Saving Large Semantic Data in Cloud: A Survey of the Main DBaaS Solutions

Bogdan IANCU, Tiberiu Marian GEORGESCU
The Bucharest University of Economic Studies, Romania
bogdan.iancu@ie.ase.ro, tiberiugeorgescu@ase.ro

In the last decades, the evolution of ICT has been spectacular, having a major impact on all the other sectors of activity. New technologies have emerged, coming up with solutions to existing problems and opening up new opportunities. This article discusses solutions that combine big data, semantic web and cloud computing technologies. The authors analyze various possibilities of storing large volumes of data in triplestore databases, which are currently the matter of choice for storing semantic web data. The paper first presents the existing solutions for installing triplestores on the premises and then focuses on triplestores as DBaaS (in cloud). Comparative analyzes are made between the various identified solutions. This paper provides useful means for choosing the most appropriate database solution for semantic web data representation, both on premises or as DBaaS.

Keywords: *Big Data, Cloud computing, Semantic Web*

1 Introduction

One of the main limitations of the World Wide Web (abbreviated WWW) is that it was not designed to be machine-readable, but only human-understandable. In 1994, at the very first International WWW Conference, five years after he invented the World Wide Web, Tim Berners-Lee introduced the idea of a semantic web that can be understood by machines [1]. WWW Consortium (abbreviated W3C) is an international organization which aims to develop standards for the WWW. W3C was founded by Tim Berners-Lee at the same conference where he announced the need for semantic web [1]. Ever since, one of the objectives of W3C is to improve the WWW, by upgrading it from a web of documents to a web of data. [2]

Until the end of the 20th century, the work on semantic web was mostly theoretical; no practical application emerged as an international standard. The first important practical approach consisted in microformats. Instead of using HTML tags for their usual purpose, microformats can be sent as metadata, annexing information understandable by machines. In the meantime, new standards have been developed (e.g. RDF, RDFS, OWL, SPRAQL and JSON-LD) as well as new approaches. [3] In this paper, the authors describe the current state of semantic web, present the main approaches used for semantic web and discuss the most important semantic web solutions for cloud computing. The aim of this article is to present and compare the main solutions for saving semantic data in cloud.

The main semantic web representations are currently based on the semantic triple. A triple is a statement that links two objects, and follows the rule Subject-Predicate-Object. Every part of the triple has a uniform resource identifier (URI) associated with it. Semantic data can be stored as a large graph, where the subject and object are represented as nodes connected by a predicate, in the form of an edge. Any information can be represented by this simplistic model. The maximum potential of this approach can be reached if all the resources on the WWW have an associated URI and are connected with as many other resources as possible by triples. In this scenario, the WWW becomes a large unified database where information from more websites can be automatically extracted and correlated by simple queries. Table 1 shows the format of an URI and describes its components.

DOI: 10.12948/issn14531305/22.1.2018.01

Table 1. URI schema

schema”://[user “@”]host[:port][/path][“?”query][“#fragment”]	
The description of an URI’s components	
Components	**Description**
schema	Access type (HTTP, HTTPS, etc.)
user	Credential used for accessing the resource
host	Domain name
port	Communication channel
path	Can contain parameters
query	Specifies the parameters
fragment	Specifies the required parts of the document

The standard for representing a semantic triple is the Resource Description Framework (abbreviated RDF). Through RDF, a triple can be represented as a succession of three URIs. There are more ways of representing triples with RDF, called serialization formats. The description language varies from one serialization format to another. The main serialization formats are RDF/XML and JSON-LD. However, Turtle (abbreviation of Terse RDF Triple Language) and N-Triples are worth mentioning, as they can be easily understood by humans. Both RDF/XML and JSON-LD serialization are based on popular standards used to represent and transfer data between applications. XML (abbreviation from Extensible Markup Language) has the advantage of being easy to interpret by applications, but in general it is considered a difficult writing format. JSON (abbreviation of JavaScript Object Notation) is used to represent data structures and transmit them between applications. All-important programming languages support JSON formats. JSON-LD (abbreviation from JSON – Linked Data) is designed to facilitate the representation of RDF relationships. [4]

Table 2. The components of an ontology

Components	Description
Classes	Collections, sets, types of objects representing an entity in an ontology
Individuals	Class instances
Relations	Describe how classes and individuals are related
Attributes	Properties or parameters of a class or individual
Restrictions	Dependencies between classes, restricting the set of valid arguments
Rules	“If-then” statements
Axioms	Logical assertions
Events	Changes of values of attributes or relationships

RDF triples are implemented with associated models named ontologies. Ontologies are designed with sets of rules, terms and vocabularies. These sets provide definitions of the entities found in reality. Ontologies are used to develop a large number of applications in different areas, such as knowledge management, intelligent information integration, information retrieval, natural language processing, database design and integration, e-commerce, bioinformatics and education [5]. The main components of an ontology are illustrated in Table 2.

RDFS (RDF Schema) and OWL (abbreviation of Web Ontology Language) are semantic tools used to represent ontologies. They are also called description languages, and define classes and attributes of URIs and their relationships.

2 Semantic Web Databases - Triplestores

A triplestore is a database built to store semantic web data in the form of semantic triples. As RDF is the standard used for Semantic Web, triplestores are also called RDF stores.

This section presents the main issues that should be considered when choosing a database for representing RDF triples. The authors describe the main triplestores available nowadays and present a comparative analysis of the existing solutions. Survey [6] presents the state of art of triplestores in 2014 and performs a qualitative analysis of the main RDF stores available at the time of writing. A similar approach is carried on in this paper, but with an update of criteria used and inclusion of the approaches and solutions developed in the meantime.

To be taken into account for the analysis performed in this paper, RDF stores should include the following components: repository and middleware. A repository is a database constructed or adapted for storing and managing RDF triples. As RDF triples are simple data structures and their XML and JSON serializations are compatible with many programming languages, almost any database can be remodelled to store RDF triples. Therefore, to be regarded as a triplestore and to be able to easily communicate with the repository, the RDF database has to include middleware components (e.g. query engine, RDF parser, APIs for connecting with other applications and storage provider). Triplestores can be implemented both in traditional (SQL) or NoSQL database. NoSQL paradigm seems more suitable for storing and managing RDF triples, since it offers more flexibility. The main NoSQL types of models are: document, key-value, column and graph. RDF stores can be implemented in all these models, but the most reliable one is the graph. [6]

The main technical characteristics to be considered when choosing a triplestore are:

- Semantic web standards supported (e.g. RDF, RDFS, OWL, SPARQL);
- Programming languages supported for connecting the triplestore with other applications;
- Support to reasoning – the ability to make logical inferences based on a set of facts and axioms, thus acquiring new knowledge;
- Licenses – both commercial and open source solutions are considered;
- Last release date – is very important, as old triplestore may not be compatible with more recent applications;
- Operating systems which are compatible with the triplestore.

Table 3 illustrates the main triplestores available in the present together with their technical characteristics.

Table 3. Technical Characteristics of Triplestores

Name	Semantic Web standards	Programming Languages	Support to reasoning	License	Last Release Date	Operating system
3Store	RDF, SPARQL	C, Lisp	No	Open source	July 2006	Linux, Mac, Windows
4Store	RDF	-	No	Open source	September 2009	Linux/Unix
AllegroGraph RDF Store	RDF, SPARQ, RDFS, OWL, RDFa	Java, LISP, Python, Prolog, C#, Ruby, Perl	Yes	Commercial	March 2018	Linux
Blazegraph (Formerly Bigdata®)	RDF, SPARQ, RDFS, OWL	Java	Yes	Commercial, Free	August 2016	Linux, Mac, Windows
Dydra	RDF, SPARQL	Java, LISP, Python, C++, Ruby,	Yes	Free	September 2017	Linux
GraphDB (former OWLIM)	RDF, RDFS, SPARQL,OWL	Java, C#	Yes	Commercial, Free	2018	Linux, Mac, Windows
Semantics.Server/Intellidimension	RDF, RDFS, OWL,SPARQL	ASP.NET, Microsoft Silverlight	Yes	Commercial, Free	June 2010	Linux, Mac, Windows
MarkLogic	RDF, SPARQL	Java, JavaScript, XQuery	Yes	Commercial, Free	2018	Linux, Mac, Windows
Marmotta (Kiwi Triplestore)	RDF, SPARQ, SKOS	Java	Yes	Open source	December 2014	Linux, Mac, Windows
Intelligent Topic Manager	SPARQL, OWL, SKOS	Java	Yes	Commercial	2018	Linux, Mac, Windows
Mulgara	RDF, SPARQ, SKOS	Java	Yes	Open Source	January 2017	Linux, Windows
Open Anzo	RDF, SPARQ, OWL	Javascript, Java, .Net	Yes	Open Source	2011	Linux, Windows
OpenLink Virtuoso	RDF, RDFS, SPARQL,OWL, GRDDL,RDFa, RDB2RDF, R2RML, Direct Mapping	C, C++, Python, PHP, Java, Javascript, C#, ActionScript, Tcl, Perl, Ruby, Obj-C	Yes	Commercial, free	April 2016	Linux, Mac, Windows
StrixDB	RDF, SPARQL	C, C++, Lua	Yes	Commercial, free	November 2010	Linux, Windows
Parliament	RDF, SPARQ, OWL, RDFS	Java, C	Yes	Open source	January 2016	Linux, Mac, Windows
Profium Sense	RDF, RDFS, SPARQL,OWL	Java, Javascript	Yes	Commercial	2018	Linux, Mac, Windows
RDFox	RDF, OWL, SPARQL	C++, Java	Yes	Open Source	2015	Linux, Mac, Windows
RDFD Store	RDF, SPARQL	Perl, C	No	Open Source	June 2006	Linux

DOI: 10.12948/issn14531305/22.1.2018.01

Rdfstore-js	RDF, SPARQL	Javascript	No	Open Source	September 2016	NA
RDF4 (former Sesame)	RDF, RDFS, SPARQL	Java, Python, PHP	Yes	Open Source	December 2017	Linux, Mac, Windows
Stardog	RDF, SPARQ, OWL	Java, Groovy, Clojure, .Net	Yes	Commercial, free	2018	Linux, Mac, Windows
Strabon	RDF, SPARQL, Geospatial, stSPARQL, GeoSPARQL	Java	No	Open source	March 2018	Linux, Windows
Apache Jena - TDB	RDF, RDFS, SPARQL, OWL, GRDDL	Java	Yes	Open source	2018	Linux, Mac, Windows
Oracle Database 12*c*	RDF, SPARQL, RDFS, OWL	Java	Yes	Commercial, Free	2018	Linux, Windows

3 Choosing the best triplestore solutions

The authors compared the data gathered for each triplestore in order to decide which solution is the best to use for storing semantic web data at the moment. Each criterion selected above has to be considered, but obviously not all of them are equally important.

Taking into account that IT technologies evolve very fast nowadays, the authors agree that last release date is can be the starting point for filtering the data. If the release date is not recent there are two issues to consider:

- The possibility that some solutions will not work together due to versions incompatibilities.
- It is less likely that the solution still has support.

The authors consider that the minimum requirement for a medium to big project should be to have the release date in 2017 or more recent and for a small to medium project from 2015 until present. Given these assumptions, there are 10 triplestores to be considered for a medium to big project and 14 for a small to medium one.

The type of license is of course important when choosing a solution. While triplestores with open source licenses offer the advantages of flexibility and the possibility of lower costs (or none), the ones with commercial licenses are more likely to have good support.

To offer good flexibility in implementation, the authors consider that the minimum requirements regarding semantic web standards supported by the triplestores should be: RDF, OWL and SPARQL.

The reasoner is an essential component in semantic web. As can be observed in Table 3, most of the triplestores have this type of tool included.

Having more programming languages available can be useful when accessing and managing store contents from local or semantic client application. On the other hand, all modern triplestores are compatible with at least one major programming language, which can be sufficient if proper specialists are engaged in the project.

The authors consider the operating system as the least important criterion chosen. All triplestores are Linux compatible, which should be good enough considering that most of the servers run on Linux distributions.

Considering the above, the authors consider the top triplestores available nowadays to be: AllegroGraph RDF Store, GraphDB (former OWLIM), MarkLogic, Mulgara, Profium Sense, RDF4 (former Sesame), Stardog, Apache Jena–TDB and Oracle Database 12c, in no specific order. Other studies came to similar conclusions, such as [6] or [7]. Future tests for evaluating reasoning capabilities, security and loading speed can further reduce the list of best triplestore. Such tests to compare performances between triplestores have been performed on some of the solutions identified [8] [9].

4 Short presentation of the main DBaaS

All the major cloud computing providers (the authors are talking here about the big four: Amazon, Microsoft, Google and IBM - see Figure 1 for details) have in their portfolios different DBMSs available in cloud as Databases as a Service. The advantages of these systems are that they can be much safer than a small-medium company could keep them and that they are scalable. This means that the customer is paying just for what he uses and once the business or the stored data grows, the customer will pay more and more without worrying about data migration, switching technologies or buying new servers. All the major providers have in their portfolios both SQL and NoSQL databases, but because this paper focuses mainly on the storage of semantic data, which is known to be unstructured, the authors will present in detail just the NoSQL databases available in cloud for each one of the providers.

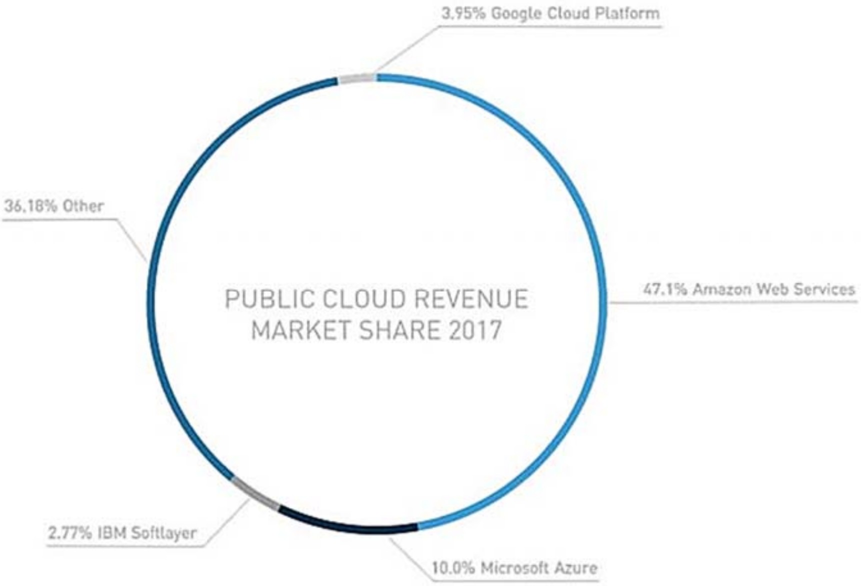


Fig. 1. Public cloud platforms market share (source: https://www.skyhighnetworks.com/cloud-security-blog/microsoft-azure-closes-iaas-adoption-gap-with-amazon-aws/)

4.1 Amazon

The cloud computing platform from Amazon, called Amazon Web Services (AWS) gives the user five different DBMSs to choose from: Amazon RDS, Amazon Redshift, Amazon DynamoDB, Amazon ElastiCache and Amazon Neptune. Even though, non-relational database systems are just DynamoDB, ElastiCache and Neptune (see Figure 2).

DOI: 10.12948/issn14531305/22.1.2018.01

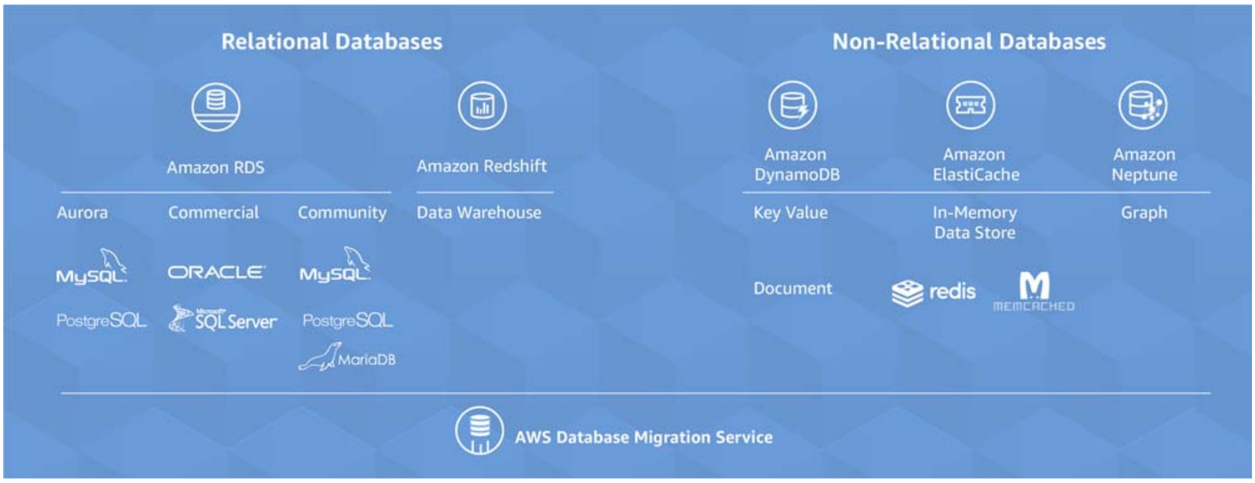


Fig. 2. Databases available in Azure Web Services (source: https://aws.amazon.com/products/databases/?nc2=h_l3_db)

DynamoDB is a fast document based or key-value store (the user can switch between those two) that uses SSD technologies, auto-scaling and auto-management. It is used by big names in the industry like Airbnb, Lyft and Netflix and accessible directly from code as a local database [10].

ElastiCache is Amazon's in memory data store service. It is fully compatible with Redis and Memcached and provides real-time data processing. It is used by companies like Airbnb and McDonald's [11].

Neptune, on the other side, is a graph database at its core and the only DBaaS that natively supports W3C standards like RDF or SPARQL. Like the other ones from Amazon, it is auto-scalable, fully-managed and ACID compliant [12].

4.2 Microsoft

Microsoft's cloud computing platform, called Azure, offers right now the following database solutions: Azure SQL Database, Azure Database for MySQL, Azure Database for PostgreSQL, SQL Data Warehouse, Azure Cosmos DB, Table Storage, Redis Cache and DataFactory. Of these, the authors focused just on Cosmos DB and Table Storage because the other ones are dedicated only to structured, hybrid data or caching.

Cosmos DB is a NoSQL database dedicated for low latency systems. It can be used as a key-value, graph, column family or document store, all-in-one. It is also compatible with APIs like SQL, JavaScript, Gremlin, MongoDB, Apache Cassandra and Azure Table Storage. [13]

Table Storage is a key-value store that uses semi-structured datasets to store large amounts of data. Client libraries for .NET, Java, Android, C++, Node.js, PHP, Ruby and Python are provided by Microsoft, but any other language that accepts HTTP requests is compatible due to the RESTful API. [14]

4.3 Google

The cloud computing platform from Google, called generically Google Cloud, provides the user with a variety of six different storage technologies: Cloud Storage, Cloud SQL, Cloud Bigtable, Cloud Spanner, Cloud Datastore and Persistent Disk. From those just Bigtable and Datastore are NoSQL databases, the other ones provide just SQL-based engines or simple file storage.

Cloud Bigtable is Google's NoSQL Big Data database service. According to them it is the same database that is used for the Search, Analytics, Maps, and Gmail products [15]. Its features include high performance for big data, security (the content of the database is encrypted), auto-scaling and automatic rebalancing of the data when cluster nodes are deleted. It is also fully compatible with Hadoop and HBase API.

Cloud Datastore is a highly-scalable NoSQL database for applications, that don't save necessary very large amounts of data [16]. It

DOI: 10.12948/issn14531305/22.1.2018.01

is very easy to use and provides also a RESTful API for data access. As for Bigtable, Google is in charge with the auto-scaling and management of the database. Another important feature of Datastore is that it is ACID compliant.

4.4 IBM

IBM's cloud platform called simple Cloud (formerly Bluemix) offers four main database engines [17]: Cloudant, Compose, Db2 and Informix. If Cloudant is a simple NoSQL JSON document store optimized for heavy I/O operations, Compose is more like a cloud database engine that can work with different DBMSs like Elasticsearch, etcd, JanusGraph, MongoDB, MySQL, PostgreSQL, Redis, RethinkDB and ScyllaDB. On the other hand, Db2 and Informix are dedicated specially to structured data (the case of the first one) or mixed data (TimeSeries, Spacial, SQL and NoSQL data altogether – the case of the second one).

Cloudant is a JSON document store, that is optimized for handling heavy workloads of concurrent reads and writes in the cloud. It can be a good choice for mobile apps that use features that need offline sync. It is accessible as a RESTful API and fully compatible with Apache CouchDB.

Compose for etcd is a key-value store that uses auto-scaling and has a great focus on data consistency. Similar, Compose for MongoDB brings all the advantages of the BSONs in the cloud.

Compose for JanusGraph is a scalable graph database optimized for storing and querying highly-interconnected data. It is not built-in to be compatible with W3C's semantic data formats, but it is with Apache TinkerPop framework.

Table 4 summarizes all the features of the presented databases with a focus on the accepted semantic web technologies.

Table 4. DBaaS systems comparison

DBaaS	Type	Querying languages	RESTful access	Other type of access	W3C's semantic data formats compatible
Amazon DynamoDB	Key-Value or Document Store NoSQL	-	Yes	Libraries for main programming languages	No
Amazon ElastiCache	In-memory NoSQL	-	No	Libraries for main programming languages	No
Amazon Neptune	Graph NoSQL	Gremlin, SPARQL	No	Libraries for main programming languages	Yes
Microsoft Cosmos DB	Multi-model NoSQL	Gremlin, DocumentDB API, Table API	Yes	Dependent of the chosen model	No
Microsoft Table Storage	Wide column NoSQL	-	Yes	Libraries for main programming languages	No
Google Cloud Bigtable	Wide column NoSQL	BigQuery	Yes	HBase API	No
Google Cloud Datastore	Document-Store NoSQL	GQL	Yes	Libraries for main programming languages	No

IBM Cloudant	Document-Store NoSQL	-	Yes	Libraries for some programming languages	No
IBM Compose for etcd	Key-Value NoSQL	-	No	Libraries for main programming languages	No
IBM Compose for MongoDB	Document-Store NoSQL	-	No	Libraries for main programming languages	No
IBM Compose for JanusGraph	Graph NoSQL	Gremlin	No	Libraries for some programming languages	No

5 Picking the winners

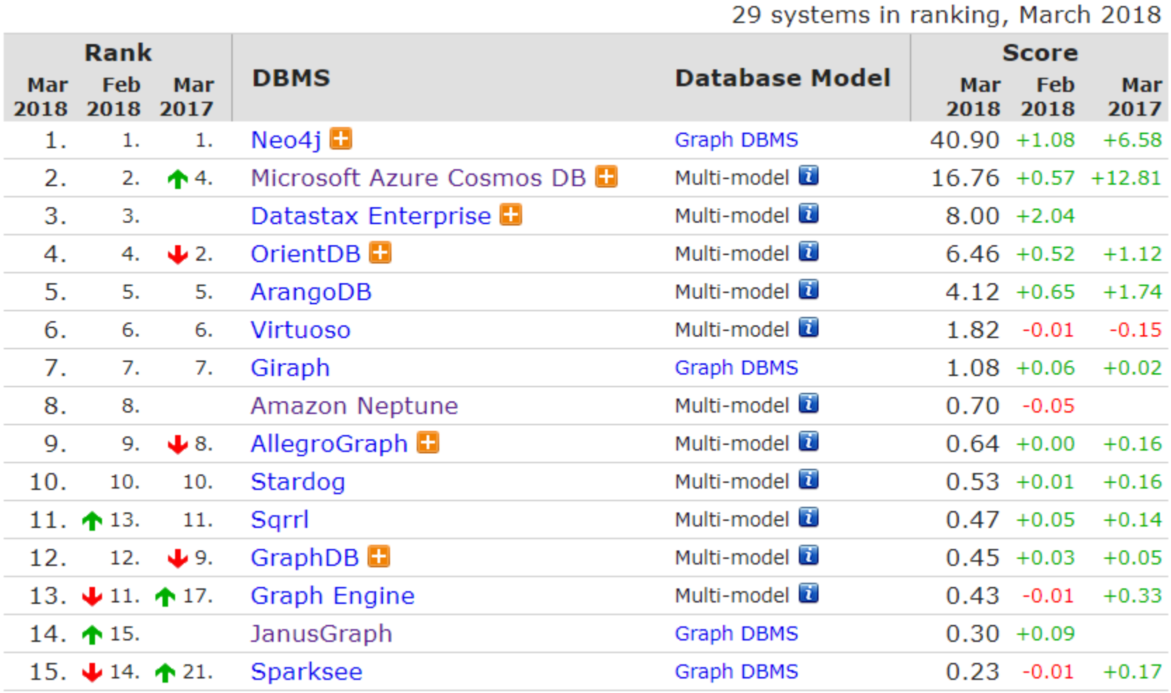
The previous chapter showed that every one of the four big cloud computing providers has the capability to offer to the client a solution for saving semantic data. Now the question that can be raised is what to choose, depending on the cloud provider that one is already using.

In our opinion, Amazon has a clear winner in the name of **Amazon Neptune**. As stated in [18] a graph database usually surpasses in terms of performance a key-value or document-store for large amounts of data, especially if the non-graph database doesn't use MapReduce techniques. Additionally, Neptune supports both W3C standards (RDF 1.1 and SPARQL 1.1) and the Apache TinkerPop framework and is ranked 4th by the DB-Engines Ranking in the category of RDF stores based on its popularity [7]. It is also placed 8th in the graph databases category (Figure 3).

From Microsoft, the winner is **Microsoft Cosmos DB**. Even if it is a hybrid type of NoSQL, it can be configured to work as a graph database that supports the Apache TinkerPop framework. It is ranked 2nd in the DB-Engines Ranking in the category of graph databases [19] and not present in the RDF stores categories because it isn't directly compatible with this format.

Google Cloud doesn't provide yet any type of graph database, so in the case of this platform it is necessary to choose between the existing NoSQL databases. Our choice is **Google Cloud Bigtable** because it is designed for large amounts of data and also supports MapReduce techniques that can significantly reduce the querying time [18]. Because Bigtable is neither a graph database nor a RDF store, it is possible to find just in the overall ranking of DB-Engines where it is placed 125th [19].

In the case of IBM Cloud, the authors picked **IBM Compose for JanusGraph** as the winner because it is the only NoSQL database from IBM that is a graph-based and because it is compatible with the Apache TinkerPop framework. JanusGraph is ranked 14th in the graph databases category by the DB-Engines Ranking (Figure 3).



29 systems in ranking, March 2018

Rank			DBMS	Database Model	Score		
Mar 2018	Feb 2018	Mar 2017			Mar 2018	Feb 2018	Mar 2017
1.	1.	1.	Neo4j	Graph DBMS	40.90	+1.08	+6.58
2.	2.	↑ 4.	Microsoft Azure Cosmos DB	Multi-model	16.76	+0.57	+12.81
3.	3.		Datastax Enterprise	Multi-model	8.00	+2.04	
4.	4.	↓ 2.	OrientDB	Multi-model	6.46	+0.52	+1.12
5.	5.	5.	ArangoDB	Multi-model	4.12	+0.65	+1.74
6.	6.	6.	Virtuoso	Multi-model	1.82	-0.01	-0.15
7.	7.	7.	Giraph	Graph DBMS	1.08	+0.06	+0.02
8.	8.		Amazon Neptune	Multi-model	0.70	-0.05	
9.	9.	↓ 8.	AllegroGraph	Multi-model	0.64	+0.00	+0.16
10.	10.	10.	Stardog	Multi-model	0.53	+0.01	+0.16
11.	↑ 13.	11.	Sqrrl	Multi-model	0.47	+0.05	+0.14
12.	12.	↓ 9.	GraphDB	Multi-model	0.45	+0.03	+0.05
13.	↓ 11.	↑ 17.	Graph Engine	Multi-model	0.43	-0.01	+0.33
14.	↑ 15.		JanusGraph	Graph DBMS	0.30	+0.09	
15.	↓ 14.	↑ 21.	Sparksee	Graph DBMS	0.23	-0.01	+0.17

Fig. 3. DB-Engines Ranking for Graph Databases in March 2018

Based on the DB-Engines Rankings and on the supported technologies it can be considered that the overall winner could be Amazon Neptune, but in the absence of a benchmark test and of a monthly average price survey of the whole group of analyzed database, it can be a subjective choice.

6 Conclusions and future work

In this article, the authors aimed to identify what DBaaS solution from the large cloud computing providers (Amazon, Microsoft, Google and IBM) can be considered when it comes to save large amounts of semantic data. After the semantic web technologies and domain standards were presented, the authors have made a comparative analysis of the existing triplestores (databases specially created for saving semantic data) available on premises. Unfortunately, none of them is present in the portfolios of the large cloud providers and many of them are not up-to-date. For this reason, the NoSQL databases available as DBaaS were analysed and the authors tried to figure out which one of them could be the best match when it comes to save large quantities of semantic data. Even though Amazon Neptune seems to be the overall winner, the authors will focus their future work on doing a benchmark testing on all the identified databases together with a cost analysis.

Acknowledgement

This paper was partially supported by Școala de Valori Association under the corporate philanthropy agreement number 176/21.03.2018 concluded in the "Cegeka Digital Youth" program.

References

[1] T. Berners-Lee, “W3 future directions,” in International World Wide Web Conference, https://www.w3.org/Talks/WWW94Tim/, CERN, Geneva, 1994.

[2] World Wide Web Consortium, “Semantic Web,” W3C, 2018. [Online]. Available: https://www.w3.org/standards/semanticweb/. [Accessed March 2018].

[3] L. a. A. M. Kaur, “Software component and the semantic Web: An in-depth content analysis and integration history,” Journal of Systems and Software, vol. 125, pp. 152-169, 2017.

[4] P. K. M. R. S. Hitzler, Foundations of semantic web technologies, CRC press, 2009.

[5] L. B. A. S. I. &. M. I. M. Băjenaru, “Abordare bazată pe Ontologii a unui sistem de E-Learning în Domeniul Managementul Resurselor Umane într-un Spital Universitar,” Revista Română de Interacţiune Om-Calculator, vol. 8, no. 2, pp. 139-156, 2015.

[6] G. E. M. S. T. Modoni, “A survey of RDF store solutions,” in 2014 International ICE Conference on. IEEE, 2014.

[7] SolidIT Consulting & Software Development Gmbh, “DB-Engines Ranking - popularity ranking of RDF stores,” SolidIT, March 2018. [Online]. Available: https://db-engines.com/en/ranking/rdf+store. [Accessed March 2018].

[8] F. L. e. a. da Rosa, “Performance evaluation of SPARQL queries in triplestore,” in Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres, 2017.

[9] V. N. B. a. I. C. Kilintzis, “Evaluation of the performance of open-source RDBMS and triplestores for storing medical data over a web service.,” in Engineering in Medicine and Biology Society (EMBC), 36th Annual International Conference of the IEEE, 2014.

[10 S. Sivasubramanian, “Amazon dynamoDB: a seamlessly scalable non-relational database service,” in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, 2012.

[11 Amazon.com, “Amazon ElastiCache- In-memory data store and cache,” Amazon, March 2018. [Online]. Available: https://aws.amazon.com/elasticache/. [Accessed March 2018].

[12 Amazon.com, “Amazon Neptune - Fast, Reliable Graph Database built for the cloud,” Amazon, March 2018. [Online]. Available: https://aws.amazon.com/neptune/. [Accessed March 2018].

[13 Microsoft Corp., “Azure Cosmos DB Documentation,” Microsoft, March 2018. [Online]. Available: https://docs.microsoft.com/en-us/azure/cosmos-db/. [Accessed March 2018].

[14 Microsoft Corp., “Table Storage | Microsoft Azure,” Microsoft, March 2018. [Online]. Available: https://azure.microsoft.com/en-us/services/storage/tables/. [Accessed March 2018].

[15 Google Inc., “Bigtable - Scalable NoSQL Database Service,” Google, March 2018. [Online]. Available: https://cloud.google.com/bigtable/. [Accessed March 2018].

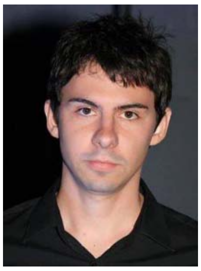
[16 Google Inc., “Datastore - NoSQL Schemaless Database,” Google, March 2018. [Online]. Available: https://cloud.google.com/datastore/. [Accessed March 2018].

[17 International Business Machines Corporation, “Data management services | IBM Cloud,” IBM, March 2018. [Online]. Available: https://www.ibm.com/cloud/data-management. [Accessed March 2018].

[18 P. e. a. Cudré-Mauroux, “NoSQL databases for RDF: an empirical evaluation,” in International Semantic Web Conference. Springer, Berlin, 2013.

[19 SolidIT Consulting & Software Development Gmbh, “DB-Engines Ranking - popularity ranking of graph DBMS,” SolidIT, March 2018. [Online]. Available: https://db-engines.com/en/ranking/graph+dbms. [Accessed March 2018].

Bogdan IANCU has graduated The Faculty of Cybernetics, Statistics and Economic Informatics from The Bucharest University of Economic Studies in 2010. He has a master's degree in Economic Informatics (2012) and a PhD in Economic Informatics starting from 2015 in the field of Ontologies and eLearning. He is an Assistant Lecturer in The Department of Economic Informatics from The Bucharest University of Economic Studies. His current research focuses on semantic technologies and ontologies innovations. Other fields of interest include machine learning, multimedia, mobile devices and IoT.



Tiberiu-Marian GEORGESCU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2012. In 2015 he has graduated the Informatics Systems for the Management of Economic Resources Master program. Currently he pursues a PhD research in Economic Informatics at the Bucharest University of Economic Studies, guided by professor Ion SMEUREANU, PhD. He is working as a Teaching Assistant in the Department of Economic Informatics and Cybernetics. His main interests in the Informatics field are cybersecurity, artificial intelligence and semantic web.

DOI: 10.12948/issn14531305/22.1.2018.01