 614–639.
[16] R. O. Kuznetsov, Application of NOSQL Databases to Implement Web Programming Capabilities, Economy and Society 5-2(36) (2017) 552–557.
[17] S. V. Chubeiko, V. S. Palaguta, Features of NOSQL and their comparison with relational databases, in: Proceedings of the Rostov State Transport University4, 2017, pp.98–101.
[18] A. O. Deniskova, NOSQL databases and their types, Theory and Practice of Modern Science 4(46) (2019) 49–52.
[19] A. O. Deniskova, Basic concepts of NOSQL, in: Materials of the International Scientific and Practical Conference Modern systems of scientific knowledge, 2019, pp.6–8.
[20] I. I. Kartavets, Prospects for the development of NOSQL. Colloquium-Journal 11-1(35) (2019) 84–87.
[21] I. I. K artavets, Relevance of the use of NOSQL databases, Colloquium-Journal 13-2(37) (2019) 62–64.
[22] A. M. Tammemägi, Research of KEY-VALUE NOSQL database management systems. Distance and Virtual Training 6(72) (2013) 60–67.
[23] Y. A. Grigoriev, Analysis of NOSQL Database Properties, Computer Science and Control Systems 2(36), (2013) 003–013.
[24] D. A. Davydov, P. S. Manylov, Trends in the development of NOSQL-DBMS, Scientific and Technical Bulletin of the Volga Region 3 (2013) 131–135.
[25] A. I. Vlasov, V. A. Shakhnov, Visual methodology for the multi-factor assessment of industrial digital transformation components, in Proceedings of the International Scientific Conference “Digital transformation in industry: trends, management, strategies”. Lecture Notes in Information Systems and Organisation, 2020.
[26] D. McCreary, A. Kelly Making Sense of NoSQL: A guide for managers and the rest of us. Manning Publications, 2013.
[27] S. Harizopoulos, D. Abadi, P. Boncz, Column-Oriented Database Systems. VLDB. Tutorial, 2009.
[28] I. Robinson, J. Webber, E. Eifrem, Graph Databases. O’Reilly Media, 2013.
[29] Everything you didn't know about the CAP theorem, URL: https://habr.com/ru/post/328792/.
[30] I. A. Rygovskiy, Analysis of the effectiveness of methods for processing large data arrays using computing systems, Informatics Problems 2 (2014) 54–58.