# An Efficient Query Retrieval From Different Domain Rdf Based on User Profile

**Mrs. Baby Mol Kurian[1] , E.Kaviya[2] , S.Kiruthika[3]**
[1]Assistant professor, Dept of CSE
[2, 3] Dept of CSE
[1, 2, 3] SRR Engineeering College,Chennai

***Abstract-*** *The main aim of this project is to construct a personalized semantic web search that can adaptively generalize profiles by queries given by users while respecting the specified privacy requirements.The large amount of heterogeneous linked data on the web poses new challenges to the database systems. We present methods extending a master RDF store which effectively combine the source information in RDF data. We specify a reliable and understandable way where results are fetched from the RDF data and delivered to the user.Subsequently we provide a profile based personalised web search that supports runtime profiling. We evaluate the presented methods and show that by the user preference based search user can make personalised search and the privacy of the user will be maintained . Finally, we show that query retrieval from different domain RDF based on user profile can also significantly improve the performance of query execution.*

***Keywords***- RDF, Linked Data, Personalised semantic web search.

## I. INTRODUCTION

The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the enormous variety of users' contexts and backgrounds, as well as the ambiguity of texts. The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. In profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be potentially effective for almost all sorts of queries, but are reported to be unstable under some circumstances.

## II. RELATED WORK

Minh Duc Nguyen1et.al[7] provided a detailed review of resource description framework, RDF is a standard language model for representing semantic data. As the concept of semantic web becomes more viable, the ability to retrieve and exchange semantic data will become increasingly more important. Systematic organisation of RDF data is a key issues in many RDF organisation have been proposed with data storage architectures and query processing for data retrieval. However, most of the proposed approaches require many join operations that result in the unnecessary processing of intermediate results for SPARQL queries. The data processing becomes considerable as the RDF data volume is increased. In this paper, an efficient structural index and a query optimizer to process queries without join operations is provided..Likewise Io Taxidou Peter M. Fischeret.al[13] done research on information diffusion in social media has attracted a lot of attention, since the produced data is fast, massive and vital. Additionally, the source of such data is important because it helps to judge the relevant and trustworthiness of the information enclosed in the data. Social media currently provide a mechanisms for source information, targeted to specific use cases. In this paper, a model for information diffusion and source data,based on the W3C PROV Data Model is provided. The advantage is that PROV is a web-native and interoperable format that allows easy publication of provenance data, and minimizes the integration effort among different systems making use of PROV.P.Pediaditi ICS-FORTHet.al[10] provided the work on the notion of RDF named graphs has been proposed in order to assign provenance information to data described using RDF triples. In this paper, they argue that named graphs alone cannot capture provenance information in the presence of RDF reasoning and updates. The RDF graph sets is organised with a set of RDF named graphs which contain the triples by the named graphs that constitute the graphset. They formalise the RDF named graphs and RDF graph sets and propose query and update languages that can be used to handle provenance information for RDF graphs taking into account RDFS semantics.David Beckettet.al[4] reviews syntaxes for RDF as denied in RDF Model describes the problems that remain after the revising. These include not clearly showing the RDF triple model and not working very well with newer XML technology such as XSLT and W3C XML Schema.. The paper then constructs new syntaxes in the two main uses as a end user syntax. It summarises existing approaches and discusses using XML or non- XML formats and then describes two new syntaxes, an outline XML one and a new textual RDF syntax N-Triples Plus based on the N-Triples test case syntax.

## III. RESOURCE DESCRIPTION FRAMEWORK

Resource Description Framework (RDF) is constructed for semantic data on a Relational Database containing Structured as well as unstructured data. A Schema is identified for the relational database and a RDF representing the schema of the database is constructed through model provided by the jena api. The Model contains all the information's about the data linkages in the schema. In this process the schema can also be altered based on admin requirement so that the search process can be effective.

The RDF is also generated by mining the text contents uploaded by the users in blogs and the contents of the file are analyzed and the meta contents are manipulated. The Text mining process analyses the text word by word and also picks up the literal meaning behind the group of words that constitute the sentence. The Words are analyzed in Word Net api so that the related terms can be found for use in the meta content in generation of RDF. The user uploaded content will also be analyzed in real-time servers in their own natural language processing strategies and the results are obtained in a RDF format so that it can be understood by other Servers.

Similar data are grouped together that relate to the same resource. The data level process is subjected to structure level processing by indexing the semantic data elements. Multiple RDFs are grouped and structured together to form a master RDF data that holds all the semantic information of a Server that support reasoning in any formats of query processing. The Different resources are interlinked with high degree of relational factors by the predicates in the triples. The Query processing is handled directly in the RDF file by iterating the triples forming a discrete relation with the Service query and the URI representing the location of the resource is returned. As this process is handled in web services in real time servers. Hence the structure-oriented approach to RDF data management where data partitioning and query processing make use of structure patterns generated by the RDF.

## IV. PERSONALISED SEMANTIC WEB SEARCH

The existing profile based personalised web search do not support runtime profiling. A user profile is typically generalised for only once offline, and used to personalize all queries from a same user. Such one profile fits all strategy certainly has disadvantages given the variety of queries. A better approach is to make an online decision on a. whether to personalize the query (by exposing the profile) and b. what to expose in the user profile at runtime. Previous works on profile based PWS mainly focus on improving the search utility and do not take into account the customization of privacy requirements. This probably makes some user privacy to be overprotected while others insufficiently protected. Many personalization techniques require iterative user interactions when creating personalized search results. To the best of our knowledge, no previous work has supported such feature.

UPS (literally for User customizable Privacy-preserving Search) framework. The framework assumes that the queries do not contain any sensitive information, and aims at protecting the privacy in individual user profiles while retaining their usefulness for PWS. UPS consists of a non trusty search engine server and a number of clients. Each client (user) accessing the search service trusts no one but himself/ herself. The key component for privacy protection is an online profiler implemented as a search proxy running on the client machine itself. The proxy makes the complete user profile, in a hierarchy of nodes with semantic data, and the user specified privacy requirements represented as a set of sensitive nodes. We provide a privacy preserving personalized web search , which can generalise profiles for each query maintaining user privacy.We provide an inexpensive mechanism for the user to decide whether to personalize a query. This can be made before each runtime profiling to enhance the speed of the search results while avoid the unnecessary exposure of the profile.

We propose an efficient and scalable distributed RDF data management system. A structure oriented approach that exploits the structure patterns exhibited by the underlying data captured using a structure index. Height- and label-parameterized structure index for RDF. For capturing the structure of the underlying data, we propose to use the structure index, a concept that has been successfully applied in the area of XML- and semi structured data management. It is basically a graph, where vertices represent groups of data elements that are similar in structure. For constructing this index, we consider structure patterns that exhibit certain edge labels containing path.. Further, we propose to leverage it for RDF data partitioning. To obtain a contiguous storage of data elements that are structurally similar, vertices of the structure index are mapped to tables. The triples with the same property label, triples with subjects that share the same structure are physically grouped. A basic strategy is to match the query against the structure index first to identify groups of data that satisfy the query structure. Then, via standard data-level processing, data in these relevant groups are retrieved and joined. Instead of performing structure- and data-level operations successively and independent from each other like in this basic strategy, we further propose an integrated strategy that aims at an optimal combination of these two types of operations.

***Example query of RDF generation***

<rdf:RDF>
<rdf:description rdf:about='http://localhost9999/Resources/Maldives/Maldives.html'>
<j.0:gnpOf>199.00</j.0:gnpOf>
<j.0:continentOf>Asia</j.0:continentOf>
<j.0:surfaceareaOf>298.00</ j.0:continentOf>
<j.0:codeOf>MDV</j.0:codeOf>
<j.0:nameOf>Maldives</j.0:nameOf>

***Querying over master RDF with profiling***

The online phase handles queries as follows:

1. When a user issues a query on the client, the proxy generates a user profile in runtime in the light of query terms. The output of this step is a generalized user profile satisfying the privacy requirements.
2. Subsequently, the query and the generalized user profile are sent together to the PWS server for personalized search.
3. The search results are personalized with the profile and delivered back to the query proxy.
4. Finally, the proxy either presents the raw results to the user, or reranks them with the complete user profile. As the sensitivity values explicitly indicate the user's privacy concerns, the most straightforward privacy preserving method is to remove subtree rooted at all sensitive-nodes whose sensitivity values are greater than a threshold. Such method is referred to as forbidding.

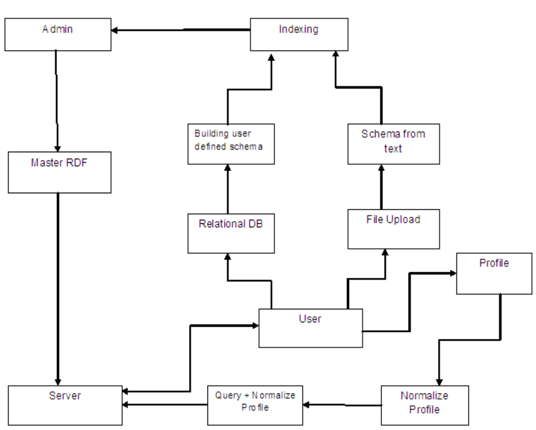


*Figure 1 Architecture of querying over master RDF with profiling*

***Generic query execution algorithm***

**Input:** cNode common node

**Data**: EP-> execution plan of the query's stores intermediate result of each query

M->Mapping query

**Output:** mappings of nodes between query & RDF data
1: EP.getNext() // get next query to be processed
2: cMatchList M(cNode) // get match list of common node
3: for each cnt getConnectorList(cMatchList) do
4: for each mVal cMatchList.getMatchValues(cnt) do
5: nextNode<- getNextNode( cNode)
6: nextNodeMatchList<-findMatches(nextNode, mVal)
7: if there is mapping of nextNode then
8: M.addMapping(nextNode, mVal, nextNodeMatchList) // mVal is now the connector of nextNode
9: else
10: remove(mVal) // remove this matching value and its // corresponding connectors
11: end
12: end
13: end
14: if there is any match for answer then
15: nCommonNode-> findNextCommonNode
16: end

## V. EXPERIMENTAL RESULTS

***Query preparation***

To prepare for the experiment, RDF data are generated that correspond to three different categories of data. They store information about a) Country (Q1), b) eduation(Q2), c) food(Q3). As a result, query characteristics vary greatly.

Query Q1- retrieves data associated with countries, which have the properties, i.e.country code,region,capital. Query Q1 is representative of simple queries .

Query Q2- retrieves data associated with education, which have properties, such as maths,science.

Query Q3- retrieves data associated with food, including gravy,sambar.

In summary, the queries represent the entire spectrum of different data areas included in the dataset of RDF . The complexity of each query varies with number of patterns.

Therefore, queries of various lengths and different ranges of the dataset are used to verify query processing performance. A complete set of queries is not used to cover all data in the RDF dataset, however, the results are indicative of overall performance.

We conduct the experiment by executing the above three queries and calculating the running time . With our structural index, RDF data for all types of query patterns can be retrieved. Since the execution plan is built as an ordered sequence of query patterns, the matched data for each query is found. By excluding unnecessary results at each processing step, intermediate results are reduced significantly, which improves running time. The objective of our experiment is to verify the efficiency of processing RDF data by evaluating the execution time of the queries. To demonstrate the performance consistency, the running time for each of the three queries that are of different complexities and extract different data domains. As shown in Figure 2, each query has a relatively stable execution time and it shows the diversity of queries and low variance on the processing times of each query.

*Table 1:Queries average running time*

| S.no | Query - 1 | Query - 2 | Query - 3 |
|---|---|---|---|
| 01 | 38 | 57 | 72 |
| 02 | 39 | 60 | 75 |
| 03 | 36 | 56 | 76 |
| 04 | 40 | 60 | 74 |
| 05 | 37 | 59 | 75 |

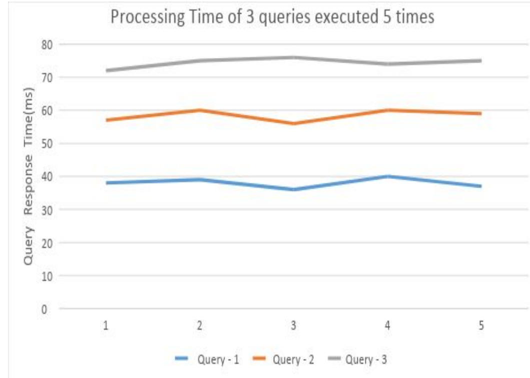


*Figure 2:Processing time of 3 queries executed 5 times*

## VI. CONCLUSION

Thus query retrieval from different domain RDF can be done efficiently based on user runtime profiling where the privacy of the user will be protected. The result will be given to the user by normalising the user profile and the related answer for the user query will be fetched from the RDF and delivered to the user. By providing user preference based web search the user can make personalized search and the privacy of the user will be maintained. The future enhancement that can be made in user preference based search where the personalised user profile can be maintained by the admin by rectifying all the privacy issues.

## REFERENCES


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

[2] P. Cudre-Mauroux, S. Agarwal, and K. Aberer, “GridVine: An infrastructure for peer information management,” IEEE Internet Comput., vol. 11, no. 5, pp. 36–44, Sep./Oct. 2007.

[3] M. Wylot, J. Pont, M. Wisniewski, and P. Cudre-Mauroux. (2011). dipLODocus[RDF]: Short and long-tail RDF analytics for massive webs of data. Proc. 10th Int. Conf. Semantic Web - Vol. Part I,pp. 778–793 [Online]. Available: http://dl.acm.org/citation.cfm?id=2063016.2063066

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

[5] M. Wylot, P. Cudre-Mauroux, and P. Groth, “Executing provenance- enabled queries over web data,” in Proc. 24th Int. Conf.World Wide Web, 2015, pp. 1275–1285.

[6] B. Haslhofer, E. M. Roochi, B. Schandl, and S. Zander. (2011).Europeana RDF store report. Univ. Vienna, Wien, Austria, Tech.Rep. [Online]. Available:

http://eprints.cs.univie.ac.at/2833/1/europeana_ts_report.pdf

[7] Minh Duc Nguyen1.et.al“”SPARQL Query Optimization for Structural Indexed RDF Data in Proc. Int. Semantic Web Conf.,2004, pp. 274–288.

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

[9] B. Liu and B. Hu, “An Evaluation of RDF Storage Systems for Large Data Applications,” in Proc. 1st Int. Conf. Semantics, Knowl.Grid, Nov. 2005, p. 59.

[10] P. Pediaditis ICS-FORTH, “On Explicit Provenance Management in RDF/S Graphs” VLDB J. Int. J. Very Large Data Bases, vol. 24, no. 1, pp. 67–91, 2015.

[11] C. Weiss, P. Karras, and A. Bernstein, “Hexastore: sextuple indexing for semantic web data management,” Proc. VLDB Endowment,vol. 1, no. 1, pp. 1008–1019, 2008.

[12] T. Neumann and G. Weikum, “RDF-3X: A RISC-style engine for RDF,” Proc. VLDB Endowment, vol. 1, no. 1, pp. 647–659, 2008.

[13] Io Taxidou Peter M Ficher, “Modeling Information Diffusion in Social Media as Provenance with W3C PROV,” in Proc. IEEE 3rd Latin Am. Web Congr., 2005, pp. 71–80.

[14] M. Atre and J. A. Hendler, “BitMat: A main memory bit-matrix of RDF triples,” in Proc. 5th Int. Workshop Scalable Semantic Web Knowl. Base Syst., 2009, p. 33.

[15] 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, 2003, pp. 131–150.