



Jarosław Kurpanik

Uniwersytet Ekonomiczny w Katowicach
Wydział Informatyki i Komunikacji
Katedra Informatyki
jaroslaw.kurpanik@ue.katowice.pl

Małgorzata Pańkowska

Uniwersytet Ekonomiczny w Katowicach
Wydział Informatyki i Komunikacji
Katedra Informatyki
pank@ue.katowice.pl

NOSQL PROBLEM LITERATURE REVIEW

Summary: In this work authors propose to perform an extensive literature review (LR) to elicit all the relevant NoSQL mechanisms, describing them and pointing out the main references. In the first section, they introduce the research methodology where they elicit the main steps to perform a LR. Afterwards, they describe the problem this research intends to help solve. They follow with a LR regarding NoSQL mechanisms and identify and describe the main NoSQL problems. The main contribution concerns that literature searching results are presented in the mind map. Generally, mind mapping is a technique for the concept relations' visualisation. At the beginning the map is created around a single concept, e.g. NoSQL, which is connected directly to another concept and the other ideas branch out from that. For mind mapping, authors use an open source tool, i.e., Mind Map Maker. Then, they summarize how they evaluate their work. Finally, they conclude about the research and further works.

Keywords: NoSQL, mind mapping, literature review, Mind Map Maker.

Introduction

In general, relational database management systems (e.g., MySQL, PostgreSQL, SQL Server, Oracle) have been considered as the best solution for data persistence and retrieval for decades. However, the increasing need for scalability and new application requirements have created new challenges for data management. Nowadays developed NoSQL systems offer a number of advantages:

- the ability to horizontally scale out throughput over many servers,
- a simple call level interface or protocol,
- supporting weaker consistency models in contrast to ACID (Atomicity, Consistency, Isolation, Durability) guaranteed properties for transactions in most traditional RDBMS (Relational Database Management System),

- efficient use of distributed indexes and RAM for data storage,
- the ability to dynamically define new attributes or data schema [Sadalage, Fowler, 2013].

The paper aims to present the state-of-the-art knowledge on NoSQL databases to reveal the knowledge gap, which would be interesting for the further research. In order to establish the conceptual and theory development it is necessary to determine the current state of knowledge within the subject area. The literature review allows to discover what is already known about the particular topic, sometimes from different disciplines and bring it together [Pickard, 2007]. The paper consists of two parts. At first, the literature review approach is explained. Next, the basic and new NoSQL models are presented, NoSQL framework is analyzed, and performance and security issues are discussed.

Literature review research methodology

The literature review is defined as the use of ideas to justify the particular approach to a topic, the selection of methods, and demonstration that particular research could contribute something new. In this paper, the thesis is formulated that literature review represents the foundation for research in NoSQL. Generally, the literature review process involves some distinct stages:

- 1) defining research goals,
- 2) specification of key words, concepts, methods, algorithms, authors, sources, which seem to be the most representative for the research goals,
- 3) identification of the relevant literature focused on the concepts,
- 4) validation of quality of collected literature,
- 5) testing the gathered literature for applicability in the study,
- 6) structuring the review of accessible literature,
- 7) describing the research problem and supporting it with good references,
- 8) specification of knowledge and advice for possible future work,
- 9) demonstration of the understanding of the work and providing some examples of the specified concepts,
- 10) evaluation of the literature review [Falcao e Cunha, Snene, Novoa, 2013; Pickard, 2007].

NoSQL database approach justification

NoSQL systems are viable solutions for applications that require scalable data repositories, which can easily scale out over multiple servers and support flexible data model and storage schema. In general, NoSQL systems are recommended for newly developed applications, but not for migrations of existing ap-

lications which are written on top of traditional relational database systems. NoSQL databases are generally designed from the ground up to require less management. For example, automatic repair and the simpler data model features should lead to lower administration and tuning requirements. NoSQL databases typically use clusters of cheap commodity servers to manage the data and transaction volumes. The cost per gigabyte or transactions per second for NoSQL can be many times less than the cost for RDBMS, which allows a NoSQL system to store and process more data at a much lower price. NoSQL application uses data that is distributed across hundreds of servers, therefore simple economics points to the benefit of using no-cost server software as opposed to that of paying per-processor license fees [Lake, Crowther, 2013].

NoSQL story and characteristics

NoSQL term for the first time was used in 1998 by Carlo Strozzi, who designed the relational database system without SQL implementation and named it Strozzi NoSQL (Fig. 1) [Sadalage, Fowler, 2013; Bajpeyee, Sinha, Kumar, 2015; He, 2015].

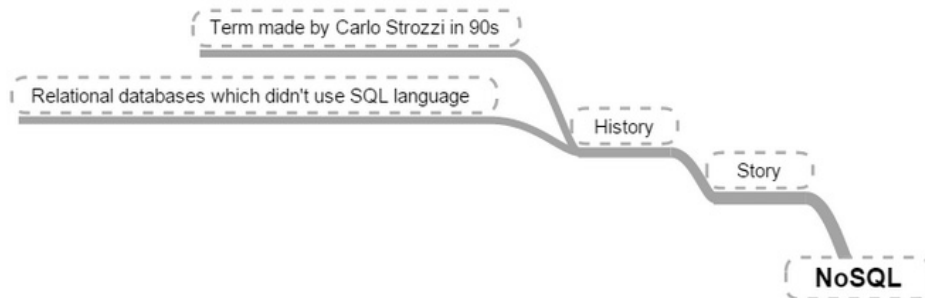


Fig. 1. MindMap – past part of story

Although the solution proposed by Strozzi was not popular, the NoSQL term is still applied. Today NoSQL means Not Only SQL, according to Eric Evans, who considered nonrelational database management systems, applied for analysis and searching big, not necessarily well-ordered data sets. Other terms used for these database systems are as follows: NewSQL, NonSQL, nonRDBMS [Ntarmos, Patlakas, Triantafillou, 2014; Gudivada, Rao, Raghavan, 2015; Jomeiri, Shamsi, Kazemi, 2015; Wu C.M., Huang, Lee, 2015; Pothuganti, 2015].

These systems are characterized by a light interface, automatic data replication realized synchronously or asynchronously, avoiding tables and schemas, working in clusters, where data divided by the cluster nodes is not tied together. Beyond that, ACID transactions are not possible in such NoSQL approach. Nonrelational databases are highly scalable and collect data from Internet portals, system logs, and

other sources in real time (Fig. 3) [Gomez et al., 2014; Ntarmos, Patlakas, Triantafyllou, 2014; Yan et al., 2014; Bajpeyee, Sinha, Kumar, 2015; Bao et al., 2015; Chevalier et al., 2015; Gudivada, Rao, Raghavan, 2015; Hwang et al., 2015; Jain, Rajput, Sayankar, 2015; Jomeiri, Shamsi, Kazemi, 2015; Naresh, Vani, 2015; Nataraja Sekhar, Saritha, Penchalaiah, 2015; Pothuganti, 2015; Kumar Shukla, Pandey, Kumar, 2015; Saur, Dumitras, Hicks, 2015; Tinkhede, Deshpande, 2015].

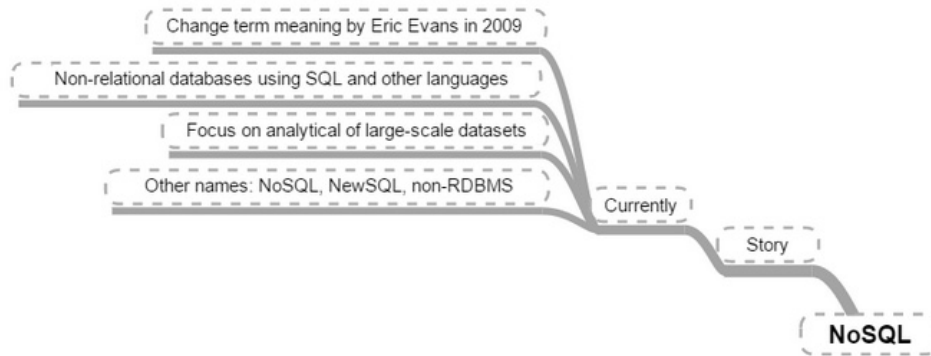


Fig. 2. MindMap – current part of story

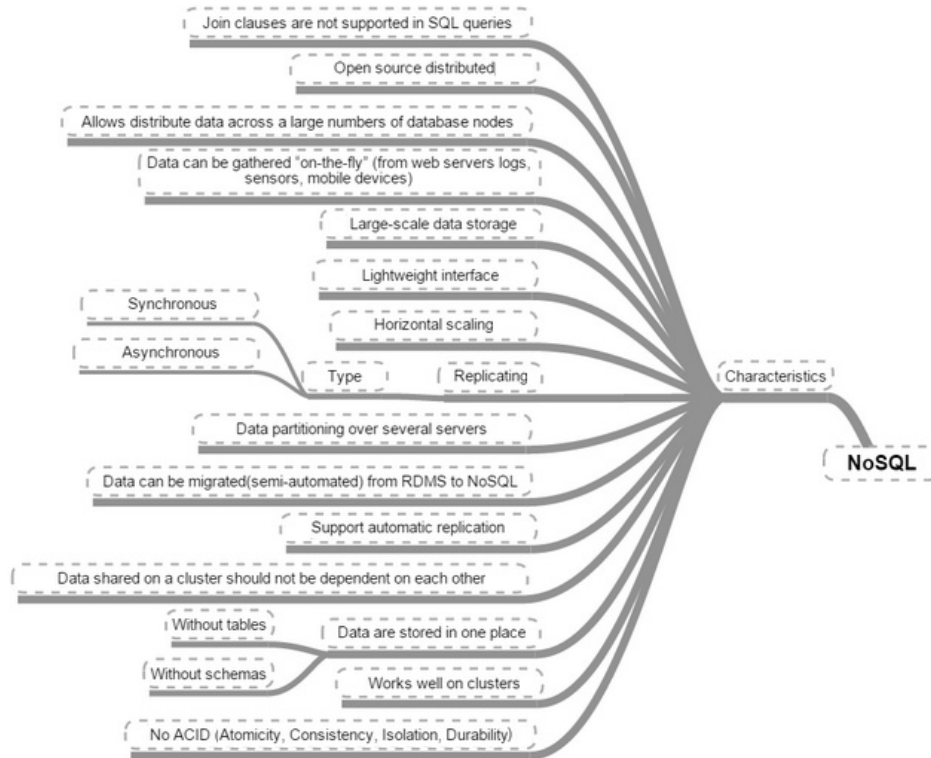


Fig. 3. MindMap – NoSQL characteristics

Theoretical principles

NoSQL solutions are based on three fundamental principles: CAP, BASE and eventual consistency. The CAP (Consistency, Availability, Partition tolerance) theory was proposed in 2000 by Eric Brewer. In 2002, Seth Gilbert and Nancy Lynch changed that theory and proved that one system at a time can fulfill two and only two of the assumptions, i.e., consistency, availability and partition tolerance. For consistency requirement compliance, all the cluster nodes should have access to the same data at the same time. Availability requirement guarantees that for each query the user will get an answer if the query succeeded or failed. A reply failure does not prevent the system from continuing to operate. The third requirement on tolerance to network partitions guarantees that the system will act correctly despite partial network damage. The system still functions when distributed replicas cannot communicate to each other. Taking into account corrected and approved CAP theory, nonrelational database management systems can work in three combinations: CA (consistency with availability), AP (availability with partition tolerance) and CP (consistency with partition tolerance) (Fig. 4).

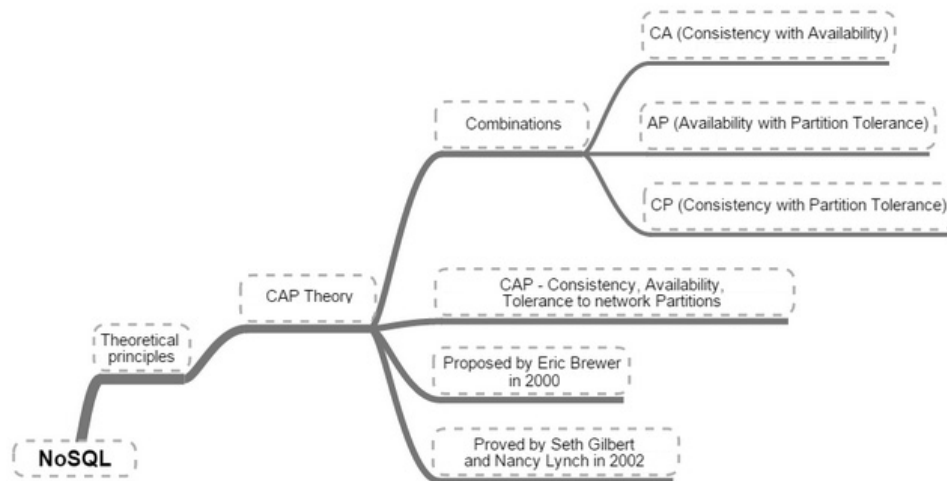


Fig. 4. MindMap – CAP theory principles

The BASE (Basically, Availability, Soft-State, Eventual Consistency) model is also fundamental to the NoSQL theory as an alternative to the ACID relational model. The BASE model is to ensure:

- basic availability i.e., data accessibility even in a damage situation, what is possible because of the high distribution of data, which is replicable among cluster nodes;
- soft state i.e., cancelling the data cohesion requirement and assuming that it is the developer's problem and that eventually even incoherent data will aim to the cohesion, but it is impossible to explicitly define the time needed for that (Fig. 5) [Ntarmos, Patlakas, Triantafillou, 2014; Bajpeyee, Sinha, Kumar, 2015; Chevalier et al., 2015; Fiannaca, Huang, 2015; Ghosh et al., 2015; Gudivada, Rao, Raghavan, 2015; He, 2015; Jomeiri, Shamsi, Kazemi, 2015; Kirti, Maan, 2015; Wu C.M., Huang, Lee, 2015; Naresh, Vani, 2015; Niewiadomski, Strobin, 2015; Kumar Shukla, Pandey, Kumar, 2015; Saur, Dumitras, Hicks, 2015; Shao, Conrad, 2015; Shibata, Kurachi, 2015; Tinkhede, Deshpande, 2015].

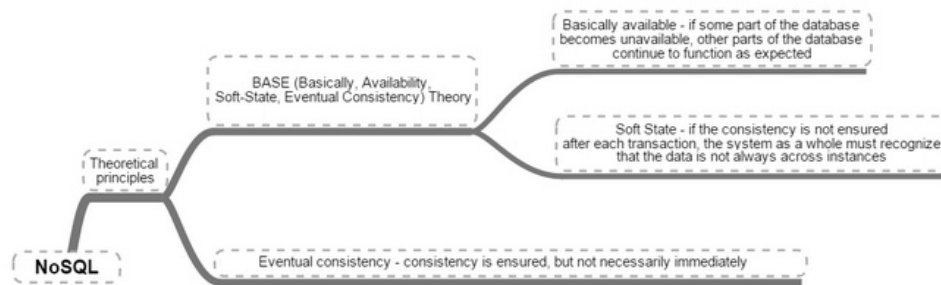


Fig. 5. MindMap – BASE theoretical principles

NoSQL classification

Nonrelational database management systems are classified according to the data model defining the logical organization of data. Nowadays, there are two data model groups: basic and new models (Fig. 6).

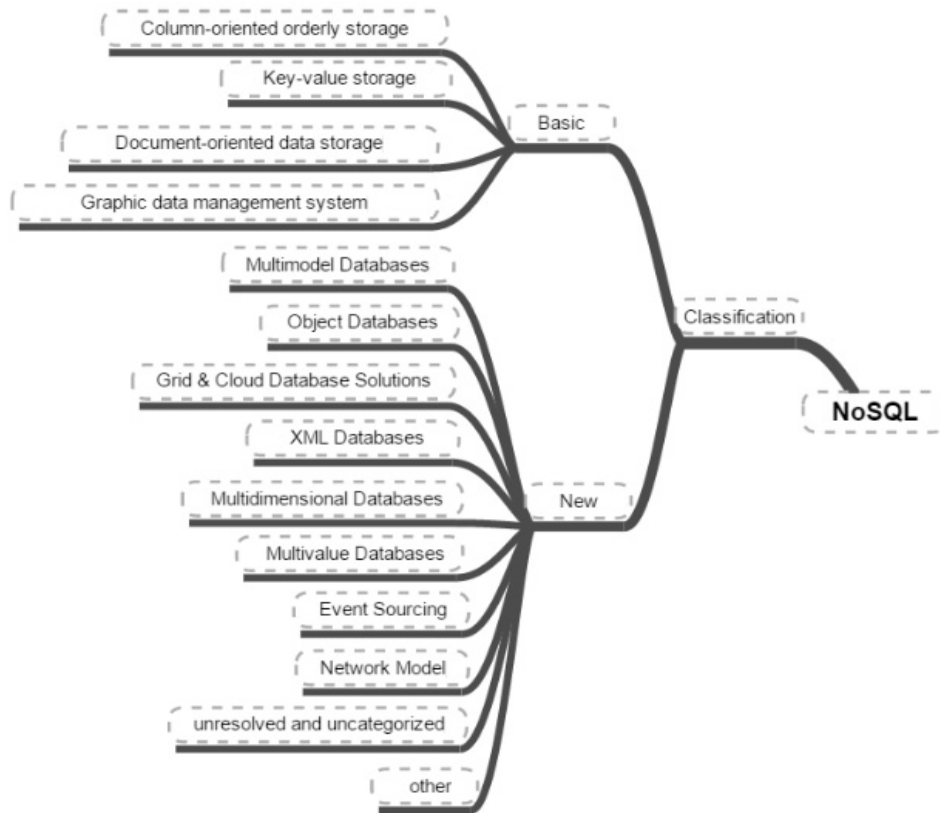


Fig. 6. MindMap – No SQL models classification

Columnar database, document database, key-value pair database and graph database models constitute the basic models group. The new models group covers multimodel, grid model, multidimensional model, multivalued model, network model and others, which cannot be explicitly classified [Sadalage, Fowler, 2013; He, 2015; Jain, Rajput, Sayankar, 2015; Kirti, Maan, 2015; NoSQL].

Basic models

Column-oriented database model

Column-oriented databases have a huge number of columns for each tuple. Each column also has a column key. Related columns have a column-family qualifier so that they can be retrieved together during a search. Because each column also has a column key, these databases are suitable for fast writes. Typical examples of the columnar database systems cover Cassandra, HBase, Hypertable (Fig. 7) [NoSQL; Sadalage, Fowler, 2013].

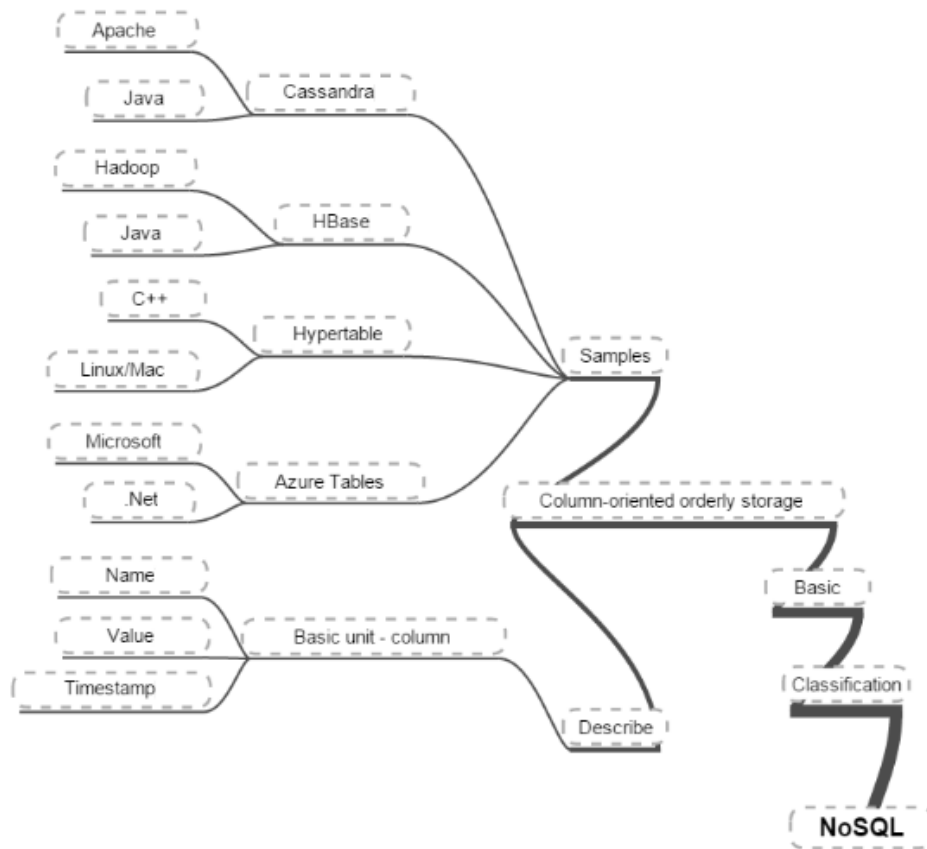


Fig. 7. MindMap – Columnar Database Models

Key-value pair database model

Key-value pair databases store data as simple key-value pairs. The keys are unique and do not have any foreign keys or constraints. They are suitable for parallel lookups because the data sources have no relationships among each other. As you can imagine, such a structure is good for high read access. Due to a lack of referential integrity, the data integrity has to be managed by the front-end applications. The well known examples of that database model are as follows: Redis, BerkleyDB, LevelDB (Fig. 8) [NoSQL; Sadalage, Fowler, 2013].

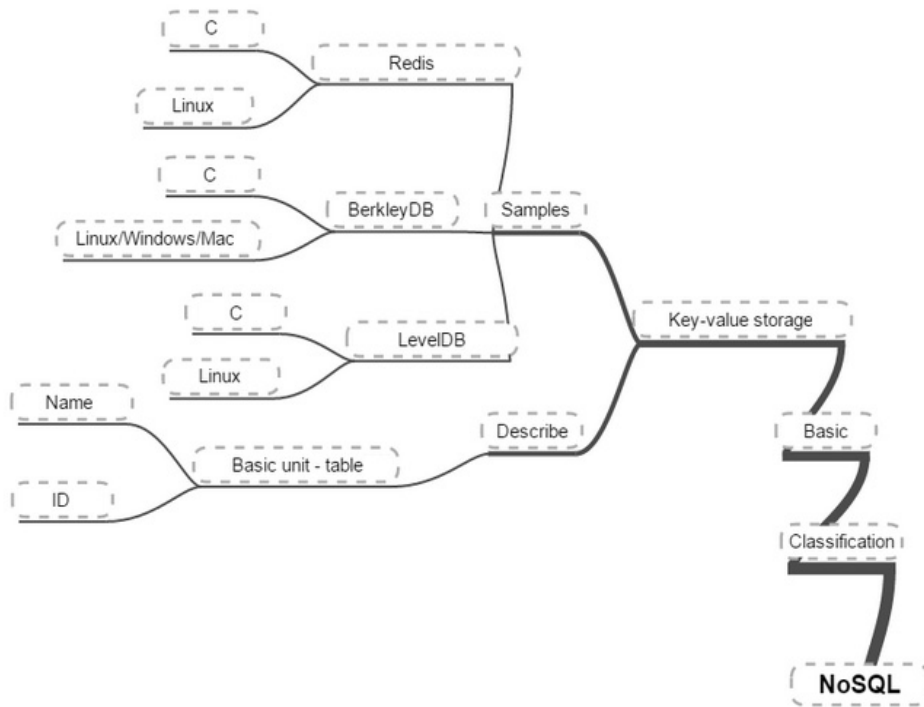


Fig. 8. MindMap – Key-Value Pair Database Models

Document database model

Document databases store text, media, and JSON (Java Script Object Notation) or XML data.

The value in a row is a blob of the aforementioned data and can be retrieved using a key. If search is needed through multiple documents for a specific string, a document database should be used. CouchDB and MongoDB are the most well-known representatives of the document database model (Fig. 9) [NoSQL; Sadalage, Fowler, 2013].

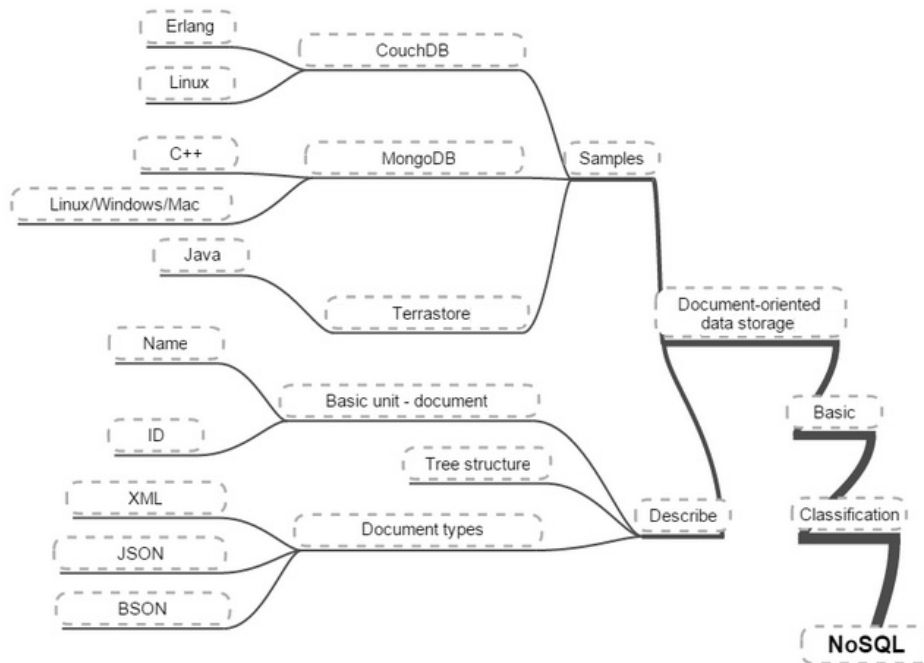


Fig. 9. MindMap – Document-Oriented Database Models

Graph database model

Graph databases store data entities and connections between them as nodes and edges. They are similar to a network database and can be used to calculate shortest paths, social network analysis, and other parameters. HyperGraph, InfoGrid, Neo4 are typical examples of such database models (Fig. 10) [NoSQL; Sadalage, Fowler, 2013].

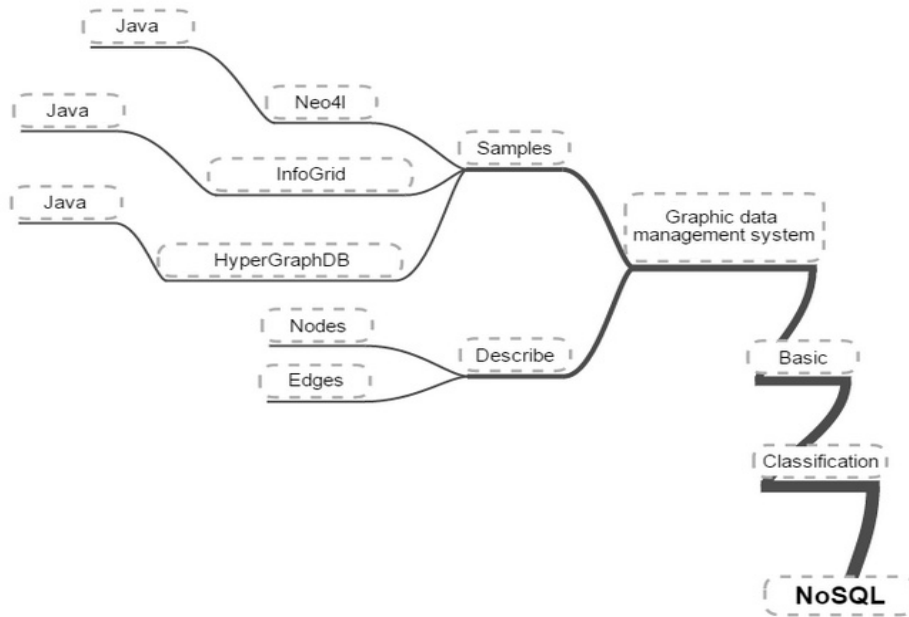


Fig. 10. MindMap – Graph Database Models

New models

Database storing data differently from the mentioned above are presented in Fig. 11.

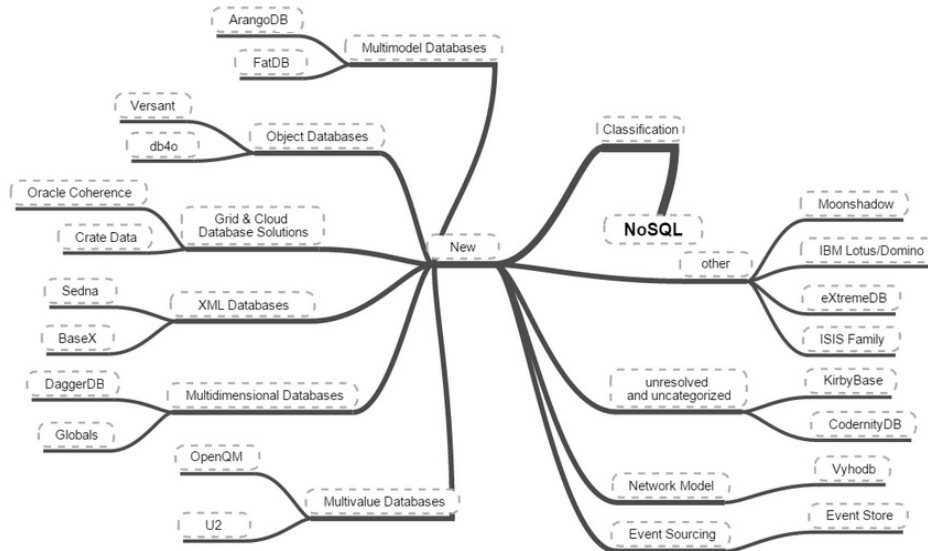


Fig. 11. MindMap – New NoSQL database models classification

The new models cover multimodel, multidimensional and multivalued databases, event sourcing, grid and cloud databases and many uncategorized solutions that are an inspiration for the further study and research.

NoSQL framework

NoSQL database framework consists of four layers, i.e., interface, data model, data distribution and data persistence. Each of the layer additionally consists of the functional modules (Fig. 12) [He, 2015; Kumar Shukla, Pandey, Kumar, 2015].

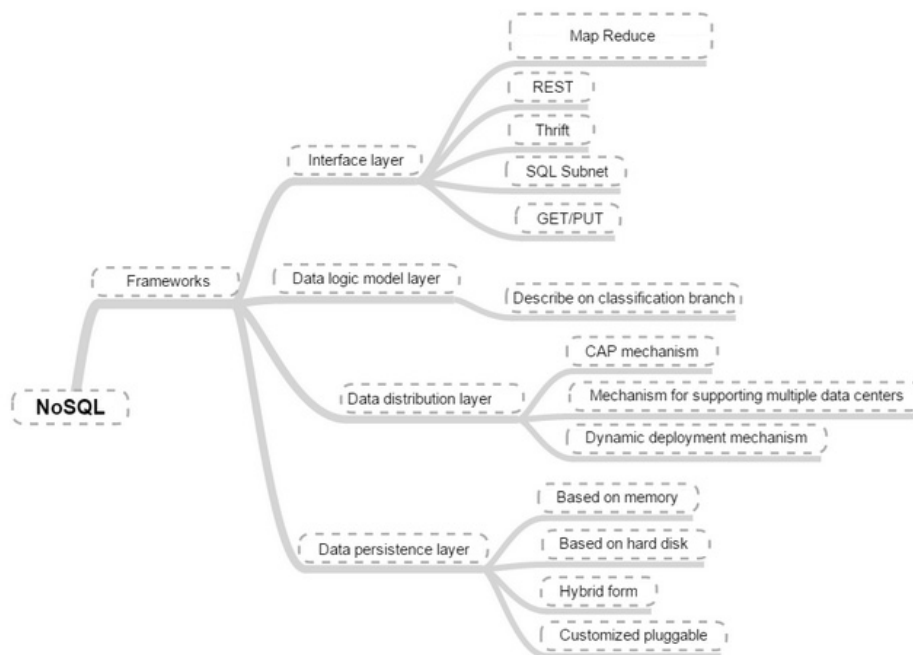


Fig. 12. MindMap – NoSQL framework layers

Interface layer

Interface layer is responsible for ensuring comfortable access to data. Modules and platforms for parallel processing of the big data in clusters provided by that layer are as follows: REST (Representational State Transfer), RPC protocol Thrift, Memcached-like Get/Put method, SQL subset and Google provided

MapReduce (Fig. 13). Map (master node) divides data into smaller, sub-ordinate nodes, which are sent to Reducer that provides further division of data or processes received portions of data. Received result is resent to the main node in form of an “answer to the query”. The main advantage of that solution is the operational dispersion, where each Map function can be realized separately on detached server [Bajpeyee, Sinha, Kumar, 2015; He, 2015; Nataraja Sekhar, Saritha, Penchalaiah, 2015; Naresh, Vani, 2015; Pothuganti, 2015; Kumar Shukla, Pandey, Kumar, 2015; Rajput et al., 2015; Tathe et al., 2015; Wassan, 2014].

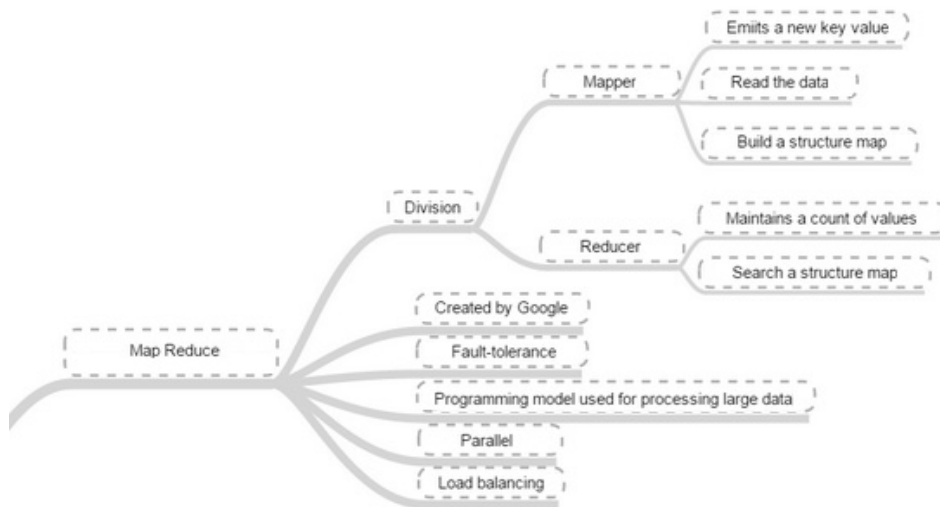


Fig. 13. MindMap – MapReduce layers framework

Data logic model layer

Data logic model layer is responsible for the description of the data logic model (Fig. 7) [He, 2015; Kumar Shukla, Pandey, Kumar, 2015]. Beyond that data can be stored by NoSQL databases in three forms: structured, unstructured and semi-structured (Fig. 14) [Hwang et al., 2015; Kirti, Maan, 2015; Shibata, Kurachi, 2015].



Fig. 14. MindMap – data structure

Data distribution layer

Data distribution layer is responsible for the distribution of data among the cluster nodes. In comparison with relational database systems NoSQL databases have many mechanisms, e.g., CAP mechanism, to support the horizontal extension, mechanism supporting the multiple data centers, and mechanisms for dynamic deployment (Fig. 12) [He, 2015; Kumar Shukla, Pandey, Kumar, 2015].

Data persistence layer

Data persistence layer is responsible for the data storing forms. There are some forms, e.g., internal storage form, which are very fast, but there is a risk of data loss, hard disk form, which ensures high persistence of data, but low speed of reads, hybrid form, which combines the advantages of the two aforementioned forms, and the last – customized pluggable form known as flexible (Fig. 12) [He, 2015; Kumar Shukla, Pandey, Kumar, 2015].

NoSQL performance

NoSQL databases are characterized by high efficiency, which is achieved by work in cluster environments. High efficiency is combined with configuring parameters, which determine load distribution, efficient concurrent processing within all cluster nodes and high usage of accessible resources. Parameters suit-

able for a load need not to be appropriate for another load, therefore the parameter selection process is continuous. Loads typical for NoSQL databases are classified as follows: write-intensive, read-intensive, and read/write-mix. Further, write-intensive workloads are divided into bulk loading, write-only and write-mostly. Read-intensive workloads are divided into read-only and read-mostly. In the last read/write mix group there are read/write mix with similar read/write proportion, read/write-mix with more read proportion and, read/write mix with more write proportion workloads (Fig. 15) [Bao et al., 2015; Ghosh et al., 2015; Hanlon et al., 2015; Katkar, 2015; Tathe et al., 2015; Tinkhede, Deshpande, 2015].

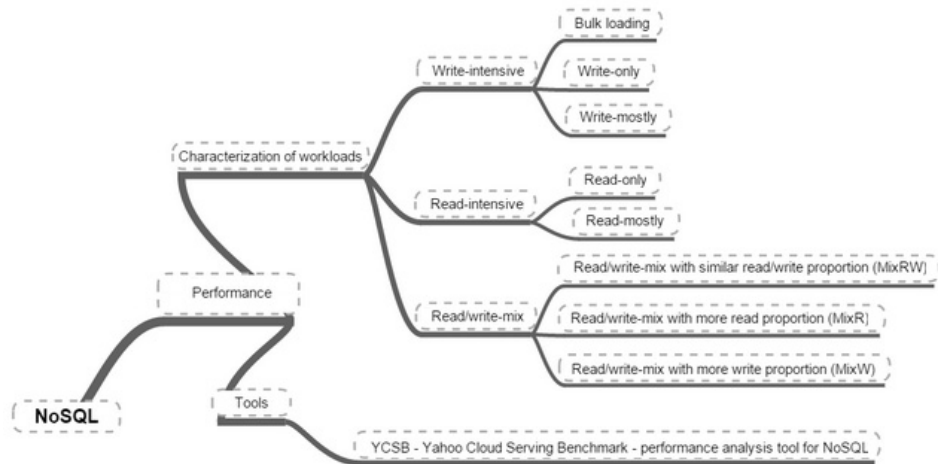


Fig. 15. MindMap – NoSQL performance

NoSQL security

NoSQL databases are still in progress, therefore the security issues, i.e., simple authentication, data cryptography, role based data access management are also under initiation and development. The security is provided on middleware level and on the operating system level. There are still lacks of good protection in face of threats such as array injection, denial of service or SQL injection (Fig. 16) [Kaushal, Khan, Kumar, 2015; Ron, Shulman-Peleg, Bronshtein, 2015].

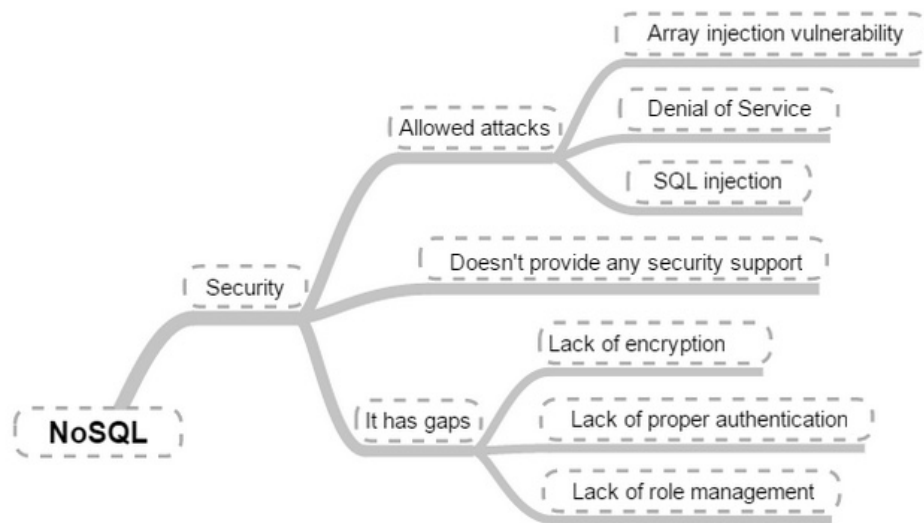


Fig. 16. MindMap – NoSQL security issues

NoSQL application

NoSQL databases are mostly used as big data storages, because they ensure high agility, relatively small delays during data reads, and high efficiency of data recording. These advantages have been noticed by huge IT firms, i.e., Facebook, Amazon, Twitter, and Google and they strongly support the NoSQL database technology development. Beyond that, NoSQL databases are applied in biological and medical research (Fig. 17) [Kumar, Babu, 2015; Kumar Shukla, Pandey, Kumar, 2015; Shang, Shi, Shang, 2015; Shibata, Kurachi, 2015; Tinkhede, Deshpande, 2015].

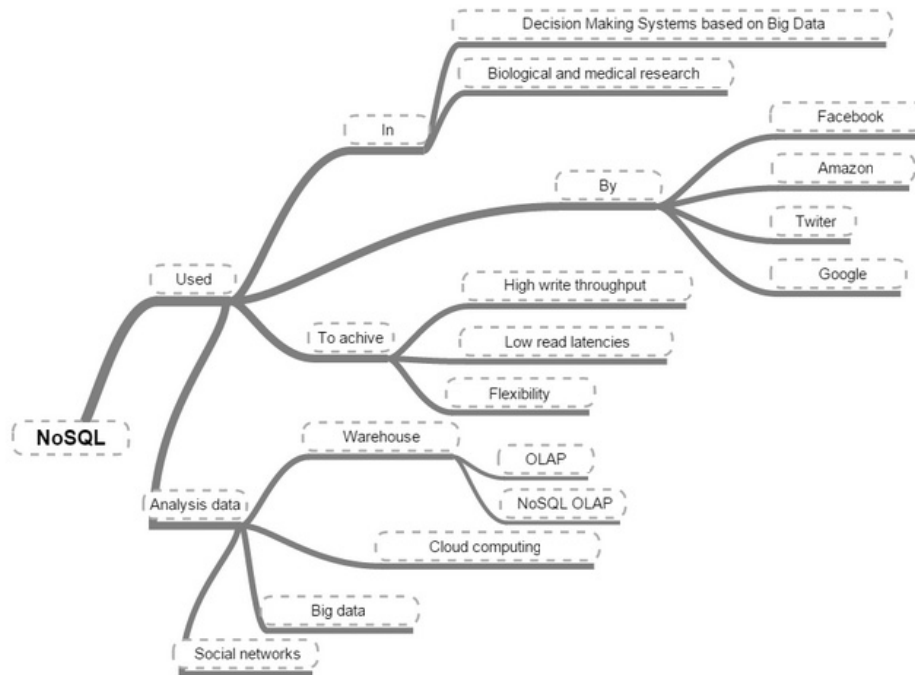


Fig. 17. MindMap – NoSQL application domains

Conclusions and future work

The paper presents the practical usage of mind mapping to support the NoSQL literature review. The applied approach permitted to structure the NoSQL concepts, reveal the knowledge and research gaps, to discuss which categories are important, and to create a certain hierarchy of categories.

NoSQL database systems are still developing, although a lot has been done. Taking into account the literature review, it should be emphasized that NoSQL setups provide an opportunity to process data in parallel within a system and across multiple systems at the same time. New architecture is still developed to ensure that RDBMS, DBMS, and NoSQL technologies have a role and can be deployed to solve the requirement of minimal and economic database usage [Krishnan, 2013]. There are still many challenges, particularly concerning data query mechanisms, data searching, security, and data migration among relational database systems and NoSQL systems (Fig. 18) [Gudivada, Rao, Raghavan, 2015; Kirti, Maan, 2015; Rajput et al., 2015].

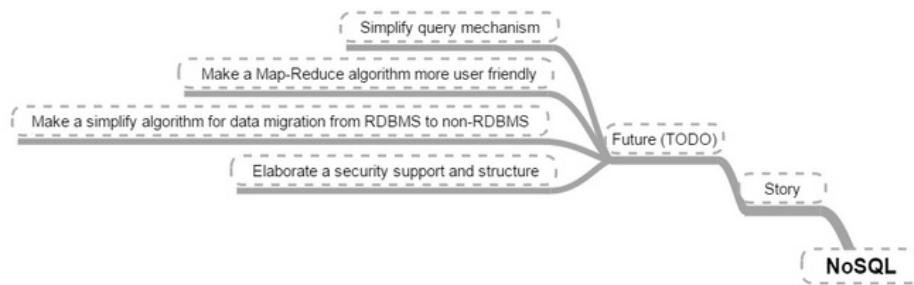


Fig. 18. MindMap – NoSQL future research directions

References

- Abd El-razek S. M., El-Mashad Y., Mastorakis N. (2015), *Enhancing Big Data Processing in Educational Systems* [in:] *Advances in Computers and Technology for Education*, Proceedings of the 11th International Conference on Educational Technologies (EDUTE '15), Dubai, United Arab Emirates, pp. 176-181.
- Ashraf A., El-Bakry H.M. (2015), *Handling Big Data in e-Learning*, “International Journal of Advanced Research in Computer Science and Technology”, Vol. 3, Issue 1, pp. 47-51.
- Bajpeyee R., Sinha S.P., Kumar V. (2015), *Big Data: A Brief Investigation on NoSQL Databases*, “International Journal of Innovations & Advancement in Computer Science”, Vol. 4, Issue 1, January, pp. 28-35.
- Bao X., Liu L., Xiao N., Zhou Y., Zhang Q. (2015), *Policy-driven Autonomic Configuration Management for NoSQL*, National University of Defense Technology, Hunan, China, Georgia Institute of Technology, Atlanta, Georgia, USA.
- Chevalier M., El Malki M., Kopluku A., Teste O., Tournier R. (2015), *Implementing Multidimensional Data Warehouses into NoSQL*, University of Toulouse, IRIT, Toulouse, France.
- Chonkar S., Suman S., Sawant S., Moondra S. (2015), *Supercharging web applications with Dart and MongoDB*, “International Journal of Current Engineering and Technology”, Vol. 5, No. 1, Valley Village, USA, pp. 409-412.
- Falcao e Cunha J., Snene M., Novoa H. (2013), *Exploring Services Science*, Springer, Berlin.
- Fiannaca A.J., Huang J. (2015), *Benchmarking of Relational and NoSQL Databases to Determine Constraints for Querying Robot Execution Logs*, Computer Science & Engineering, University of Washington, USA, 1-8.
- Gadekar M.D., Jadhav B.M., Shaikh A.S., Kokare R.B. (2015), *Natural Language (English) To MongoDB Interface*, “International Journal of Advanced Research in Computer Engineering & Technology”, Vol. 4, Issue 3, pp. 1081-1083.
- Ghosh M., Gupta I., Gupta S., Kumar N. (2015), *Fast Compaction Algorithms for NoSQL Databases*, University of Illinois, University of California, USA.

- Gomez A., Ouanouki R., Ravello A., April A., Abran A. (2014), *Experimental Validation as Support in the Migration from SQL Databases to NoSQL Databases*, Department de genie logiciel et des TI Ecole de Technologie Superieure, Montreal, Canada.
- Gudivada V.N., Rao D., Raghavan V. (2015), *Renaissance in Data Management Systems: SQL, NoSQL, and NewSQL**, Marshall University, Huntington, University of Louisiana at Lafayette, Lafayette.
- Han M. (2015), *The Application of NoSQL Database in Air Quality Monitoring*, International Conference on Intelligent Systems Research and Mechatronics Engineering, Zhengzhou, China, pp. 101-104.
- Hanlon M.R., Dooley R., Mock S., Dahan M., Nuthulapati P., Hurley P. (2015), *A Case Study for NoSQL Applications and Performance Benefits: CouchDB vs. Postgres*, Texas Advanced Computing Center, Austin, USA.
- He Ch. (2015), *Survey on NoSQL Database Technology*, "Journal of Applied Science and Engineering Innovation", Vol. 2, No. 2, Information Technology Service Center of Hexi University, Gansu Zhangye, China, pp. 50-54.
- Hwang J.S., Lee S., Lee Y., Park S. (2015), *A Selection Method of Database System in Bigdata Environment: A Case Study From Smart Education Service in Korea*, "International Journal Advance Soft Computing Application", Vol. 7, No. 1, March, Johor, Malaysia, pp. 9-21.
- Jain P.J., Rajput B., Sayankar A. (2015), *Implications for Data Analyses and Higher Education with SQL and Data Analysis*, "International Journal for Engineering Applications and Technology", April, Nagpur, India, pp. 80-86.
- Jomeiri A., Shamsi M., Kazemi E. (2015), *Comparative Study of Column Oriented NoSQL Databases on Characteristics*, "International Journal of Enhanced Research in Science Technology & Engineering", Vol. 4, Issue 4, Haryana, India, pp.118-124.
- Katkar M. (2015), *Performance Analysis for NoSQL and SQL*, "International Journal of Innovative and Emerging Research in Engineering", Vol. 2, No. 3, Mumbai, India, pp. 12-17.
- Kaushal A., Khan A., Kumar V. (2015), *Big Data: A Brief Investigation on Different Privacy Issues*, "International Journal of Innovations & Advancement in Computer Science", Vol. 4, Issue 1, Jan 2015, Delhi, India, pp. 122-129.
- Kirti, Maan P. (2015), *Database for Unstructured, Semistructured data-NoSQL*, "International Journal of Advanced Research in Computer Engineering & Technology", Vol. 4, Issue 2, pp. 466-469.
- Krishnan K. (2013), *Data Warehousing in the Age of Big Data*, Morgan Kaufmann, Elsevier, Amsterdam.
- Kumar Shukla R., Pandey P., Kumar V. (2015), *Big Data Frameworks: At a Glance*, "International Journal of Innovations & Advancement in Computer Science", Vol. 4, Issue 1, Delhi, India, 169-175.
- Kumar V., Babu A. (2015), *Domain Suitable Graph Database Selection: A Preliminary Report*, 3rd International Conference on Advances in Engineering Sciences & Applied Mathematics, London, UK, pp. 26-29.

- Lake P., Crowther P. (2013), *Concise Guide to Databases, A Practical Introduction*, Springer-Verlag, London.
- Liu Ch., Fu Z., Yang Z., Xiu J. (2015), *General Research on Database Migration from RDBMS to HBase*, International Symposium on Computers & Informatics, Atlantis Press, Paris, France, pp. 507-514.
- Naresh K.N.V.S., Vani D.Y. (2015), *A New Trusted Security System for Cloud Computing Infrastructure*, "International Journal of Emerging Technology & Research", Vol. 2, Issue 1, Bhimavaram, India, pp. 62-69.
- Nataraja Sekhar G., Saritha S.S.J., Penchalaiah C. (2015), *HBase Performance Testing On Multi-node Cluster Setup*, "International Journal Of Engineering And Computer Science", Vol. 4, Issue 4, April, Mandasaur, India, pp. 11272-11278.
- Niewiadomski A., Strobin Ł. (2015), *Wielokrotne przyspieszenie działania aplikacji poprzez zastosowanie technologii nierelacyjnych baz danych*, Politechnika Łódzka, Łódź.
- Nitnaware Ch., Khan A. (2015), *Comparative Analysis of HBase Data Storage Model and RDBMS for Location based Application*, "International Journal of Science and Research", Vol. 4, Issue 4, April, Nagpur, India, pp. 1328-1330.
- NoSQL, <http://nosql-database.org/> (accessed: 12.05.2015).
- Ntarmos N., Patlakas I., Triantafillou P. (2014), *Rank Join Queries in NoSQL Databases*, "Proceedings of the VLDB Endowment", Vol. 7, No. 7, pp. 493-504.
- Pickard A.J. (2007), *Research Methods in Information*, Facet Publishing, London.
- Pothuganti A. (2015), *Big Data Analytics: Hadoop-Map Reduce & NoSQL Databases*, "International Journal of Computer Science and Information Technologies", Vol. 6 (1), Telangana, India, pp. 522-527.
- Rajput S.H., Jain A.S., Patil P.B., Patil M.D. (2015), *Mongo DB GUI Operation Using Python*, "International Journal of Advanced Research in Computer and Communication Engineering", Vol. 4, Issue 3, Chennai, Tamil Nadu, India, pp. 122-123.
- Ron A., Shulman-Peleg A., Bronshtein E. (2015), *No SQL, No Injection? Examining NoSQL Security*, IBM, Beer Sheva, Israel.
- Sadalage P., Fowler M. (2013), *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Pearson Education – Addison Wesley, Boston.
- Saur K., Dumitras T., Hicks M. (2015), *Evolving NoSQL Databases Without Downtime*, College Park, USA.
- Shang S., Shi M., Shang W. (2015), *A Healthy Aging Real-Time Monitoring System Based on Data Mining*, International Conference on Automation, Mechanical Control and Computational Engineering, Beijing, China, pp. 47-52.
- Shao B., Conrad T. (2015), *Are NoSQL Data Stores Useful for Bioinformatics Researchers?*, "International Journal on Recent and Innovation Trends in Computing and Communication", Vol. 3(3), Bikaner, India, pp. 1704-1708.

- Shibata T., Kurachi Y. (2015), *Big Data Analysis Solution for Driving Innovation in On-site Decision Making*, "Fujitsu Science Technology Journal", Vol. 51, No. 2, pp. 33-41.
- Tathe G.A., Patil P.S., Parle S.C., Pathare V.S., Deshmukh S.S. (2015), *Hadoop Based Analytics on Next Generation Medicare System*, "International Journal of Computer Science and Information Technologies", Vol. 6(2), pp. 1451-1453.
- Tinkhede S.A., Deshpande S.P. (2015), *Big Data – The Vast Growing Technology with its Challenges and Solutions*, "International Journal of Computer Science and Mobile Applications", Vol. 3, Issue 1, pp. 33-38.
- Wassan J.T. (2014), *Discovering Big Data Modelling for Educational World*, IETC, University of Delhi, New Delhi, India.
- Wu C.M., Huang Y.F., Lee J. (2015), *Comparisons Between MongoDB and MS-SQL Databases on the TWC Website*, "American Journal of Software Engineering and Applications", Vol. 4(2), USA, pp. 35-41.
- Yan Y.B., Leng Y.B., Lai L.W., Chen Z.C., Shanghai H.L.G. (2014), *Beam Data Logging System Base on Nosql Database at Ssrf*, Proceedings of PCaPAC2014, Karlsruhe, Germany, pp. 185-187.

STUDIA LITERATURY DOTYCZĄCEJ PROBLEMU BAZ DANYCH NOSQL

Streszczenie: W artykule autorzy zaproponowali wykonanie obszernych studiów literatury dla przedstawienia konstrukcji środowiska NoSQL, opisali problem i wskazali odnośniki do literatury. W pierwszej części przedstawili metodę badań i opisali etapy studiów literatury jako metody badań. Następnie przedstawili przedmiot badań, zidentyfikowali i opisali główne problemy badawcze. Wyniki badań przedstawili w postaci mapy myśli. Wykorzystali ku temu narzędzie otwarte, tj. Mind Map Maker. Ocenili wyniki i wskazali możliwe dalsze prace badawcze.

Słowa kluczowe: NoSQL, mapy myśli, studia literaturowe, narzędzie Mind Map Maker.