A Review on the Evolution of Resource Description Framework

Dr.R Ravinder Reddy[*], Ch Aparnadevi[1], Dr.Y Ramadevi[2]

[*, 1, 2] Department of Computer Science and Engineering, CBIT, Hyderabad 500075, India

*Assistant Professor and Corresponding author, email: ravindra_rkk@cbit.ac.in

[1] Student, M.Tech, email: appu.717amu@gmail.com

[2] Professor and Head, email: yrd@cbit.ac.in

ABSTRACT The enormous growth in computational speed and storage leads us to distributed and cloud computing. Today, the world runs on digital data, with the invention of the web, most of our data and resources are stored in the web. It has a vast amount of data which continues to grow at greater speed. In the past few years, technology has progressed with the increase in data size, creating boundless demand for a place to store and retrieve it. With this rapid increase, the issue of data scalability has become very important and various solutions using several new technologies have been developed to emphasize on scalability and query efficiency. One such standard that has been graded by the World Wide Web Consortium is the Resource Description Framework (RDF). This data model technology can be utilized to build an efficient and scalable system to retrieve the web resources. Cloud Computing, an emerging technology, also seems to offer some incredible benefits related to data scalability and efficiency. A new technique that takes the advantage of both RDF data management system and Cloud is being adopted to resolve the scalability issues, making it easy to apply all the data operations.

Keywords: Data, NoSQL, Resource Description Framework (RDF), web, Cloud, Diplocloud.

1. INTRODUCTION

With the invention of the web, so many problems are solved in the Internet community, But at the same time it creates boundless problems in retrieving the resources inefficiently. The digital technology has increased the amount of data storage in these days. The generated data might be structured or unstructured. This data needs a technology for further references. In the early age of computing, data was saved in the form of flat files. Over the years the trend has changed to databases. With the exponential growth of the internet in 1990s, the web resources have begun to be maintained in huge databases. Maintenance of these resources was a big problem because the web is a very huge collection of the web pages. The extension of web called Semantic web [5] emerged to solve these problems, and to maintain the resources effectively. It provides the framework that allows the huge collection of web pages to reuse and share the resources. One such fundamental framework which gives data a new model is Resource Description Framework (RDF) [23]. This gives a clear idea of maintaining these resources on the web. The same resources can be distributed among the web that is machine processable and understandable. The evolution of web technology has started in the early 90's; the enormous growth of the web has created many challenges in representing the resources in the World Wide Web.

The technological challenges demand an improvement in the current resource management systems respectively in the web, starting from the architectural design to the implementation details. Various solutions were developed to handle large data systems in an improvised manner. Cloud computing can be the base layer of many data systems with few requirements such as elasticity, cost-effectiveness, and the ability to scale up or down. A data management framework called the Resource description framework from NoSQL [4] is concerned about the way the information is organized in a convenient manner for efficient processing. A new technique which takes the advantage of both- NoSQL [4] and Cloud [2] [28] is being adopted to resolve the issue of scalability.

The exponential growth of the computing power and the communication changes in the era of computer industry. The combined power comes in the form of distributed and cloud computing. Using these methodologies we can publish our resources in the web. It changes the storage technology and retrieving the data.

It is clear that we are living in a data deluge stage, given the huge volume of data from a mixture of sources and its growing amount of generation. This survey focuses on scalable and efficient systems, that consists of a set of tools and mechanisms to store, extract, and improve data by taking advantage of power of distributed processing to perform complex conversion and analysis in respect to the web. In this paper we conduct a review in the evolution of web resource management.

2. RELATED WORK

The conventional database management systems (DBMS) provide an environment that is both convenient and efficient for structured data storage. It provides users a procedural way to create, update, retrieve, and manage the data. It also provide the security and protection to the database. But managing the data in the web is different; it contains both the structure as well as unstructured data.

Maintaining databases with the huge amount of data and making it available to others is a vital challenge. The need to boost the efficiency of data transfer in both centralized and decentralized systems, parallel and distributed mechanisms are came into the existence. Although parallel databases have their own advantage in terms of performance, the distributed databases overcome the availability disputes that came from parallel databases. Now, in recent years, traditional data management systems (RDBMS) [9] [11], have failed to deal with challenges related to vast data. Contrary to other systems, Resource Description Framework (RDF) [5], is a data model with both NoSQL [4] and cloud [2] [28] is being adapted to mange vast data. In both parallel and distributed systems this concept has made its mark.

Some of the peculiar old systems motivated the adoption of Resource Description Framework (RDF). A previous GridVine system [14] is used based on the triple-table storage approach with hash-partitioning in a decentralized peer-to-peer networks where RDF data is distributed. A semantic infrastructure is maintained in a P2P access structure. This structured P2P network layer manages data, schemas, and schema mappings that are supports decentralized indexing, key load-balancing, and efficient routing systems which were based on the principle built on data independence. Logical layer is separated from physical layer. The related information is queried transparently by an iterative query reformulation. There is a rapid increase in the structured information that is generated automatically in these days. In order to facilitate a scalable resort that organizes the data in distributed, autonomous, and complex data spaces is currently in the process of emerging.

Storing and querying RDF data facilities has been adopted by many systems, over the semantic web community such as Jena [12], Sesame [13], and TripleProv [15]. A modern system handles queries which efficiently extend a native RDF store. It considers two different storage models with lineage and instance data which are physically co-locate, and trace provenance at two granularity levels by implementing algorithms for each. It has been left out for further work to enhance provenance support in the database system as a distributed version. To optimize memory consumption and query execution time with a dynamic storage model is a challenge.

Virtuoso [16] made known by Erlin et al., as data in the form of RDF quads that contains elements such as: the graph, subject, predicate and the object. The quads that are associated in the form of a single table are partitioned based on the subject. Two indexes are implemented, one default index (set as a primary key) which is GSPO (Graph, Subject, Predicate, Object) and other auxiliary bitmap index (OPGS). Triples are hash partitioned across multiple machines and queries are processed in parallel.

Owens et al. [12] introduced a storage data in three B+-tree indexes. The concept is on SPO, POS, and OSP permutations, where all elements of all triples are managed by index. A query is divided into basic graph patterns (BGP) which are mapped to the stored RDF data. Taking advantage of this graph structure, a number of further approaches have been proposed to store RDF.

Hadoop and MapReducer [3] is a software framework that simplify the process of writing applications dealing with the vast amount of data in parallel which are from large clusters of commodity hardware that are in reliable and fault-tolerant manner. The datasets are divided into independent chunks by the map task in a parallel manner. Now, these independent chunks are sorted and given as input to they reduce task. MapReduce algorithm and Hadoop Framework have become supporting tools for querying on big RDF graph and to perform data transfer. The linked data written in the form of RDF/XML syntax is converted into a huge set of N-triples and uploaded onto Hadoop. They are stored in data nodes of Hadoop Distributed File System (HDFS) [19]. A query in SPARQL [8] [29] where an RDF graph is queried is analyzed and converted into a specific N-triple format so as to obtain the results using Jena Algebra.

3. EVOLUTION OF DATABASES AND WEB

Database management systems are widely differed. Flat files, relational, network, and hierarchical all refer to the way a DBMS organizes information within it. Figure 1 gives an idea of evolution of DBMS that varies throughout the time.

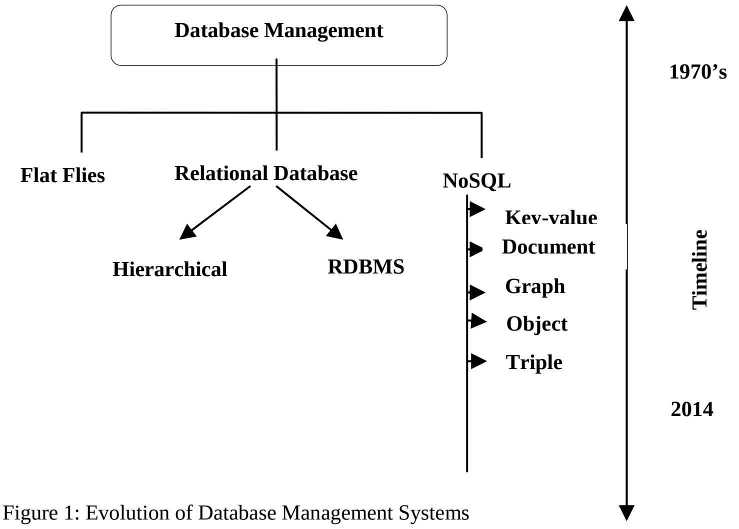

Figure 1: Evolution of Database Management Systems

3.1 Flat Files**:**

This is probably an easiest way to understand, but at present it is rarely used. You assume it as a single huge table. These types of databases were used earlier in 1980s, when data retrieval was only required in case of concerns. As this is a simple way to store files, a flat file system [27] becomes increasingly inefficient if more data is added to the database. The original Macintosh computer uses this kind of file system, creatively called the Macintosh File System (MFS) [20]. To overcome this dispute Hierarchical File System (HFS) [6] came into existence that became more efficient as it was based on a Relational directory structure.

3.2 Relational Database (RDB):

The flat file lags in reducing a lot of redundant data at data entry. A relational database is used store data which contains relationships within them. These are represented in the form of tables with rows and columns. Every table consists of records called 'Tuples'. Each record is classified by an attribute containing a unique value. A query language Structured Query language (SQL), used for database interaction was known to create relational databases.

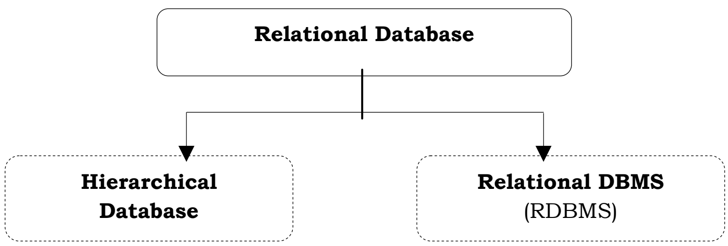

Figure 2: RDB Structure

3.2.1 Hierarchical Database and RDBMS:

Hierarchical Database [6] is very similar to our folder structure which has related records together like a family tree. Every folder contains a sub-folder and each sub-folder can still hold more sub-folders. At the end, files will be in either of the sub-folders. Though these databases are simple, there are some restrictions like they cannot be used often to relate structures that exist in the real world. Insertions and deletions can be very complex in child level as access to it is only through the parent. This restriction extends the relational database that is shown in figure 2. Hierarchical databases with one-one mapping data structures were used to overcome this. A different structure which allows one-many mapping such as the table is known as the Relational database management system (RDBMS) [9] [11]. For many-many mapping, Network databases came into existence.

3.3 NoSQL Database**:**

NoSQL [4] means Not Only SQL, over the last few years there is a rise of a new type of databases, known as NoSQL databases. It provides different types such as:

- Key-value store [17]: key value pairs
- Document store [18]: XML
- Graph store: Graphs
- Object database [7]: Data as Objects

– Triple/quad store [10] : RDF

Key Value Databases [17]: A key of a key/value pair can be a unique value of the set and are easily looked up to approach the data.

Document Oriented Databases [18]: These databases consider a document as a whole and do not split the document into constituent name/value pairs. Document databases allow indexing of documents by primary identifier and also on the basis its properties.

Graph Based Databases: Graph structures are used in graph database with nodes and edges that represent and store data. A graph database provides index-free adjacency in any storage system. By this each, element consists of a direct pointer to its neighboring element which does not require index lookups.

Table 1: Differences between the Databases

DATABASES	AUTHORS	YEAR	PRO'S AND CON'S
Flat Files	Herman Hollerith	1980	+ Easy, Simple. – Redundancy.
Relational	Edgar F. Codd	1990	+ Flexibility. – Physical Storage Consumption.
Hierarchical	IBM	1992	+ Speed of access. – Predefined path.
NoSQL	Carlo Strozzi	1998	+ Support for unstructured data. – No Relations between tables.

The relational database created by E.F. Codd solved the dispute on shortage of standard way to store data. But, thereafter relational databases also got into trouble of not handling big data, due to this complication there was a need of a database which can handle every type of problems, then NoSQL database was developed. These databases tend to support unstructured data which was the beneficial element that lead in the development of the web. With the introduction of NoSQL other databases flawed by their own disputes, comparison of all the databases are represented in Table 1.

"WWW" abbreviated as World Wide Web is the one which lets people browse data or documents that may be structured or unstructured connected with hyperlinks. The concept of web was by a group of researchers included "Tim Berners Lee" in 1991 who is now called as the father of the web. The percept of the web is to use the hyperlinks and navigate the documents with a program known as browser. A document or simply a web page is written in a markup language called HTML (Hyper Text Markup Language). It encodes the document and links up to the different documents with help of tags. URL's which are a unique address used to identify these documents to point any resources of the web. With the increasing fame and expanded success of the web, demand has risen never before. Web is the most amazing attribute that it is an affably large and self-organized artifact in reality. Day by day millions of web administrators add, delete, move, and change their pages and links.

Although web infrastructure allows us to publish and retrieve documents, added on data makes it difficult to interpret the information. Here is why the Semantic Web [5] came into the picture, today's web is of documents, whereas the semantic web is all about "things" (people, places, events, etc.), and describing the relationships between them. The Semantic Web is built on syntax where URI's is used to represent data, commonly in triples based structures. Most of the triples of URI's that can be held in databases, or exchanged on the web with formats or a set of precise syntax introduced especially for this task. These syntax are called "Resource Description Framework" syntaxes. Syntax is used to represent the web metadata that allow complex and accurate results on the web. All these contents require a database to be accessed easily and storing data regardless of having structured or unstructured data. One such kind is NoSQL system that is being penetrating in the Semantic Web. They offer the great performance and scale that the web-based data, much more flexible than traditional relational systems.

NoSQL includes a wide range of technologies and architecture, which tend to solve the problem of scalability and big data and its performance issues that relational databases couldn't address. The Resource Description Framework (RDF) [5] is one of the frameworks from NoSQL which is increasingly being adapted to model data exchanged on the Web. RDF data enables us to make flexible statements about resources. Resources can uniquely be identified by URI's which are then distributed among the web. The development of RDF is motivated by its crucial characteristics like abstract syntax, formal semantic, simple data model.

4. RESOURCE DESCRIPTION FRAMEWORK (RDF)

As Semantic Web [5] [26] is not replacement of the current web but only its extension, it extends the principle of web from documents to data that can be accessed using general web architectures with URI's. Metadata enhances the access of information by summarizing basic information about data which can make working with specific data easier. More complex, focused web searches are also allowed by metadata to get more accurate results. A Resource Description Framework [RDF] [23] is a W3C proposed standard for interpreting the architecture necessary for supporting web metadata [5]. It using a variety of syntax notations and represented in an XML [20] formats. According to the recent years, data in web is growing in large volume with many storage systems. RDF has been a widespread format for the storage and retrieval of data [33]. As the datasets increase the efficiency of storage and retrieval of data becomes difficult due to its sheer size.

RDF databases can differ from alternate solutions that are available in NoSQL. These databases are standardized one in NoSQL solutions which are built on a simple and uniform data model. They offer data interoperability and flexibility among varies implementations that are being adopted at present. They tend to be inherently distributive and schema-less with decentralized labeled graph. The key differentiator is the system is framed upon linked data technology of W3c's standard. This database framework when compared to ordinary NoSQL database systems [21], have some benefits like:

- Simple and unique data model.
- Triples and URI's.
- Syntax is layered and serialized that allow powerful encoding.
- Schema less technology with self documenting.
- Linked data that provide flexibility.

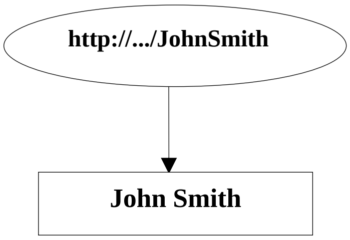

Figure 3: Representation of RDF

Figure 3 represents RDF with resource, property and object. John Smith, in ellipse considered as resource which has a Uniform Resource Identifier (URI) "http://.../JohnSmith". A resource can have many properties from above example "http://.../JohnSmith" can have other properties like business card or telephone etc... Figure 3 shows only one property, John Smith's full name. Properties are represented by an arc, and also labeled related to the resource. URL's can be considered as properties but they are long and inconvenient so they are showed in XML form. The XML syntax is divided in two parts, first one before the ':' is called a namespace prefix and represents a namespace. The second one after the ':' is called a local name and represents a name in that namespace. This syntax is also called RDF/XML which is used to represent the properties as they are convenient script for representing them in text and diagrams. Every property contains a value which is known as literal. Literals here can be string of characters which are considered as objects. These are represented in rectangles.

4.1 RDF Schema

Semantics for RDF data is modeled by a data-modeling vocabulary called RDF Schema [21]. This allows us to describe groups of related resources and relationship between them. Classes contain groups of resources followed by properties which are expressed by RDF vocabulary that is defined with namespace and prefixed as 'rdfs'.

RDF design is expected to meet few goals like:

- extensible URI based vocabulary
- using XML based syntax and data types
- making statements about resources
- make formats to understood easily

4.2 RDF Graph:

The triples in the RDF can be collectively represented in the form of a graph which is called RDF Graph [30]. The nodes are the set of subjects and objects and the relationship between them i.e., the edges connected are considered as predicates. This can be shown in an example; Figure 4 which give the idea of RDF syntax and Figure 5 representing the RDG Graph.

Efficient data storage and query processing that can expand to large amount of possibly schema-less data has become an important issue. RDF data storing [10] [22] can be broadly categorized into three sub-categories: triple-table, property-table, and graph-based approaches.

Triple-Table: RDF data of the form (subject, object, predicate) is stored as a triple in one large table containing three column schemas. In order to reduce joins, indexes are added for each column.

Property-Table: This is an improved RDF data that is organized by allowing multiple triple patterns. These refer to the same subject to be retrieved without an expensive join.

Graph-Table: Statements are represented as graphs with nodes as resources and values. The directed edges within the nodes are considered as properties.

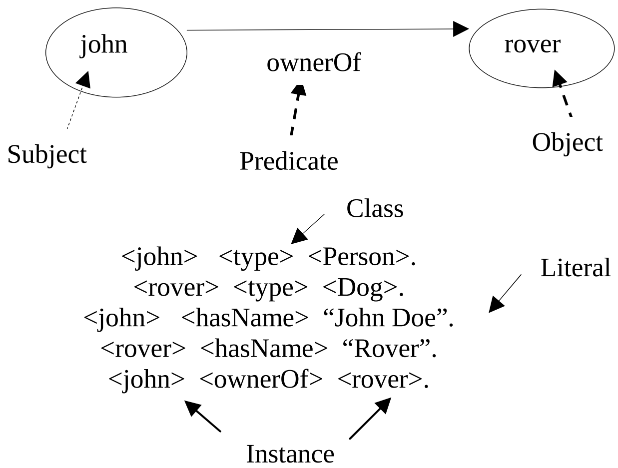

Figure 4: RDF syntax

Managing large volumes of RDF data has become a challenge due to the sheer size, the multiplicity, and the complexity brought by RDF reasoning. These scenarios can be handled through the distributed storage architectures. Cloud computing [2], an emerging technology for the scalability, fault tolerance, and elasticity features is adopted for managing RDF data, thus giving us a way to deploy the distributed and parallel architectures. The systems designed and implemented for a cloud environment are more generally, those large-scale RDF data management systems that can be easily deployed.

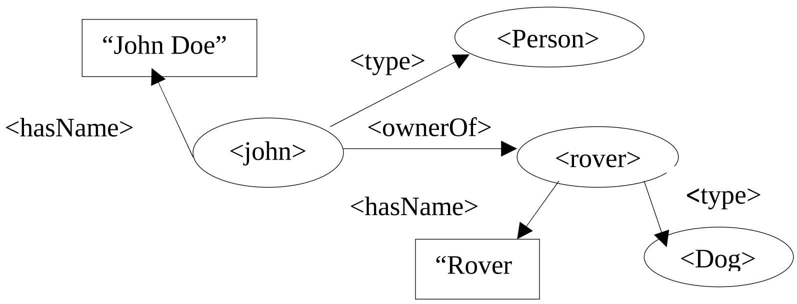

Figure 5: RDF Graph

Although RDF data management [21] has advances in different ways, dealing with large amount of data is still very challenging. Splitting data with simple techniques and managing the graphs with classical min-cut algorithms has become inefficient for distributed operations leading to high number of joins. To resolve these issues with the classical techniques a new system “Diplocloud” is evolved with two main solutions.

5. DIPLOCLOUD: AN RDF DATA MANAGEMENT SYSTEM

Many systems came into existence to introduce and optimize the RDF storage [33] and SPARQL [29] query processing. DiploCloud [1], is one of the recent and efficient models to make RDF data processing easier. It is introduced for distributed and cloud environments. A non-relational storage format is considered which semantically relates to the data. A new model which partitions an RDF graph using both manual and adaptive partition techniques reduces number of joins. Unlike other systems, here the RDF graph is generated on three main structures: RDF molecule clusters, template list and key index. Data is partitioned and a unique key is given to generate a unique URI based on hash technique and processed further. This flow of the system is presented step by step shown in Figure 6. DiploCloud considers physiological analysis of both instance and schema level data. Contrary to other RDF data management systems it adopts new algorithms to partition and distribute data.

The large RDF graphs [31] [32] are often inefficient to manage. To tackle with this problem many useful graph partitioning techniques came into existence. These techniques gave low standard quality which tends to give poor performance. Here RDF graph for the given data a new partitioning techniques for better and efficient results are implemented.

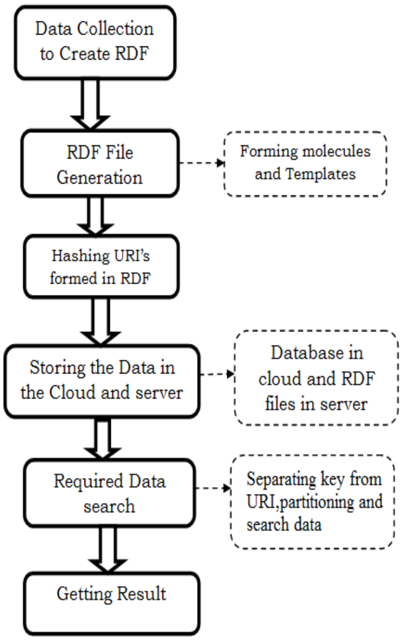

Figure 6: Flow of the system

The two partitioning techniques that are used are:

Manual partitioning: The root nodes specified by the database administrator and also how concatenations of the nodes are done is stated. By this specification a path following which molecules should be physically extended based on can also be handled by the data administrator. To the required extent the path is set manually and then it is extended to further details. As the search [24] can be frequent this partitioning technique is extended to adaptive partitioning.

Adaptive partitioning: This is the most flexible partitioning technique. As said earlier frequent paths can exists while search, in this technique these paths are utilized for better performance. The system manages a sliding-window 'w' keeping the track of the recent history of the search. In this way, the system gently adapts the workload and emerge frequent paths in the RDF graph. The overall size of the molecule is kept small. The advantage of this method is that it automatically fit to the dynamic workload by increasing and decreasing the graph path

The new partitioning technique's that efficiently and effectively processes the RDF data [25] gives the system a new model. Contrary to previous systems this system can handle the efficient performance problem and also manage large data. By adapting the frequent paths it also reduces the search time which is a tremendous advantage to the system. Thus giving a profitable solution to the problems appearing in today's world.

6. CONCLUSION

The sheer volume of information can make searching the web annoying. Resource Description Framework, with its focus on machine-understandable semantics, has the ability to save time and obtain more accurate search results. Metadata is processed by RDF that favor interoperability between applications and help in exchanging machine understandable information on the web. In this survey, the focus is on presenting the state-of-the-art techniques in the area of RDF data management in distributed or parallel era. A classification of the systems is done based on which they implement three fundamental functionalities: data storage, query processing, and result analysis. The results can be obtained in a better way by setting the frequent search according to the required priority as a future work.

References

[1] Marcin Wylot and Philippe Cudre-Mauroux “DiploCloud: Efficient and Scalable Management of RDF data in cloud” in Proc. IEEE Data Engg, Vol. 28, No.3, March 2016.

[2] D.J. Abadi, "Data Management in the Cloud: Limitations and Opportunities", IEEE Data Eng. Bull, vol. 32, no. 1, pp. 3-12, March 2009.

[3] M.F. Husain, P. Doshi and L. Khan, B. Thuraisingham, "Storage and Retrieval of Large RDF Graph Using Hadoop and MapReduce", Proc. First Int'l Conf. Cloud Computing, 2009.

[4] http://nosql-database.org/.

[5] O. Lassila, “Web Metadata: A Matter of Semantics”, IEEE Internet Computing, vol. 2, no. 4, pp. 30-37, 1998.

[6] A. Meier, R. Dippold, J. Mercerat “Hierarchical to relational database migration” IEEE Computer Society, IEEE Software Volume: 11, Issue: 3, May 1994.

[7] D. H. Fishman, "An Object-Oriented Database Management System", ACM Trans. Office Information Systems, vol. 5, no. 1, pp. 48-69, 1987.

[8] E. PrudHommeaux and A. Seaborne, et al., “SPARQL query language for RDF,” W3C Recommendation, vol. 15, 2008.

[9] D. E. Difallah, A. Pavlo, C. Curino, and P. Cudr_e-Mauroux,“OLTP-Bench: An extensible testbed for benchmarking relational databases,” Proc. VLDB Endowment, vol. 7, no. 4, pp. 277-288, 2013.

[10] Faye, O. Cure, and Blin, “A survey of RDF storage approaches,” ARIMA J., vol. 15, pp. 11–35, 2012.

[11] Bala Subramaniam, “DataSecurity in RDBMS” Journal of Computer Science and Security, 2012.

[12] K. Wilkinson, C. Sayers, H. A. Kuno, and D. Reynolds, “Efficient RDF Storage and Retrieval in Jena2,” in Proc. 1st Int. Workshop Semantic Web Databases, pp. 131–150, 2008.

[13] Broekstra, J. Kampman, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", ISWC2002, 2002.

[14] K. Aberer, P. Cudre-Mauroux, M. Hauswirth, and T. van Pelt,“GridVine: Building Internet-scale semantic overlay networks,” in Proc. Int. Semantic Web Conf., pp. 107–121, 2004.

[15] M. Wylot, P. Cudre-Mauroux, and P. Groth, “TripleProv: Efficient processing of lineage queries in a native RDF store,” in Proc. 23rd Int. Conf. World Wide Web, pp. 455–466, 2014.

[16] O. Erling and I. Mikhailov, “RDF Support in the virtuoso DBMS,” in Networked Knowledge-Networked Media. New York, NY, USA: Springer, pp. 7–24, 2009.

[17] Iskandar Setiadi and Achmad Imam Kistijantoro “TippyDB: Geographically-aware distributed NoSQL Key-Value store” IEEE Data Engg 2nd International Conference, 2015.

[18] Shaolong Zhang “Application of document-oriented NoSQL database technology in web-based software project documents management system” IEEE Data Engg, ICIST, 2013.

[19] Hadoop Distributed File System, http://hadoop.apache.org/docs/r1.0.4/hdfsdesign.html, 2013.

[20] F. Guterl, “Design case history: Apple's Macintosh: A small team of little-known designers, challenged to produce a low-cost, exceptionally easy-to-use personal computer, turns out a technical milestone” IEEE Data Engg, Vol:21, Issue: 12,1984

[21] Olivier Curé and Guillaume Blin, "RDF Database Systems: Triples Storage and SPARQL Query Processing", 2015.

[22] Chunming Hu, Xixu Wang, Renyu Yang and Tianyu Wo, “ScalaRDF: A Distributed, Elastic and Scalable In-Memory RDF Triple Store” IEEE 22nd International Conference, 2016.

[23] www.Consortium,"RDFPrimer",http://www.w3.org/TR/rdfprimer/, February 2004.

[24] Xiang Lian, Lei Chen and Zi Huang, “Keyword Search Over Probabilistic RDF Graphs” IEEE Transactions on Knowledge and Data Engineering, Vol: 27, 2015.

[25] Long Cheng and Spyros Kotoulas, “Scale-Out Processing of Large RDF Datasets”, IEEE Transactions on Big Data, Vol: 1, 2015.

[26] Natsuda Kasisopha and Pornpit Wongthongtham, “Semantic wiki-based ontology evolution”, IEEE Transactions on Knowledge and Data Engineering, 2009.

[27] Xuan Zhang and Gagan Agrawal, “Supporting high performance bioinformatics flat-file data processing using indices” IEEE International Symposium on Parallel and Distributed Processing, 2008.

[28] Mohammad Husain, James McGlothlin, Mohammad M. Masud, Latifur Khan and Bhavani M. Thuraisingham, “Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing”, IEEE Transactions on Knowledge and Data Engineering, Vol: 23, 2011.

[29] “Towards efficient SPARQL query processing on RDF data”, Chang Liu; Haofen Wang; Yong Yu; Linhao Xu Tsinghua Science and Technology, Vol: 15, 2010.

[30] Xi Chen, Huajun Chen, Ningyu Zhang and Songyang Zhang, “SparkRDF: Elastic Discreted RDF Graph Processing Engine With Distributed Memory”, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol: 1, 2015.

[31] Olivier Cure, Hubert Naacke, Tendry Randriamalala and Bernd Amann, “LiteMat: A scalable, cost-efficient inference encoding scheme for large RDF graphs”, IEEE International Conference on Big Data (Big Data), 2015.

[32] Leonidas Deligiannidis, “Exploring large readable and interactive RDF graphs”, 2nd Conference on Human System Interactions, 2009.

[33] Gianfranco E. Modoni, Marco Sacco and Walter Terkaj, “A survey of RDF store solutions”, International Conference on Engineering, Technology and Innovation (ICE), 2014