Top-K Entity Units Retrieval Over Big Data

Da Zhang
University Of Miami
Coral Gables,Florida,33146
zhang.1855@miami.edu

Mansur R. Kabuka
University Of Miami
Coral Gables,Florida,33146
m.kabuka@miami.edu

ABSTRACT

During the past several years, data size has increased explosively. This data explosion tendency has impacted various fields ranging from biomedical engineering, business consulting to social media and mobile application. Big Data is a two sided sword. While it provides incredibly treasured insights in commercial scope and innovative discovery in the scientific field, Big Data also has many challenges, such as complication in data storage, data processing, data analysis and data visualization. Among all these challenges, keyword searching over a large volume of data prevails as one of the four tasks defined by Bizer et al. at the year of 2012. Keyword searching refers to retrieving the objects relevant to the entities of concern using scientific computational methods. Consequently, efficiently solving the problem of keyword searching can contribute as a foundation to diverse Big Data applications.

Keywords

Big Data; Information Retrieval; Keyword Searching

1. INTRODUCTION

The volume of data size has increased significantly in recent years due to large amounts of data generated by various organizations, social networks, devices and applications. DBLP[1] has over 1,200,000 objects and more than 2,480,000 links, and the YAGO [2] data set includes 10 million triples about the facts and entities. Data.gov [3], which maintains the largest open-government and machine-readable data, has more than a 194,708 datasets up to date, having impact from improved civic services, informed policy to research and scientific disciplines. Consequently, the ever growing data size exceeds any single computer's capability to manipulate and analyze. Additionally, current trend shows that the solution to the Big Data challenge is gradually changing from accelerating hardware capability to establishing software solutions infrastructure [1]. Along with other dilemmas, keyword searching task emerges as the essential basis for numerous applications: recommender system, Big Scholar citation system, etc. According to the definition of keyword search [2] "The objective of keyword research is to generate, with good precision and recall, large number of terms that are highly relevant yet non-obvious to the given input keyword." Therefore, accomplishing this keyword searching task can be a challenging and time-consuming target that requires enormous computing framework to guarantee efficient and reliable data storage, processing and analysis. Additionally, keyword searching over a large volume of data remains as one of the four tasks defined by Bizer et al.[3]. at the year of 2012.

[1]http://dblp.uni-trier.de/
[2]https://datahub.io/dataset/yago
[3]https://www.data.gov/

WWW'17 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3053063

2. RELATED WORK

Traditional keyword searching approaches have always utilized the inverted index method[4][5][6] to process keyword queries, which is effective for unstructured data but not for structured data[7] such as data in *.ttl* or *.rdf* formats[8]. Due to the nature of the internal relation between objects in the web, next generation web search engines will require link information, or more commonly, the capability of integrating correlative entities that connected through associations[9]. Endeavor has already been invested in this research area[8][10][11][12][13]. However, most of the keyword searching methods require users be familiar with SPARQL[14] query language, which is a specific language with capabilities for querying graph patterns. The results returned by SPARQL queries can be result list sets or subgraphs. Leal et al. proposes a kSP method of searching Top-K relevant semantic places given a place keywords set, which integrates keyword search with location-based retrieval. Different from SPARQL-based keyword searching methodology,this method [8] does not require proficiency in SPARQL query language, which makes the method easy to use for common users. As stated in[8], given a set of keywords, by measuring the graph distance between the place and the occurrence of the covered keywords at the nodes of the tree data structure, the algorithm[8] returns the Top-K places with the most relevant places to the query location based on the aggregate ranking function. However, this method is limited by scalability since it can not be deployed distributively. Thus, it is necessary to design a system integrating software tools sup-

ported by hardware back-end to provide a reliable, efficient and distributed solution to keyword searching problem.

3. DEFINITION

Definition 1. Keywords Searching problem: Given a set of keywords $V = \{v_1, v_2 \dots v_m\}$, identify and extract the subgraphs that cover all or the maximum portion of the keywords from original large information graph.

In other words, different from a traditional method which returns a list of separated entities, a set of entities which we identify as Entity Unit associated with the internal relationship will be returned to the user.

Definition 2. Searching Match: Given a keyword set $V = \{v_1, v_2 \dots v_m\}$ consisting of m distinct keywords, a searching match is satisfied if an Entity Unit represented by a subgraph $G' =< V', E' >$ where $V' = \{v'_1, v'_2 \dots v'_n\}$ and $E' = \{e_1, e_2 \dots e_n\}$ satisfies all the following conditions:

1. $V' = \{v'_1, v'_2 \dots v'_n\}$ covers all the keywords $V = \{v_1, v_2 \dots v_m\}$. In other words, $V \subset V'$.
2. If there is an edge e_i connecting given keywords $v_i \rightarrow v_j$, in the Entity Unit subgraph, there should also be an edge from $v'_i \rightarrow v'_j$ connected by edge e_i in G'.
3. If no Entity Units subgraph covers all the keywords set V, we return Entity Units that cover the most number of keywords.

Definition 3. Ranking Function is a function adopted to evaluate the relatedness of candidate matches. Given parameter K which is the number of results the user wishes to retrieve, the algorithm finds Q searching matches where $K < Q$. The ranking function F is responsible for calculating their ranking scores and selecting the K most significant results among the Q candidates.

4. ARCHITECTURE

The emergence of Hadoop[15], Spark[16], Graph Databases [17][18][19] and many other distributed data storage and processing tools and technologies provide us an opportunity of processing a large volume of data distributed efficiently at a large scale. In Figure 1, we propose a conceptual keyword searching infrastructure which integrates Data Storage Module, Data Analyzing Module and Data Pre-processing Module. An application layer on the top presents to the end user a unified and easy to use interface without worrying about query language. As a consequence, the internal software framework, hardware infrastructure, and data communication are transparent to end users.

As Fig.1 describes, the infrastructure includes three main modules which can communicate with each other to accomplish data filtering, data loading, and data analyzing tasks.

1. Data Pre-processing Module is responsible for transforming the data such that the data is structured and can be loaded into back-end storage through Titan API[4].
2. Data Storage Module optimizes data repository and stores data distributed across multiple clusters.

[4]http://titan.thinkaurelius.com/

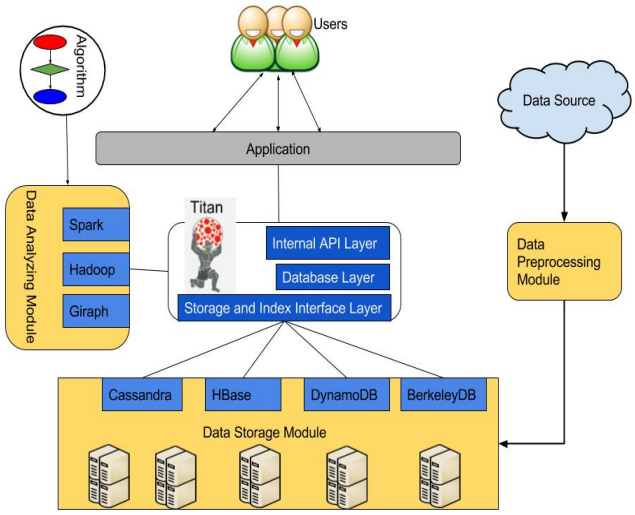

Figure 1: Distributed Key Word Search Infrastructure

3. Data Analyzing Module incorporates with ranking function and algorithms to discover the internal pattern and generate Top-K relevant results and return them to users.

5. HARDWARE IMPLEMENTATION

In this section, we demonstrate the concrete hardware implementation of building the Data Storage Module. Here, we choose Apache HBase[17], which is an open sourced, distributed and non-relational data storage as our back-end database. Based on that, we use Titan as our graph traversing tool which is also an open-sourced and optimized interface for querying graphs containing hundreds of billions of vertices and edges distributed and stored across the multi-machine environment[20]. Finally, based on HBase, we use an open-source cluster-computing framework Apache Spark[16] to distributively store the data into the back-end database.We present in Figure 2 how Spark and HBase intercorrelate and communicate with each other to accomplish distributive data storage task. According to Figure 2, since HBase is built based on HDFS[21] file system which is a distributed storage system, it can serve as sources for reading Spark Resilient Distributed Datasets(RDDs) and the destination for writing RDDs. Therefore, the original dataset can be first partitioned in memory using Spark and later distributively loaded into HBase to be saved on disks.

5.1 Data Source

In the following experiments, we use a small portion dataset comprising of 1 million facts of YAGO[22] knowledge base as our testing benchmark. We measure the data loading time by tuning the number of machines in the cluster. YAGO[22] is the semantic representation of Wikipedia[23], WordNet[24] and GeoNames[25] comprised of 120,000,000 triples. The accuracy of YAGO was evaluated to be above 95% using a sample of facts. Figure 3[26] is a graph representation provided by the YAGO official website about all the facts related to the search term "Elvis_Presley". In Table 1, we

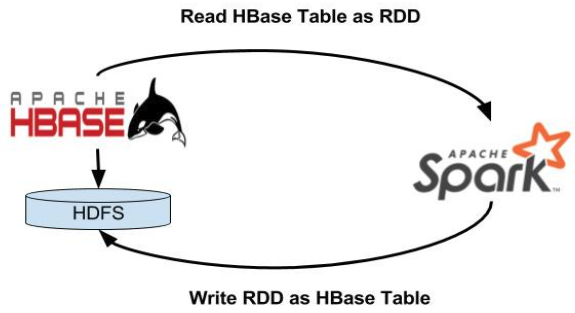

Figure 2: Apache Spark and Apache HBase Intercorrelation

list the sample facts from YAGO[22]. All the fact statements follow the formats of $< Subject, Predicate, Object >$ ($< S, P, O >$). From the graph representation in Figure 3, it is not hard to see that $< S, P, O >$ triples in YAGO are represented by vertex-edge-vertex format in the graph representation. Here, S and O are represented by vertices and P is the edge connecting the two vertices, which is quite applicable for storing in the graph database.

Table 1: Sample Facts from YAGO

Subject	Predicate	Object
Lachy_Hulme	actedIn	Macbeth_(2006_film)
Arenberg	isLocatedIn	Central_Europe
Neal_Kenyon	hasGender	male
Neal_Shusterman	created	Downsiders
Boyatt_Wood	isLocatedIn	Borough_of_Eastleigh
John_Steinbeck	wasBornIn	Salinas,_California
Lisa_Moretti	wasBornIn	Los_Angeles
Stavisky	isLocatedIn	France
Elvis_Presley	influences	Jack_Ketchum
Elvis_Presley	linksTo	Audi

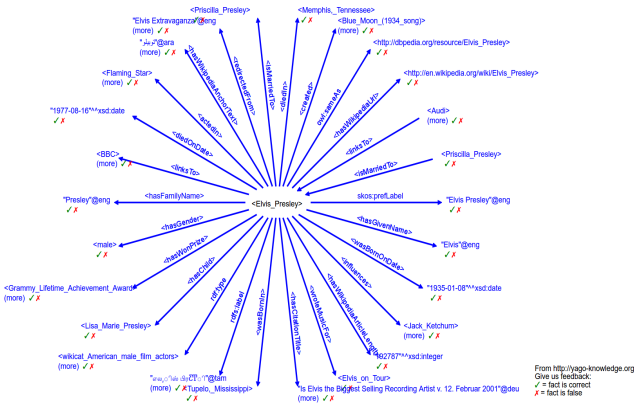

Figure 3: YAGO Demo

5.2 Testing Environment

We deploy our Data Storage Module on Amazon AWS cloud service and choose t2.xlarge EC2 as our instances. As depicted in Figure 4, we set up the multi-machine environment with one master and numerous slave nodes. In the subsequent experiments, we construct single master and starts from two slave nodes as our initial configuration. Then we gradually increase the number of slaves by two until we reach 10 slaves to record the i million data loading time.

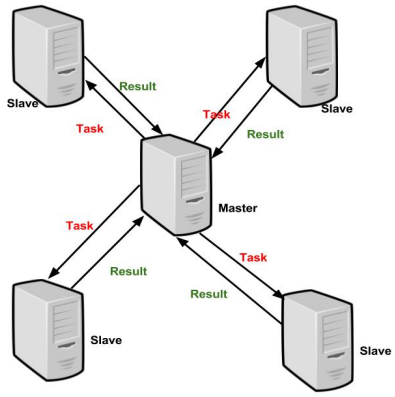

Figure 4: Master and Slave Architecture

5.3 Experimental Results

The experiments results are shown in Figure 5. We measure the data loading time in seconds for 1 million YAGO triples using a different number of machines. From Figure 5, we can identify that increasing number of machines decreases the loading time decreases significantly from 35.1s down to 13.74s. This implication is actually promising for our further research since it provides a scalable and unified way for loading and processing large dataset distributively.

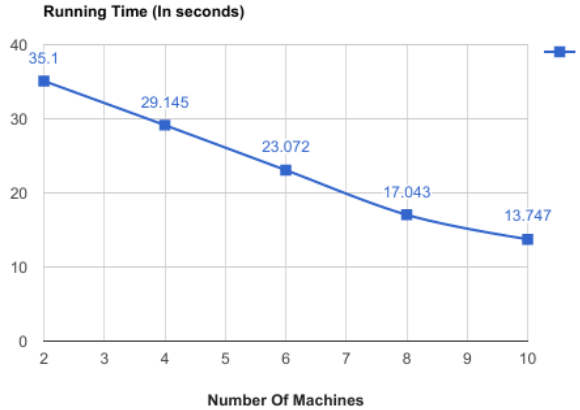

Figure 5: Number of Machines and Loading Time

6. CONCLUSIONS

Keyword searching over Big Data acts as a vital role and basis for multiple applications. The ever increasing data size, data complexity and data heterogeneity provides both challenge and opportunity for industries as well as academic disciplines. In this paper, we propose a software infrastructure supported by the distributed back-end hardware clusters to address the problem of keyword searching over Big Data. In the future, we plan to establish and integrate the framework collectively and finally provide end users on-demand service. Our project is still under progress. In this paper we particularly focus on the hardware implementation part. We will leave the rest for future research. Further work needs to be done to establish an efficient algorithm to perform keyword searching over large amount of data. Although the current study is based on a small sample of the dataset , the findings suggest that by tuning the number of machines and the configurations of each machine, we can decrease the loading time and searching time substantially. In the future, based on the hardware foundation we build an Entity-Unit keyword searching algorithm using the same experimental setup.

7. REFERENCES

[1] C. Lynch, "Big data: How do your data grow," *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.

[2] P. Joshi, I. Pathan, and A. Khan, "Keyword Generation for Search Engine Advertising," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 6, pp. 367–373, 2014.

[3] C. Bizer, P. Boncz, M. L. Brodie, and O. Erling, "The meaningful use of big data: Four perspectives – four challenges," *SIGMOD Rec.*, vol. 40, pp. 56–60, Jan. 2012.

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[5] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, pp. 77–87, Mar. 1997.

[6] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, (New York, NY, USA), pp. 158–166, ACM, 1999.

[7] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou, "Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 903–914, ACM, 2008.

[8] J. Shi, D. Wu, and N. Mamoulis, "Top-k relevant semantic place retrieval on spatial rdf data," in *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, (New York, NY, USA), pp. 1977–1990, ACM, 2016.

[9] L. Bo, L. Xianglong, and W. Li, *The Next-Generation Search Engine: Challenges and Key Technologies*, pp. 239–248. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[10] W. Dong, Z. Lei, and Z. Dongyan, "Top-k queries on rdf graphs," *Information Sciences*, vol. 316, pp. 201–217, 2015.

[11] Y. T. Yu, L Chang, "Scalable keyword search on large data streams," *IEEE 25th International Conference*, pp. 1199–1202, 2009.

[12] G. Piao, S. showkat Ara, and J. G. Breslin, "Computing the semantic similarity of resources in dbpedia for recommendation purposes," in *Joint International Semantic Technology Conference*, pp. 185–200, Springer, 2015.

[13] J. P. Leal, V. Rodrigues, and R. Queirós, "Computing semantic relatedness using dbpedia," in *OASIcs-OpenAccess Series in Informatics*, vol. 21, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[14] L. U. Quilitz, Bastian, *Querying Distributed RDF Data Sources with SPARQL*, pp. 524–538. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[15] Apache Software Foundation, "Apache hadoop version 2.6.5," 2016. `https://hadoop.apache.org`.

[16] Apache Software Foundation, "Aparche spark version 2.1.0," 2016. `http://spark.apache.org/`.

[17] Apache Software Foundation, "Aparche hbase version 1.3.0," 2017. `https://hbase.apache.org/`.

[18] Amazon, "Amazon dynamodb," 2012. `https://aws.amazon.com/dynamodb/`.

[19] Apache Software Foundation, "Aparche cassandra version 3.10," 2016. `http://cassandra.apache.org/`.

[20] Y. Mehta and S. Buch, "Semantic proximity with linked open data: A concept for social media analytics," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 337–341, April 2016.

[21] Apache Software Foundation, "Aparche hadoop hdfs," 2016. `http://hortonworks.com/apache/hdfs/`.

[22] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, ACM, 2007.

[23] Wikimedia Foundation, "Wikipedia," 2017. `https://en.wikipedia.org/wiki/Wikipedia`.

[24] University of Princeton, "Wordnet version 2.1," 2015. `https://wordnet.princeton.edu/`.

[25] www.geonames.org, "Geonames," 2017. `www.geonames.org`.

[26] Max Planck Institute for Informatics, "Yago: A high-quality knowledge base," 2017. `https://gate.d5.mpi-inf.mpg.de/webyago3spotlx/SvgBrowser`.

[27] U. S. Administration, "Data.gov," 2017. https://www.data.gov/.

[28] A. Passant, "Measuring semantic distance on linking data and using it for resources recommendations.," in *AAAI spring symposium: linked data meets artificial intelligence*, vol. 77, p. 123, 2010.