A Survey on Semantic Web and Big Data Technologies for Social Network Analysis

Sercan Kulcu
Computer Engineering Dept.
TOBB University of
Economics and Technology
Ankara, Turkey
Email: skulcu@etu.edu.tr

Erdogan Dogdu
Computer Engineering Dept.
TOBB University of
Economics and Technology
Ankara, Turkey
Email: erdogandogdu@gmail.com

A. Murat Ozbayoglu
Computer Engineering Dept.
TOBB University of
Economics and Technology
Ankara, Turkey
Email: mozbayoglu@etu.edu.tr

***Abstract*—Social Network Analysis (SNA) has become a very important and increasingly popular topic among researchers in recent years especially after emerging Semantic Web and Big Data technologies. Social networking services such as Facebook, Google+, Twitter, etc. provide large amounts of data that can be used for social network analysis by researchers. Semantic Web technology plays an important role for collecting, merging, and aggregating social network data from heterogeneous sources more easily, robustly and in an interoperable manner. Today, data scientists use several different frameworks for querying, integrating and analyzing datasets located at different sources. Meanwhile, most of the big social data is in unstructured or semi-structured format. Big data architectures allow researchers to analyze unstructured data in a time and cost-efficient way. New approaches for SNA are needed to combine Semantic Web and Big Data technologies in order to utilize and add capabilities to existing solutions. To be able to analyze large scale social networks, algorithms should have scalable designs in order to benefit from the emerging Big Data technologies. This survey focuses on recently developed systems for SNA and summarizes the state-of-the-art technologies used by them and points out to future research directions.**

***Keywords*-Social network analysis, semantic web, big bata**

I. Introduction

Social network analysis (SNA) is a multidisciplinary research area that brings social sciences and computer science together. Social networks are growing rapidly and gaining popularity, so SNA has become one of the most studied research field in recent years by researchers from different disciplines. SNA has two big challenges: it has to (1) process very large datasets in a reasonable time, (2) integrate several distinct datasets into a new larger one that is semantically consistent. These are very challenging tasks.

In order to overcome the integration and adaptation problems, data analysts have begun using semantic web, mostly Resource Description Framework[1] (RDF), to store and process social data. To be able to analyze large datasets, they also utilize Big Data technologies, mostly parallel processing frameworks and graph databases.

Social networks consist of connected entities and relationships that exist between them, usually represented as a collection of nodes and edges. A node can be described as a representation of a real world entity while an edge indicates a relation between those entities. Social Network Analysis can be considered as investigating hidden connections between entities in a network and characterize the structure of the network, mostly by using graph algorithms. SNA methods primarily originate from well known graph theory topics. In the same manner, a social network can be represented and considered as a graph. According to Lee et al. Graph Mining Algorithms and Semantic Web Technologies have common characteristics to analyze a given social network to extract hidden knowledge [1].

SNA uses the graph algorithms to characterize the structure of the network, for example, discovering strategic positions in a given network or finding specific subnetworks, etc. In general, SNA tries to come up with valuable and useful associations and knowledge that are hidden in the social data. SNA may also help to prove hypotheses such as "friends are likely to have similar interests" [2]. There have been a lot of similar studies on SNA. Ostrowski developed a method to show that the structure of a network reflects the community dynamics for identification of influence and power among members [3]. Tsourakakis et al. analyzed homophily and transitivity [4]. Mislove et al. studied computing the longest distance between the nodes [5]. Corbellini et al. proposed an architecture to develop systems allowing to run graph algorithms over large scale graphs efficiently [6].

Online social network platforms such as Facebook, Instagram, Twitter, Google+, LinkedIn, etc. allow people to share, communicate, and interact with each other by using several applications. Social data consists of the content that is generated by members and structure of the network. In general, most of the graph mining algorithms such as partitioning a graph into subgraphs, are NP-hard problems, therefore the developed systems use heuristic based algorithms and keep social data distributed for efficient processing.

Social data is growing rapidly in size, variety and complexity while generally being kept in unstructured

[1]RDF, https://www.w3.org/RDF/

format [7]. In order to overcome the problem of analyzing large amounts of data that is in unstructured format, databases are being moved from relational to non-relational architecture. Non-relational databases are more scalable, network oriented, and sparsely populated than relational databases. NoSQL databases do not require a schema and avoid join operations to scale horizontally, so they usually perform better than the traditional databases on SNA.

Analysis of a given social network is a very costly and time consuming process. To solve this performance problem of large scale networks (datasets), researchers have started using parallel processing platforms. For example, Google developed MapReduce programming model for handling large scale data and Pregel for large scale graph problems. Hadoop framework is an open source alternative to Google's solutions for MapReduce algorithm. More recently Apache Spark, another open-source project, has been developed and is based on the MapReduce algorithm, and runs faster than Hadoop MapReduce due to the efficient utilization in-memory computing approach while processing data [7]. Another powerful approach is using graph databases for performance bottlenecks. In order to be able to benefit from Big data solutions, algorithms must be adapted to support parallel execution and databases need to be distributed among several platforms.

The rest of this survey paper is structured as follows. Section II reviews Semantic Web technologies. Big Data architectures and solutions are presented in Section III. Section IV outlines some of the open research topics. Finally, Section V concludes this study.

II. Semantic Web Technologies

Semantic Web technologies try to fill the knowledge gap between humans and computers. Semantic Web basically consists of two main fundamental technologies: (1) a model definition to represent and define associations between resources, which is possible by using "Resource Description Framework" (RDF) and (2) vocabularies that are used to represent the semantics of resources, which is possible by using "RDF Schema" (RDFS[2]) and "Web Ontology Language" (OWL[3]).

As social networks are becoming larger especially on the Web, Semantic Web started to attain critical importance for providing a standard model for exchanging, merging, transformation, and querying of social network data. The key benefit of the Semantic Web is to ensure the interoperability and to make sure that data is in machine readable format so it enhances the potential of Big Data analytics. Semantic Web frameworks provide a graph model (RDF), a query language (SPARQL[4]) and a schema definition (RDFS and OWL) to represent and query social data [8].

[2]RDF Schema, https://www.w3.org/TR/rdf-schema/
[3]The Web Ontology Language, https://www.w3.org/OWL/
[4]SPARQL, https://www.w3.org/TR/rdf-sparql-query/

Semantic Web promotes the graph-based representation of social data by pushing the RDF standard. Linked Data[5] has appeared to overcome the problems such as integrating different datasets from different domains. Ontologies, the key technologies in this context, provide a more flexible and distributed approach for representing data. Linking a collection of distributed datasets forms knowledge graphs. Popular knowledge graphs are Cyc[6], Freebase[7], Wikidata[8], DBpedia[9], YAGO[10], NELL[11], Google's Knowledge Graph[12], Yahoo!'s Knowledge Graph, Microsoft's Satori, and Facebook's Social Graph (based on RDFa[13]) [9].

RDF: RDF is a flexible and graph-like data model, used for representing resources on the Web, so it has become very popular to represent semantic data. It provides a mechanism for linking two resources. An RDF statement is called as a triple, in the form of <subject, predicate, object>, where subject stands for a unique resource, object denotes either a unique resource or a literal, and predicate is used to show the relationship between subject and object. A triple can be considered as a labeled directed edge between two nodes (resources) and a set of such nodes and edges form a directed graph. An example semantic graph is presented in Figure 1 illustrating people (nodes) and their relationships (edges). Notice that each edge has a label.

Social networks are complex data structures and generally modeled as graphs. An RDF graph can be described as a finite set of RDF triples. In the graph model, data is stored on nodes [10]. Opuszko and Ruhland developed a recommendation system using Semantic Web technologies by creating RDF graphs from existing datasets. They showed that better link prediction performance is achieved by RDF graph based SNA and a semantic similarity measure they developed [11].

RDF Schema: RDF Schema, an extension to RDF, is a modelling vocabulary to describe classes and properties (or relationships between classes).

OWL: The Web Ontology Language (OWL) extends the expressiveness of RDF Schema to characterize classes and properties. It was designed to be used when the information needs to be processed by machines instead of just presenting. OWL extends RDFS by defining new language primitives and provides a much richer set of vocabularies than RDFS.

SPARQL: SPARQL is the standard pattern-matching query language for RDF graphs, and considered as the backbone of the semantic web-based applications. It can

[5]Linked Data, https://www.w3.org/standards/semanticweb/data
[6]Cyc, http://dev.cyc.com/
[7]Freebase, https://developers.google.com/freebase/
[8]Wikidata, https://www.wikidata.org
[9]DBpedia, http://wiki.dbpedia.org/
[10]YAGO, http://www.yago-knowledge.org/
[11]NELL, http://rtw.ml.cmu.edu/rtw/
[12]Knowledge Graph, https://developers.google.com/knowledge-graph/
[13]RDFa, https://www.w3.org/TR/xhtml-rdfa-primer/

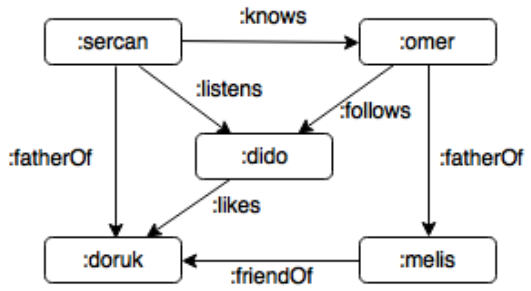

Figure 1: Visualization of an RDF Graph

be used to query conjunctions or disjunctions of graph patterns. Lee et al. showed that all SNA algorithms can be implemented using SPARQL [12]. The SPARQL can be used to query diverse RDF data sources. Property paths, the new feature of SPARQL, were studied by Kostylev et al. to query over RDF Graphs [13], Reutter et al. has added the recursion operator to SPARQL to be used in SNA applications [14].

A. TripleStores

A triplestore can be seen as a database for RDF triples. It is capable of storing and querying RDF triples in a repository by using a query language. RDF triples can be represented as graphs, therefore RDF is defined as a graph data model, SPARQL is defined as a graph query language, and triplestores are defined as a graph database system, respectively. The performance of triplestores has high importance especially for real-time applications, Saleem et al. developed a customizable framework to benchmark triplestores, namely FEASIBLE [15].

Triplestores, also known as RDF stores, usually provide an interface for querying triple data using SPARQL, also called SPARQL endpoint. Traditional triplestores run on single machine, and some of the well known examples are Jena TDB[14], RDF4J(formerly Sesame)[15], RDFSuite[16], AllegroGraph[17], RDF-3X[18], Hexastore [16], and RDFox [17]. Distributed triplestores run on several machines, they are more scalable than centralized systems, and some of the examples are Virtuoso[19], EAGRE [18], TriAD [19], 4Store[20], and TripleRush [20].

Bizer et al. compared the performance of four popular triple stores (Jena TDB, Jena SDB, Sesame and Virtuoso). The results show that Virtuoso has the best performance for big datasets, while Sesame performs better for smaller datasets [21].

[14]Jena TDB, https://jena.apache.org/documentation/tdb/
[15]RDF4J, http://rdf4j.org/
[16]RDFSuite, https://www.w3.org/2001/sw/wiki/RDFSuite
[17]AllegroGraph, http://franz.com/agraph/allegrograph/
[18]RDF-3X, https://www.openhub.net/p/rdf3x
[19]Virtuoso, http://virtuoso.openlinksw.com/
[20]4Store, https://www.w3.org/2001/sw/wiki/4store

B. RDF Vocabularies

RDF vocabularies[21] are a set of core ontologies describing certain domain or application areas in terms of semantic Web protocols (RDF, RDFS, OWL). Some of the frequently used RDF vocabularies in Social Network Analysis applications are FOAF[22] [22], SKOS[23] [23], SIOC[24], SCOT[25], MOAT[26], SemSNI [24], ActOnto [25], RELATIONSHIP[27], MUTO[28], AclOnto, and InterestOnto [26] and OntoSNA [27].

FOAF (Friend Of A Friend) ontology is used for the representation of user profiles and their friendships in social networks. FOAF can be used to merge personal information that is extracted from different sources [28]. Pankong et al. proposed a framework for semantic social network analysis by using FOAF [29].

SKOS (Simple Knowledge Organization System) presents a data model for sharing, linking, and defining knowledge about concepts and their relations. MOAT (Meaning of a Tag) provides an easy and collaborative way of defining the meaning of a tag, semantically [23].

SIOC (Semantically Interlinked Online Communities) represents the activities of communities (blogs, forums, mailing lists, wikis, etc.) on the Web. One of the extension of SIOC ontology is SIOC Types (SIOCT). Kassiri and Belouadha proposed ActOnto, which is another extension of SIOC ontology, for describing the activities of the member that exists in the network [25].

SemSNI (Semantic Social Network Interaction) ontology is designed to keep the visits made by a member or private messages among members [24]. Mahmood et al. applied social network analysis techniques on RDF graphs that is built from research papers by using SemSNA ontology [30]. Kassiri and Belouadha proposed InterestOnto for tagging and AclOnto for the access control [26].

C. Prototype systems and Frameworks

Flink system has been developed by Mika et al. to extract, aggregate and visualize social networks data. It uses semantic web technologies for reasoning on personal information [2]. Wang et al. developed an application that can query information that is extracted from Facebook user profiles [31]. LinkProbe is a prototype designed by Chen et al. to predict the existence of links among members in social networks [32]. Leida and Chu developed a system, for scalable query processing over RDF graphs that can distribute workload across the machines in the same cluster [33]. Tramp et al. proposed a distributed architecture

[21]RDF vocabularies, https://www.w3.org/standards/techs/rdfvocabs
[22]FOAF, http://xmlns.com/foaf/spec/
[23]SKOS, https://www.w3.org/2004/02/skos/
[24]SIOC, https://www.w3.org/Submission/sioc-spec/
[25]SCOT, http://rdfs.org/scot/spec/
[26]MOAT, http://lov.okfn.org/dataset/lov/vocabs/moat
[27]RELATIONSHIP, http://vocab.org/relationship/
[28]MUTO, http://muto.socialtagging.org/core/v1.html

that is built on Semantic Web technologies by combining vocabularies such as WebID, FOAF, and other protocols like Semantic Pingback and PubSubHubbub [34]. Razis et al. developed a system to discover similar twitter accounts by using Virtuoso and FOAF ontology [35]. Calvanese et al. developed an Ontology-Based Data Access (OBDA) system to answer SPARQL queries over RDF graphs [36]. Golbeck and Rothstein merged FOAF profiles from different sources [37]. Ereteo et al. developed SemTagP that detects and characterizes the communities from RDF graphs [38].

III. Big Data Architectures

All definitions about big data agree that it is huge in size, unstructured and has different data formats. It is a complex and challenging task to gather, store, query, process, and analyze large scale data. Big data is characterized by five Vs: *Volume* (data size), *Velocity* (speed of data produced and delivered), *Variety* (data is comprised of heterogeneous sources), *Value* (extracting useful and worthy information), and *Veracity* (security, privacy, and trust) [39].

Big data technologies are used on very large datasets, originating from different domains. Extracting knowledge from these datasets is more difficult than traditional datasets [40].

Social media sites are valuable social data providers. Some of the frequently used social media sites and the technologies that are used by them are tabulated in Table I.

Table I: Social network applications and the technologies they use

Social Network Service	**Used Big Data Technologies**
Amazon	DynamoDB, SimpleDB
eBay	MongoDB, Neo4J
Facebook	Cassandra, HBase, Giraph, Presto, Neo4j
Flickr	MongoDB, Neo4j
Foursquare	Cassandra, CouchDB, InfoGrid, MongoDB, Riak
Google+	BigTable
Instagram	Cassandra
Linkedln	AllegroGraph, HBase, MongoDB, Neo4J
Twitter	Cassandra, FlockDB, HBase, Neo4j, Pig
Yahoo!	Hbase
Yandex	MongoDB
Youtube	BigTable

The literature shows that big data architectures on social network analysis are grouped into two types: parallel processing frameworks for graph processing and NoSQL database related technologies to store/access RDF data.

A. Parallel processing frameworks

Graph processing frameworks provide an environment to analyze social networks by allowing developers to develop scalable and fault-tolerant applications in a distributed manner. The challenges on graph processing are constructing networks by using unstructured data, analyzing the structure of the network, and inferring hidden knowledge. Graph analytics have proven to be valuable tools in solving these challenges [41]. Some of the key technologies are MapReduce[29] programming paradigm, Hadoop[30] framework, Spark[31], and Flink [2].

The MapReduce, developed by Google, is a data-parallel processing framework. While MapReduce is used for massive data analysis, there are still debates on its performance and efficiency [42]. In order to achieve the best MapReduce performance, whole graph must be available in the memory while the algorithm is running. MapReduce solves this problem by storing data on distributed machines and processing them in parallel.

Apache Hadoop is an open source implementation of MapReduce algorithm. It has a collection of frameworks and utilities that are designed for storing and processing big data. Hadoop is designed to be fault tolerant and handle failures at the application level by hiding details from users. One machine, among other machines on a cluster, controls and organizes the processes with scheduling and load-balancing features. Hadoop Distributed File System (HDFS[32]), part of Hadoop, is a distributed file system that is responsible for storing big data over a cluster of commodity machines in data blocks with multiple copies for fault-tolerance. Although MapReduce can provide good scalability for batch processing, it is not efficient for interactive jobs or real-time queries. Gupta et al. shared the story of Twitter's "Who to Follow" user recommendation product. The architecture is completely based on Hadoop specifically, all algorithms are implemented in Pig [43]. Goasdoue et al. presented CliqueSquare, an efficient RDF data management platform built on top of Hadoop for storing and processing large amounts of RDF data [44]. Curtiss et al. developed Unicorn [45] and Busch et al. developed EarlyBird [46] for searching the social graph. Junghanss et al. developed GRADOOP to analyze graph data for social network analysis [47].

Apache Spark is a parallel computation platform for large scale data processing. It has in-memory storage abstraction called Resilient Distributed Datasets (RDDs) that allows applications to keep and process data in memory by minimizing disk access [48]. It can run applications up to 100x times faster than Hadoop MapReduce [49]. Furthermore, unlike Hadoop MapReduce, developers can perform iterative computations very easily by using Spark [50]. Cure et al. demonstrated that Spark can process distributed RDF queries more efficiently than Hadoop [51] and implemented an RDF store, namely HAQWA [52].

Marcu et al. evaluated performances of Spark and Flink systems. Results show that Spark performs better for large

[29]Google MapReduce, http://research.google.com/archive/mapreduce.html
[30]Apache Hadoop, http://hadoop.apache.org/
[31]Apache Spark, http://spark.apache.org/
[32]HDFS, http://wiki.apache.org/hadoop/HDFS

scale graph processing while Flink performs better for batch processing [53].

Scripting Languages: High level languages have been developed for the big data platforms to allow developers to develop applications more easily. Some of the the well known languages are: Pig Latin for Pig[33], HiveQL for Hive[34], and LINQ for Dryad [54].

Pig, developed at Yahoo, allows developers to execute data flows in parallel on Hadoop platform. Pig includes a language to process data, it is called Pig Latin, which includes special operators for data operations (join, sort, filter, etc.) [55].

Facebook and Netflix use Hive and support HiveQL language [56]. Hive converts each SQL subquery to a MapReduce job and merges subqueries that share the same keys into one job to reduce the total number of jobs. RDF triples can be stored and accessed on HDFS by designing an extra relational layer as Mammo et al. used Hive in their system [57]. Liu et al. designed an algorithm to translate SPARQL queries to MapReduce jobs and showed that it runs 2 times faster than traditional methods [58]. Mammo showed that Presto has better performance than Hive, and it can be used to process large amounts of RDF data. Query processing in distributed and in-memory engines takes less time to respond on big RDF datasets than query processing engines that rely on MapReduce [48]. Presto is more scalable than Hive according to tests [57].

Graph Processing Systems: There are many graph processing platforms. Some of them are Pegasus[35], Pregel [59], Giraph[36], GraphX[37], GraphLab[38], GPS[39] , Hama [60], Trinity [61], Twister [62], Haloop [63], and SIGNAL/COLLECT[40].

Pegasus is a Hadoop-based graph mining system, and runs algorithms in parallel by converting them to a series of MapReduce jobs [64]. Pregel is a distributed programming framework, similar to MapReduce, but has much more efficient support for iterative computations over graphs [59]. Pregel has in-memory and vertex-centric approach to avoid communication overheads [65]. Apache Giraph is an open source implementation of Pregel, based on Hadoop framework and has been developed by Facebook. Giraph is built for high scalability, and uses HDFS to store data [66].

GraphX extends Spark RDD abstraction, which is called Resilient Distributed Graph (RDG). Xin et al. showed that GraphX is 8 times faster than a general Hadoop MapReduce application [48]. GraphLab is based on the shared memory abstraction. Unlike Pregel, it is not a vertex-centric model, since each vertex can access the data of neighbor nodes [67]. Urika provides both software and hardware based solutions to process SPARQL queries over RDF triples.

Guo et al. have identified that comparison of graph processing platforms according to their performance is a very complicated task. Dataset used, characteristics of the algorithm, and platform diversity can undesirably affect the results in the wrong way [68].

Lim et al. compared Pegasus, GraphX, and Urika graph analysis platforms and showed that GraphX is the best in iterative graph operations like finding connected components, and Urika outperforms the others in non-iterative operations like computing degree distribution [69]. MapReduce is 2 times slower than Giraph and GraphX, but it can run without keeping the whole graph data in memory [70]. Batarfi et al. evaluated the performances of five graph processing systems (GraphChi, GraphX, Giraph, GPS, and GraphLab) and showed that the standalone system, GraphChi, is slower than the other distributed large scale systems. GraphX is shown to be 2 times faster than GraphChi, Giraph, and GraphLab [65]. Guo et al. compared Hadoop, Stratosphere [71], Giraph, GraphLab, YARN, and Neo4j platforms and the results show that Hadoop is the worst platform for running multi-iteration algorithms [68].

B. NoSQL Database related technologies

A database model is a representation of entities and the relationships between them. NoSQL Database related technologies are separated into two categories as Tabular databases and Graph databases.

Tabular Databases: Some popular examples of tabular databases are Hbase[41], Cassandra[42], Accumulo[43], and MongoDB[44].

HBase is an open source, non-relational and distributed database model like Google's BigTable. Using Apache HBase is advantageous when data is needed to be accessed in real-time and randomly. eBay, Yahoo, Facebook etc. use Hbase [56]. Barnawi et al. compared the performance of Giraph by using HDFS, Hive and HBase storage systems. The results show that HDFS and Hive perform better than HBase [72].

Cassandra is a distributed database system for managing very large amounts of structured data. Cassandra has been developed by Facebook initially [73]. CumulusRDF[45], implemented on Cassandra, is competitive among other distributed RDF stores [74].

[33]Pig, https://pig.apache.org/
[34]Hive, https://hive.apache.org/
[35]Pegasus, http://www.cs.cmu.edu/~pegasus/
[36]Giraph, http://giraph.apache.org/
[37]Spark Graphx, http://spark.apache.org/graphx/
[38]GraphLab, http://select.cs.cmu.edu/code/graphlab/
[39]GPS, http://infolab.stanford.edu/gps/
[40]SIGNAL/COLLECT, http://uzh.github.io/signal-collect/

[41]Apache Hbase, https://hbase.apache.org/
[42]Apache Cassandra, http://cassandra.apache.org/
[43]Accumulo, https://accumulo.apache.org/
[44]MongoDB, https://www.mongodb.com/
[45]CumulusRDF, https://www.w3.org/2001/sw/wiki/CumulusRDF

Graph Databases: Graph databases are generally the best choice for managing complex and densely connected data. Tabular databases are more complex and time consuming than graph databases while analyzing social networks [10]. Query time is independent from the size of the graph [75] and data is not stored in tables, each node is directly connected to other nodes [76] on graph databases. Well known graph databases are RDF TripleStore, Neo4J[46], DEX [77], InfiniteGraph[47], HyperGraphDB[48], FlockDB[49], OrientDB[50], and Titan[51].

Neo4j is a graph database, especially optimized for graph structure. It can scale up to billions of nodes and edges [76]. DEX is a high performance graph database and allows edges to be directed or undirected [77]. Titan[51] is a highly scalable graph database, provides batch processing and runs on Hadoop framework. HyperGraphDB is a distributed graph database that supports hypergraphs. Hypergraph does not contain standard graph properties due to the fact that one edge can point to multiple edges [78]. Infinitegraph[47] is a distributed graph database, based on a graph structure. It is especially designed for high throughput.

Mathew and Kumar compared the five mostly used NoSQL graph databases on SNA (Neo4J, FlockDB, OrientDB, InfoGrid, and AllegroGraph) and showed that Neo4J is better than the others [79].

Query Languages: Graph query languages play an important role on social network analysis mostly due to performance issues. Some of the developed query languages are Cypher[52], Gremlin[53], GraphQL [80], GraphLog, QGraph, and SNQL [81].

IV. Future Research Directions

Social networks keep large amounts of social data that contains personal and private data about members of the network. Therefore, data must be handled and preserved properly. Ensuring data privacy is still a challenging and open problem [82]. Privacy and trust are major concerns for services and applications on social networks [83].

Social network services like Google+, Facebook, Twitter, etc. do not allow a researcher to get the whole graph to analyze, so algorithms should be designed with the consideration that they work correctly with the limited data provided [84].

Matching two user profiles across multiple social networks is an important problem on SNA, and there are still open issues in this area [85].

[46]Neo4J, https://neo4j.com/

[47]InfinityGraph, http://www.objectivity.com/products/infinitegraph/

[48]HyperGraphDB, http://www.hypergraphdb.org/

[49]FlockDB, https://github.com/twitter/flockdb

[50]OrientDB, http://orientdb.com/orientdb/

[51]Titan, http://titan.thinkaurelius.com/

[52]Cypher, http://neo4j.com/docs/developer-manual/current/cypher/

[53]Gremlin, https://tinkerpop.apache.org/gremlin.html

Efficiently processing large scale RDF datasets is another open research area [86] [87]. Distributed RDF stores have been developed and are still being researched.

The size of the semantic RDF data is increasing rapidly, so it is infeasible to store, query and process them on a single machine. Researches therefore have focused on developing distributed datastores for efficient processing, especially using Hadoop-based distributed storage and processing frameworks. But, Hadoop is not designed for RDF processing or iterative graph processing. In Hadoop, all data is read in batch mode from HDFS and written to HDFS after each iteration, and immediate results should be written to HDFS back to be able to use them on following iterations. Most of the parallelized graph algorithms run iteratively, requiring a large number of map and reduce jobs [88]. Spark is a strong alternative to Hadoop, because of having low disk I/O cost and better performance. Therefore, many systems developed before are being ported to Spark-based frameworks. For example, Facebook has begun to port their system from Hive to Spark[54]. Some of the hybrid systems developed are Spark-based (SPARQLGX [89], S2RDF [90], SeBiDa [91]), Accumula-based (RYA [92]), Pig-based (PigSPARQL [93], SPARQling [94]), Hbase-based (RDFChain [95], H2RDF+ [96]), Hadoop-based (SHARD [97], nHopDB [98], SHAPE [99], HadoopRDF [100], RAPID+ [101], EAGRE [18], HadoopSPARQL [102], CliqueSquare [44]), SimpleDB-based (Stratustore [103]), Cassandra-based (CumulusRDF [74]), and Trinity-based (Trinity.RDF [87]).

V. Conclusion

Analyzing relations between individuals in social networks has become an interesting and challenging research area due to the emerging growth of social networks. To be able to analyze the interactions and characterize the structure of the network such as computing degree centrality, or clustering coefficient, etc., it is required to use big data tools and frameworks.

Large scale social network data analysis is a challenging task. Big data is not related with only the size of the data, but also about its diversity and heterogeneity in data formats and sources, and decentralized control on distributed data [104]. The performance of big data architectures depends on the properties of tasks. For example, Hadoop-based systems are suitable for batch processing and generally used for very large scale datasets while Spark-based systems are suitable for interactive jobs and real-time queries.

Ontologies are extensively used for data integration due to its ability to support schema-less and heterogeneous data. Social data can be represented as RDF triples by using existing ontologies. RDF graph is a collection of RDF triples

[54]https://code.facebook.com/posts/1671373793181703/apache-spark-scale-a-60-tb-production-use-case/

and can be queried by using SPARQL. To benefit from big data technologies, queries should be translated to jobs that can run in parallel. Decentralized RDF triplestores are the key technologies to handle large scale RDF data efficiently.

REFERENCES

[1] S. Lee, S. R. Sukumar, and S.-H. Lim, "Graph mining meets the semantic web," in *Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on*. IEEE, 2015, pp. 53–58.

[2] P. Mika, "Flink: Semantic web technology for the extraction and analysis of social networks," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, no. 2, pp. 211–223, 2005.

[3] D. A. Ostrowski, "Semantic social network analysis for trend identification," in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 178–185.

[4] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos, "Doulion: counting triangles in massive graphs with a coin," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 837–846.

[5] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 29–42.

[6] A. Corbellini, C. Mateos, D. Godoy, A. Zunino, and S. Schiaffino, "An architecture and platform for developing distributed recommendation algorithms on large-scale social networks," *Journal of Information Science*, vol. 41, no. 5, pp. 686–704, 2015.

[7] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: Recent achievements and new challenges," *Information Fusion*, vol. 28, pp. 45–59, 2016.

[8] M. San Martín and C. Gutierrez, "Representing, querying and transforming social networks with rdf/sparql," in *European Semantic Web Conference*. Springer, 2009, pp. 293–307.

[9] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, no. Preprint, pp. 1–20, 2016.

[10] R. kumar Kaliyar, "Graph databases: A survey," in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. IEEE, 2015, pp. 785–790.

[11] M. Opuszko and J. Ruhland, "Classification analysis in complex online social networks using semantic web technologies," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, 2012, pp. 1032–1039.

[12] S. Lee, S. R. Sukumar, S. Hong, and S.-H. Lim, "Enabling graph mining in rdf triplestores using sparql for holistic in-situ graph analysis," *Expert Systems with Applications*, vol. 48, pp. 9–25, 2016.

[13] E. V. Kostylev, J. L. Reutter, M. Romero, and D. Vrgoč, "Sparql with property paths," in *International Semantic Web Conference*. Springer, 2015, pp. 3–18.

[14] J. L. Reutter, A. Soto, and D. Vrgoč, "Recursion in sparql," in *International Semantic Web Conference*. Springer, 2015, pp. 19–35.

[15] M. Saleem, Q. Mehmood, and A.-C. N. Ngomo, "Feasible: A feature-based sparql benchmark generation framework," in *International Semantic Web Conference*. Springer, 2015, pp. 52–69.

[16] C. Weiss, P. Karras, and A. Bernstein, "Hexastore: sextuple indexing for semantic web data management," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1008–1019, 2008.

[17] Y. Nenov, R. Piro, B. Motik, I. Horrocks, Z. Wu, and J. Banerjee, "Rdfox: A highly-scalable rdf store," in *International Semantic Web Conference*. Springer, 2015, pp. 3–20.

[18] X. Zhang, L. Chen, Y. Tong, and M. Wang, "Eagre: Towards scalable i/o efficient sparql query evaluation on the cloud," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 565–576.

[19] S. Gurajada, S. Seufert, I. Miliaraki, and M. Theobald, "Triad: a distributed shared-nothing rdf engine based on asynchronous message passing," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 289–300.

[20] P. Stutz, M. Verman, L. Fischer, and A. Bernstein, "Triplerush: A fast and scalable triple store," in *Proceedings of the 9th International Conference on Scalable Semantic Web Knowledge Base Systems-Volume 1046*. CEUR-WS. org, 2013, pp. 50–65.

[21] C. Bizer and A. Schultz, "The berlin sparql benchmark," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 2, pp. 1–24, 2009.

[22] D. Brickley and L. Miller, "Foaf vocabulary specification 0.98," *Namespace document*, vol. 9, 2012.

[23] T. Baker, S. Bechhofer, A. Isaac, A. Miles, G. Schreiber, and E. Summers, "Key choices in the design of simple knowledge organization system (skos)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 20, pp. 35–49, 2013.

[24] G. Erétéo, M. Buffa, F. Gandon, and O. Corby, "Analysis of a real online social network using semantic web frameworks," in *International Semantic Web Conference*. Springer, 2009, pp. 180–195.

[25] A. El Kassiri and F. Z. Belouadha, "Actonto: An extension of the sioc standard for social media analysis and interoperability," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*. IEEE, 2014, pp. 62–67.

[26] A. El Kassiri and F.-Z. Belouadha, "Towards a unified semantic model for online social networks analysis and interoperability," in *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*. IEEE, 2015, pp. 1–6.

[27] G. Erétéo, M. Buffa, F. Gandon, P. Grohan, M. Leitzelman, and P. Sander, "A state of the art on social network analysis and its applications on a semantic web," in *SDOW2008*, 2008.

[28] P. Mika, "Social networks and the semantic web," in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 2004, pp. 285–291.

[29] N. Pankong, S. Prakancharoen, and M. Buranarach, "A combined semantic social network analysis framework to integrate social media data," in *Knowledge and Smart Technology (KST), 2012 4th International Conference on*. IEEE, 2012, pp. 37–42.

[30] Q. Mahmood, M. A. Qadir, and M. T. Afzal, "Document similarity detection using semantic social network analysis on rdf citation graph," in *Emerging Technologies (ICET), 2013 IEEE 9th International Conference on*. IEEE, 2013, pp. 1–6.

[31] R.-C. Wang, T.-H. Su, C.-P. Ma, S.-H. Chen, and H.-H. Huang, "Social network data retrieving using semantic technology," in *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*. IEEE, 2013, pp. 322–327.

[32] H. Chen, W.-S. Ku, H. Wang, L. Tang, and M.-T. Sun, "Linkprobe: Probabilistic inference on large-scale social networks," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 290–301.

[33] M. Leida and A. Chu, "Distributed sparql query answering over rdf data streams," in *2013 IEEE International Congress on Big Data*. IEEE, 2013, pp. 369–378.

[34] S. Tramp, P. Frischmuth, T. Ermilov, S. Shekarpour, and S. Auer, "An architecture of a distributed semantic social network," *Semantic Web*, vol. 5, no. 1, pp. 77–95, 2014.

[35] G. Razis and I. Anagnostopoulos, "Discovering similar twitter accounts using semantics," *Engineering Applications of Artificial Intelligence*, vol. 51, pp. 37–49, 2016.

[36] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao, "Ontop: Answering sparql queries over relational databases," *Semantic Web*, no. Preprint, pp. 1–17, 2016.

[37] J. Golbeck and M. Rothstein, "Linking social networks on the web with foaf: A semantic web case study," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'08. AAAI Press, 2008, pp. 1138–1143. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620163.1620249

[38] G. Ereteo, F. Gandon, and M. Buffa, "Semtagp: Semantic community detection in folksonomies," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, vol. 1, Aug 2011, pp. 324–331.

[39] T. Das and P. M. Kumar, "Big data analytics: A framework for unstructured data analysis," *International Journal of Engineering Science & Technology*, vol. 5, no. 1, p. 153, 2013.

[40] W. Tan, M. B. Blake, I. Saleh, and S. Dustdar, "Social-network-sourced big data analytics." *IEEE Internet Computing*, vol. 17, no. 5, pp. 62–69, 2013.

[41] W. M. Campbell, C. K. Dagli, and C. J. Weinstein, "Social network analysis with content and graphs," *Lincoln Laboratory Journal*, vol. 20, no. 1, pp. 61–81, 2013.

[42] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, "Parallel data processing with mapreduce: a survey," *AcM sIGMoD Record*, vol. 40, no. 4, pp. 11–20, 2012.

[43] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh, "Wtf: The who to follow service at twitter," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 505–514.

[44] F. Goasdoué, Z. Kaoudi, I. Manolescu, J. Quiané-Ruiz, and S. Zampetakis, "Cliquesquare: efficient hadoop-based rdf query processing," in *BDA'13-Journées de Bases de Données Avancées*, 2013.

[45] M. Curtiss, I. Becker, T. Bosman, S. Doroshenko, L. Grijincu, T. Jackson, S. Kunnatur, S. Lassen, P. Pronin, S. Sankar *et al.*, "Unicorn: A system for searching the social graph," *Proceedings of the VLDB Endowment*, vol. 6, no. 11, pp. 1150–1161, 2013.

[46] M. Busch, K. Gade, B. Larson, P. Lok, S. Luckenbill, and J. Lin, "Earlybird: Real-time search at twitter," in *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 2012, pp. 1360–1369.

[47] M. Junghanns, A. Petermann, K. Gómez, and E. Rahm, "Gradoop: Scalable graph data management and analytics with hadoop," *arXiv preprint arXiv:1506.00548*, 2015.

[48] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, "Graphx: A resilient distributed graph system on spark," in *First International Workshop on Graph Data Management Experiences and Systems*. ACM, 2013, p. 2.

[49] M. Mammo and S. K. Bansal, "Presto-rdf: Sparql querying over big rdf data," in *Australasian Database Conference*. Springer, 2015, pp. 281–293.

[50] R. Elshawi, O. Batarfi, A. Fayoumi, A. Barnawi, and S. Sakr, "Big graph processing systems: State-of-the-art and open challenges," in *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on*. IEEE, 2015, pp. 24–33.

[51] O. Curé, H. Naacke, M.-A. Baazizi, and B. Amann, "On the evaluation of rdf distribution algorithms implemented over apache spark," *arXiv preprint arXiv:1507.02321*, 2015.

[52] O. Cure, H. Naacke, M.-A. Baazizi, and B. Amann, "Haqwa: a hash-based and query workload aware distributed rdf store," *Poster Session ISWC*, 2015.

[53] O.-C. Marcu, A. Costan, G. Antoniu, and M. S. Pérez, "Spark versus flink: Understanding performance in big data analytics frameworks," in *Cluster 2016-The IEEE 2016 International Conference on Cluster Computing*, 2016.

[54] Y. Yu, M. Isard, D. Fetterly, M. Budiu, Ú. Erlingsson, P. K. Gunda, and J. Currey, "Dryadlinq: A system for general-purpose distributed data-parallel computing using a high-level language." in *OSDI*, vol. 8, 2008, pp. 1–14.

[55] V. R. Konasani and S. Kadre, "Introducing big data analytics," in *Practical Business Analytics Using SAS*. Springer, 2015, pp. 509–540.

[56] A. Bhardwaj, A. Kumar, Y. Narayan, P. Kumar *et al.*, "Big data emerging technologies: A casestudy with analyzing twitter data using apache hive," in *2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*. IEEE, 2015, pp. 1–6.

[57] M. Mammo and S. K. Bansal, "Distributed sparql over big rdf data: A comparative analysis using presto and mapreduce," in *2015 IEEE International Congress on Big Data*. IEEE, 2015, pp. 33–40.

[58] L. Liu, J. Yin, and L. Gao, "Efficient social network data query processing on mapreduce," in *Proceedings of the 5th ACM workshop on HotPlanet*. ACM, 2013, pp. 27–32.

[59] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 135–146.

[60] S. Seo, E. J. Yoon, J. Kim, S. Jin, J.-S. Kim, and S. Maeng, "Hama: An efficient matrix computation with the mapreduce framework," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 721–726.

[61] B. Shao, H. Wang, and Y. Li, "Trinity: A distributed graph engine on a memory cloud," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 2013, pp. 505–516.

[62] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: a runtime for iterative mapreduce," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. ACM, 2010, pp. 810–818.

[63] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "Haloop: efficient iterative data processing on large clusters," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 285–296, 2010.

[64] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. F. da Silva, M. Livny *et al.*, "Pegasus, a workflow management system for science automation," *Future Generation Computer Systems*, vol. 46, pp. 17–35, 2015.

[65] O. Batarfi, R. El Shawi, A. G. Fayoumi, R. Nouri, A. Barnawi, S. Sakr *et al.*, "Large scale graph processing systems: survey and an experimental evaluation," *Cluster Computing*, vol. 18, no. 3, pp. 1189–1213, 2015.

[66] M. Han, K. Daudjee, K. Ammar, M. T. Özsu, X. Wang, and T. Jin, "An experimental comparison of pregel-like graph processing systems," *Proceedings of the VLDB Endowment*, vol. 7, no. 12, pp. 1047–1058, 2014.

[67] Y. Low, J. E. Gonzalez, A. Kyrola, D. Bickson, C. E. Guestrin, and J. Hellerstein, "Graphlab: A new framework for parallel machine learning," *arXiv preprint arXiv:1408.2041*, 2014.

[68] Y. Guo, M. Biczak, A. L. Varbanescu, A. Iosup, C. Martella, and T. L. Willke, "How well do graph-processing platforms performs an empirical performance evaluation and analysis," in *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*. IEEE, 2014, pp. 395–404.

[69] S.-H. Lim, S. Lee, G. Ganesh, T. C. Brown, and S. R. Sukumar, "Graph processing platforms at scale: Practices and experiences," in *Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 42–51.

[70] M. Capotă, T. Hegeman, A. Iosup, A. Prat-Pérez, O. Erling, and P. Boncz, "Graphalytics: A big data benchmark for graph-processing platforms," in *Proceedings of the GRADES'15*. ACM, 2015, p. 7.

[71] A. Alexandrov, R. Bergmann, S. Ewen, J.-C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl *et al.*, "The stratosphere platform for big data analytics," *The VLDB Journal*, vol. 23, no. 6, pp. 939–964, 2014.

[72] A. Barnawi, O. Batarfi, R. Elshawi, A. Fayoumi, R. Nouri, S. Sakr *et al.*, "On characterizing the performance of distributed graph computation platforms," in *Technology Conference on Performance Evaluation and Benchmarking*. Springer, 2014, pp. 29–43.

[73] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

[74] G. Ladwig and A. Harth, "Cumulusrdf: linked data management on nested key-value stores," in *The 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2011)*, 2011, p. 30.

[75] M. A. Rodriguez and P. Neubauer, "The graph traversal pattern," *arXiv preprint arXiv:1004.1001*, 2010.

[76] Ł. Warchał, "Using neo4j graph database in social network analysis," *Studia Informatica*, vol. 33, no. 2A, pp. 271–279, 2012.

[77] N. Martinez-Bazan, S. Gomez-Villamor, and F. Escale-Claveras, "Dex: A high-performance graph database management system," in *Data Engineering Workshops (ICDEW), 2011 IEEE 27th International Conference on*. IEEE, 2011, pp. 124–127.

[78] B. Iordanov, "Hypergraphdb: a generalized graph database," in *International Conference on Web-Age Information Management*. Springer, 2010, pp. 25–36.

[79] A. B. Mathew and S. M. Kumar, "Analysis of data management and query handling in social networks using nosql databases," in *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*. IEEE, 2015, pp. 800–806.

[80] H. He and A. K. Singh, "Graphs-at-a-time: Query language and access methods for graph databases," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 405–418. [Online]. Available: http://doi.acm.org/10.1145/1376616.1376660

[81] M. San Martın, C. Gutierrez, and P. T. Wood, "Snql: A social networks query and transformation language," *cities*, vol. 5, p. r5, 2011.

[82] S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. A. Beyah, "On your social network de-anonymizablity: Quantification and large scale evaluation with seed knowledge." in *NDSS*, 2015.

[83] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 47, 2013.

[84] N. Vesdapunt and H. Garcia-Molina, "Identifying users in social networks with limited information," in *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 2015, pp. 627–638.

[85] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. P. Gummadi, "On the reliability of profile matching across large online social networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1799–1808.

[86] A. Potter, B. Motik, and I. Horrocks, "Querying distributed rdf graphs: The effects of partitioning," in *10th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2014)*. Citeseer, 2014, p. 29.

[87] K. Zeng, J. Yang, H. Wang, B. Shao, and Z. Wang, "A distributed graph engine for web scale rdf data," in *Proceedings of the VLDB Endowment*, vol. 6, no. 4. VLDB Endowment, 2013, pp. 265–276.

[88] J. Magnusson and T. Kvernvik, "Subscriber classification within telecom networks utilizing big data technologies and machine learning," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. ACM, 2012, pp. 77–84.

[89] D. Graux, L. Jachiet, P. Geneves, and N. Layaïda, "Sparqlgx: Efficient distributed evaluation of sparql with apache spark," in *The 15th International Semantic Web Conference*, 2016.

[90] A. Schätzle, M. Przyjaciel-Zablocki, S. Skilevic, and G. Lausen, "S2rdf: Rdf querying with sparql on spark," *arXiv preprint arXiv:1512.07021*, 2015.

[91] M. N. Mami, S. Scerri, S. Auer, and M.-E. Vidal, "Towards semantification of big data technology," in *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 2016, pp. 376–390.

[92] R. Punnoose, A. Crainiceanu, and D. Rapp, "Rya: a scalable rdf triple store for the clouds," in *Proceedings of the 1st International Workshop on Cloud Intelligence*. ACM, 2012, p. 4.

[93] A. Schätzle, M. Przyjaciel-Zablocki, and G. Lausen, "Pigsparql: Mapping sparql to pig latin," in *Proceedings of the International Workshop on Semantic Web Information Management*. ACM, 2011, p. 4.

[94] S. Hagedorn, K. Hose, and K.-U. Sattler, "Sparqling pig-processing linked data with pig latin." in *BTW*, 2015, pp. 279–298.

[95] P. Choi, J. Jung, and K.-H. Lee, "Rdfchain: chain centric storage for scalable join processing of rdf graphs using mapreduce and hbase," in *Proceedings of the 2013th International Conference on Posters & Demonstrations Track-Volume 1035*. CEUR-WS. org, 2013, pp. 249–252.

[96] N. Papailiou, D. Tsoumakos, I. Konstantinou, P. Karras, and N. Koziris, "H 2 rdf+: an efficient data management system for big rdf graphs," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 909–912.

[97] K. Rohloff and R. E. Schantz, "High-performance, massively scalable distributed systems using the mapreduce software framework: the shard triple-store," in *Programming Support Innovations for Emerging Distributed Applications*. ACM, 2010, p. 4.

[98] J. Huang, D. J. Abadi, and K. Ren, "Scalable sparql querying of large rdf graphs," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1123–1134, 2011.

[99] K. Lee and L. Liu, "Scaling queries over big rdf graphs with semantic hash partitioning," *Proceedings of the VLDB Endowment*, vol. 6, no. 14, pp. 1894–1905, 2013.

[100] J.-H. Du, H.-F. Wang, Y. Ni, and Y. Yu, "Hadooprdf: A scalable semantic data analytical engine," in *International Conference on Intelligent Computing*. Springer, 2012, pp. 633–641.

[101] P. Ravindra, H. Kim, and K. Anyanwu, "An intermediate algebra for optimizing rdf graph pattern matching on mapreduce," in *Extended Semantic Web Conference*. Springer, 2011, pp. 46–61.

[102] C. Liu, J. Qu, G. Qi, H. Wang, and Y. Yu, "Hadoopsparql: a hadoop-based engine for multiple sparql query answering," in *Extended Semantic Web Conference*. Springer, 2012, pp. 474–479.

[103] R. Stein and V. Zacharias, "Rdf on cloud number nine," in *4th Workshop on New Forms of Reasoning for the Semantic Web: Scalable and Dynamic*, 2010, pp. 11–23.

[104] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.