Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

SPARQL2Hive: Translating SPARQLqueries on Hive using A meta-model-based techniques

[1]P. ANITHA, [2]Naga Jyothi Dhulipalla, [3]Veneela Aladi, [4]Rama Devi Bogani, [5]K. Narayana Rao

[1,2,3,4]Department of Computer Science and Engineering, QIS College of Engineering & Technology, Ongole, AP, India, E-Mail qispublications@qiscet.edu.in

[5]Department of Computer Science and engineering, Rise Krishna Sai Prakasam Group Of Institutions, Ongole, AP, India

Article Info
Page Number: 116-125
Publication Issue:
Vol 69 No. 1 (2020)

Article Received: 10 August 2020
Revised: 15 September 2020
Accepted: 20 October 2020
Publication: 25 November 2020

Abstract— As a result of correctly querying RDF data, new problems have been created by the extension of Web documents. Traditional RDBMS can successfully adapt and search for scattered data. With the development of Hdfs and its use of the Mapreduce Model with the Hive database engine, the ethics of data collection and retrieval have altered. In this work, we introduce SPARQL2Hive, a MapReduce-based, cost-effective SPARQL querying programme that enables ad hoc SPARQL querying parsing on massive RDF networks. Instead of directly translating from one language to another, SPARQL2Hive uses Hive's parser as a bridge between SPARQL and MapReduce. A data warehousing platform called Hive is built on top of parallel processing and uses Hadoop to search computers. This extra degree of virtualization renders our method agnostic of the present version of Hdfs, ensuring interoperability with potential updates to the Hadoop platform since they will be handled by the underpinning Hive level. Our strategy is to employ the SPARQL and Hive conceptual and suggest a conversion between them using the ATL vocabulary. SPARQL2Hive is compared to Apache hadoop SPARQL solutions.

Keywords—Semantic Web, RDF, ATL, SPARQL, Hive.

I. INTRODUCTION

The prominence of the Future Internet has grown in both academia and industry over the last few centuries.DRF is a W3C-standardized framework for the Future Internet, and Elasticsearch is the nosql database used to retrieve DRF data. Because DRF is extensively utilised as a metadata structure in implementations, effective data warehousing and information retrieval approaches for DRF data must be explored. The W3C recommends SPARQL as a retrieval interface for DRF and Ontology libraries. It's intended for usage on the World wide web statistics, allowing for searches across a multitude of sources, regardless of formatting. It is easy to expand SPARQL searches due to unexpected information sources.The expansion of Information retrieval has posed new issues in terms of querying DRF data properly. Typical data models adjust and retrieve scattered data effectively. The economics of data acquisition and retrieval have altered with the creation of Hdfs and its execution of the Proposed Method with Swarm, a database engine.

Information Systems Engineering provides several product and procedure approaches and concepts for creating useful information systems. Since its a advent of big information,

business intelligence has to discover ways to handle huge amounts of information in a handful of formats [1]. Model-Driven Engineering (IDM)has developed like a matter of debate between computer experts in recent years, in both the United States and elsewhere.both in academia and in enterpriseBy providing for a higher conceptual focus than standard computing, IDM has enabled numerous notable advancements in the construction of complex systems [2]. It is a type of generative engineering in that all are a part of an algorithm is generated.Patterns are used to develop applications.A prototype is an ideal, a simulation of a system that's also adequate to comprehend the modelled system and address queries that one has about it [3]. Various ideas that are connected to one another can be used to explain a system. On this same those certain hand, conceptual search innovations have accomplished excellent achievement, because they're no longer capable of handling huge quantities of information [4]. To address this issue, Big Data can store sizeable amounts of information big props to very effective tools at the storage capacity levels such as NoSQL, Apache Hive, and Prompt, but we still need a way to patch and turn between such 2 components [5].In this work, we offer SPARQL2Hive, a scalable solution climate model engineering that allows us to convert sophisticated SPARQL searches into Hive programmes [6]. This article is structured as follows: The second part gives an outline of the relevant work. The RDF, SPARQL, and Modeling Driven technologiesThe final segment discusses design tools. The fourth section is devoted to the exposition ofthe strategy To end, the findings of the tests are shown and analysed in section 5.the efficacy of our strategy The final section offers the findings and views.

II. RELATED WORKS

Numerous previous studies have advocated employing Big Information platforms such as Hadoop, Hdfs, Mongodb, and others to overcome problems of scaling of Successful Utilization in the field [7]. Hive The Hive platform is a technology for building enormous data centers [8]. It is built on an additional storage named memstore, which interfaces with other NoSQL databases (mostly HBase) and an elasticsearch called HiveQL, which is similar to SQL. For Big Information analytics. [9] developed and contrasted Beehive, Pig, and Hadoop.The determination is centered on every product's kind of vocabulary, graphical user, accessible methods, and volume of data. We published a research in [10] on the usage of Mongodb monitoring and managing huge Data sets using the 4 Json concepts: Online transaction processing stores centered columnar, important directed, memorandum, and bar chart.The research [11] presented an outline on DRF triplestores premised on Datastores, as well as a brief summary of the information retrieval instruments in each experimental data. Between them is PigSPARQL [12], which is a method for treating Data sets. PigSPARQL's position is to plot and translate Sql statements together into PigLatin[13] dialect curriculum.Jena-HBase [14], a highly scalable RDF storage founded on HBase[15], the NoSQL dbms, seems to be another Domain Specific approach based on Big Information technology. Graph database is a paragraph database that stores data in HDFS and processes data using Dremel. Its design is aimed at large-scale data administration in a fully decentralized environment [16]. The architecture of this system is split into 2 parts: storing

and questioning. The RDF documents are stored in Hdfs databases, and the Maya Foundation is used to conduct SPARQL queries[17].

III. BACKGROUND

Domain Reference Format(DRF) is a W3C protocol that was created to link computer metadata to Web content. DRF examples are defined as a collection of sequences that form a logical information. An DRF trio is made up of three parts: Frequently used topic, condition, and argument (S, P, O):

The topic is a URI (Universal Resource Identification), which is an identification for a website. A property can be a digitally text, a section of a homepage, a group of webpages, or even an item that is not Internet but can be assigned a URI.The precondition is a feature that defines the item. Each field outlines its permissible meanings, the sorts of assets it may represent, and the interactions it has with other variables.The price of the asset is an item, which might be a URI (i.e., a resource) or a textual integer. Figure 1 is an instance of a triple RDF. An RDF triple may be used to convey the phrase "social compact authored by william daniel thoreau."

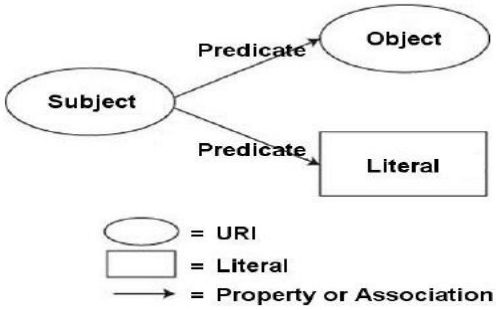

Fig. 1. Triple RDF

The W3C recommends the SPARQL Interface and RDF Linguistic Search (SPARQL). This vocabulary not only allows you to describe searches on DRF databases, but it also allows you to select the format of the results. As a result, it is a general programming vocabulary for any forms of data as long as they are specified in DRF. The SPARQL system not only provides for the interrogation of an RDF network via its Search verb, but also the creation of a fresh graph from chosen components via its Build stipulation. This decisions that enable SPARQL to be seen as a transformational rules languages for RDF-based modeling technique. Test Driven Technology [19] puts the paradigm at the centre of the construction phase, allowing it to transition from a meditative position to an uniting role in relation to other tasks in the computer enterprise procedure. The Breakcore should consequently be viewed as a technique of integrating disparate technological domains in order to get toward sophisticated software manufacturing. Prototype development then offers ways to modelling, metamodeling, topic selection, conversion, and formulate and implement to help in the development of this new software. These techniques are complemented by development techniques and coding tools, but also by testing and evaluation of the frames' compliance to metamodels [20].

IV. PROPOSED METHODOLOGY

We now explain our strategy, which consists of three fundamental characteristics: an origin meant to represent, the SPARQL metamodel, and a destination metamodel, the Hive effectivenes. The third component is the conversion between these two modeling technique, which we do using the language ATL [18], as seen in Picture 2. Each SPARQL query provided by a client adheres to our SPARQL conceptual schema, therefore this demand have to go through a transition utilising the ATL languages that turns the SPARQL clauses into Hive clauses, and this Hive programme will also comply to our Hive conceptual schema.

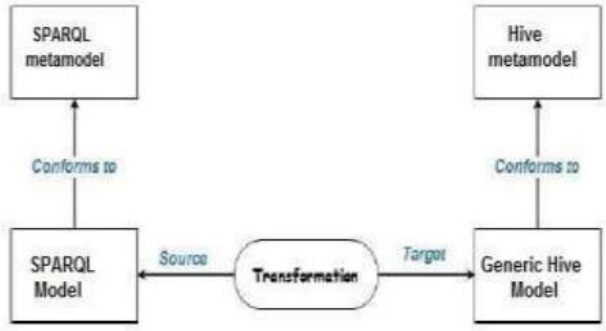

Fig. 2. Proposed Method

To create a SPARQL object model, it is vital to understand the basic architecture of the queries. Initially, this design is similar to that of the Sql queries. Furthermore, there are three different types of SPARQL queries: Choose, Build, and Request. We will start with the Choose query because it is the most common and important. This question can extricate Data sets according to criteria outlined in the Within which stipulation, so it is a question. The essential phrases appear in every SELECT query: Pick, Origin, and the conditional statement WHERE.

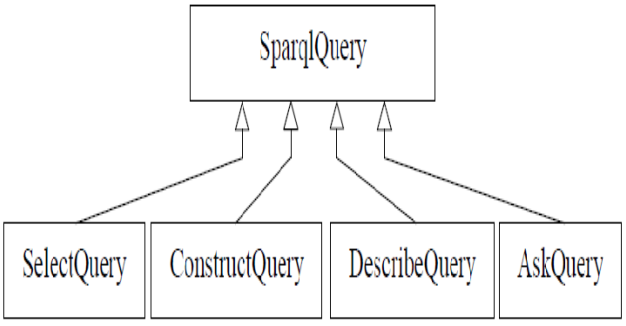

Fig. 3. SPARQL Metamodel

Hive is Hdfs "data store," providing a vocabulary for extracting "interpersonal" architecture from quasi or incomplete information (flat files, JSON, web logs, Hbase, Cassandra ...). With, Possessing, Part On, and Sort According to are all keywords of a Hadoop Sql search. Our Bee masing - masing is depicted in the diagram beneath.

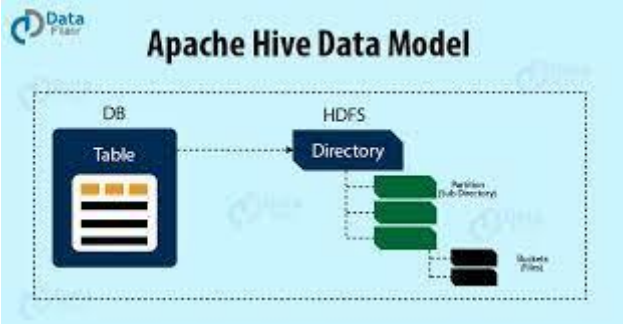

Fig. 4. Hive Metamodel

The instructions that specify whether components of the set of data are recognised and navigated to construct the aspects of the base classifier (prototype process (designated Adapter)) are contained in an ATL[21] programme. ATL presents an extra type of inquiry, in add here to conversions of existing theories, that allows users to describe inquiries on the models (conversions of type arrangement towards language (called Enquiry)). The ATL word's dual nature (unambiguous and propulsive) is one of its distinguishing features[22]. The prescriptive portion allows a component of the transformation's origin conceptual schema to be explicitly associated with a component of the transformation's destination conceptual schema.QVT [23] was also presented by Wow as part of the Malondialdehyde methodology (Query, Look, Transition). A inquiry is a statement that accepts a modelling source and picks specified components from it. A paradigm that draws from other representations is called a perspective.An intake version is transformed to change it or build a new ve rsion.

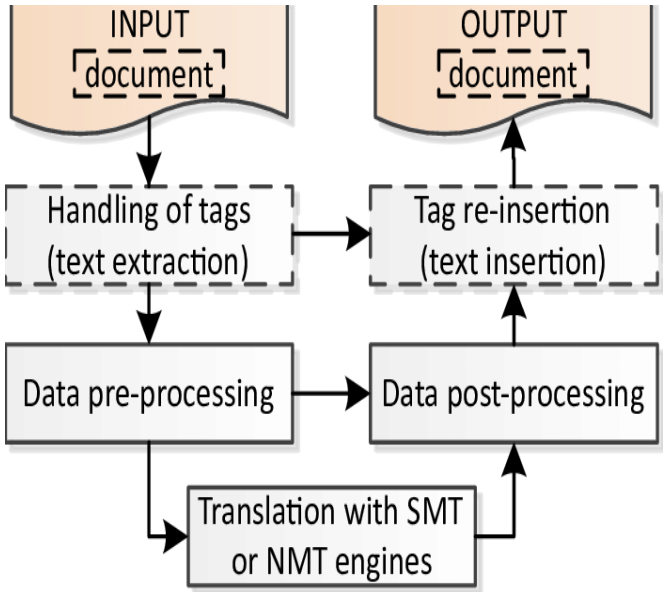

Fig. 5. Translation Machine proess

ATL programme comprises of instructions that indicate how the destination figure's components must be constructed depending on the parent figure's components. The paradigm is always followed while establishing these criteria.The effect of applying this adjustment is shown in the schematic diagram (Fig.6)

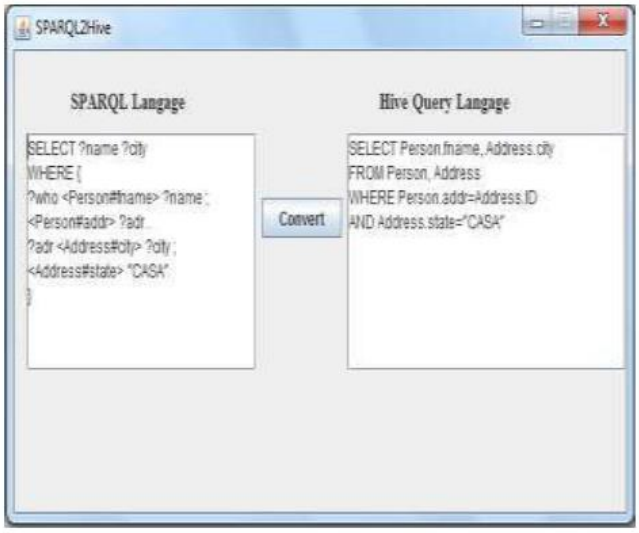

Fig. 6. Converting a SPARQL query to a Hive application using an example

V. RESULTS AND DISCUSSION

We utilised three occurrences of the LUBM[24] Benchmarking to assess the effectiveness of our technique. To further understand our SPARQL2Hive technology, the eight LUBM searches are conducted on such three databases of distinct sized. The settings and background of our encounters, the edition of Hive, and the specifics of the databases are all presented in the first section. The findings will be analyzed after that. Furthermore, we shall assess the influence of the sampling data size on the data modelling performance, and then we shall explain the outcomes of this assessment, that can be displayed visually and would demonstrate the productivity of our SPARQL2Hive technology. SPARQL2Hive runs on Hdfs 3.xy with Hive 3.1.0 on a computer with a 2.3 Gigahertz CPU that can hold up to 4 Terabytes of solid disc space and 16 Gigabytes of Memory. Those three features, LUBM1, LUBM2, and LUBM5, were utilised in this research and had the corresponding sextuplets numbers: These two databases have 139 billion references, 277 billion threefold, and 488 billion threefold, respectively, and their sizes are 8.5 Gigabytes, 22, 78 Gigabytes, and 56, gigabytes. The following table summarises the findings received for the constraints of these 3 matches.We evaluate our SPARQL to Hive technology to Maya by running LUBM Benchmarking searches utilising the 3 databases LUBMI, LUBM2, and LUBM5. SPARQL2hive is much more efficient than Maya at the performance of LUBM Benchmark searches. The comparative outcomes for all LUBM searches are shown in Figure 7.Based on the above findings, we may infer that SPARQL2Hive is a modular, resilient, and fault-tolerant platform. These findings demonstrate the utility of SPARQL2Hive for dealing with large amounts of DRF data [25]. It doesn't take long for SPARQL2Hive to put the information. Since it converts a Sql statements to a HiveQL application in an easy manner. Contrary to the Jasper architecture, whose functioning is a tad more tricky since the demand goes through a series of processes that take forever, notably for uploading preparation and analysis information for restoration, and Jasper consumes a large amount of assets like Memory.

TABLE I. DATASETS LOADING TIME

Dataset	**LUBM1**	**LUBM2**	**LUBM5**
Loading time in milliseconds	1.18	2.95	2.7

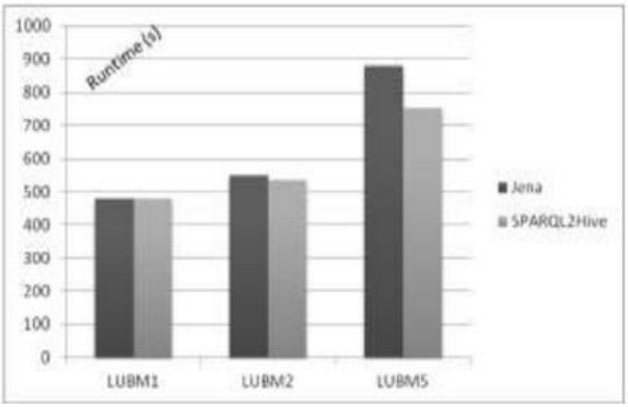

Fig. 7. Runtime using LUBM1

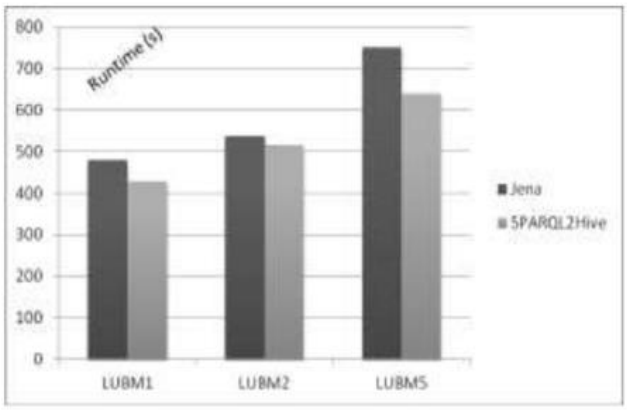

Fig. 8. Runtime using LUBM2

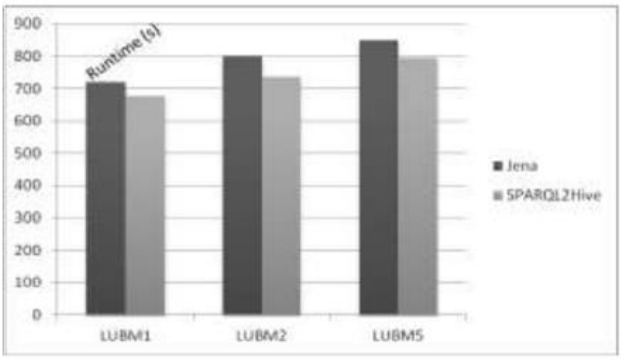

Fig. 9. Runtime using LUBM5

VI. CONCLUSION AND FUTURE SCOPE

The growing volume of DRF data presents academics with a new problem in maintaining such a big quantity of DRF data. Big Digital retention technologies such HBase [14] are the focus of study, and querying technologies such Hive are used for administration. In this paper, we introduce SPARQL2Hive, a novel structure founded on the conceptual approach for transforming SPARQL searches into HiveQL programmes.

REFERENCES

1. H. Thakkar, R. Angles, M. Rodriguez, S. Mallette and J. Lehmann, "Let's build Bridges, not Walls: SPARQL Querying of TinkerPop Graph Databases with Sparql-Gremlin," 2020 IEEE 14th International Conference on Semantic Computing (ICSC), 2020, pp. 408-415, doi: 10.1109/ICSC.2020.00080.
2. M. Atzori, "Computing Recursive SPARQL Queries," 2014 IEEE International Conference on Semantic Computing, 2014, pp. 258-259, doi: 10.1109/ICSC.2014.54.
3. K. M. Kyu and A. N. Oo, "Enhancement of Query Execution Time in SPARQL Query Processing," 2020 International Conference on Advanced Information Technologies (ICAIT), 2020, pp. 153-158, doi: 10.1109/ICAIT51105.2020.9261805.
4. S. Malik, A. Goel and S. Maniktala, "A comparative study of various variants of SPARQL in semantic web," 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), 2010, pp. 471-474, doi: 10.1109/CISIM.2010.5643493.
5. M. Banane, A. Erraissi and A. Belangour, "SPARQL2Hive: An approach to processing SPARQL queries on Hive based on meta-models," 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), 2019, pp. 1-5, doi: 10.1109/ICMSAO.2019.8880393.
6. W. Li et al., "SHOE: A SPARQL Query Engine Using MapReduce," 2013 International Conference on Parallel and Distributed Systems, 2013, pp. 446-447, doi: 10.1109/ICPADS.2013.78.
7. X. Zhang and J. Van den Bussche, "On the Power of SPARQL in Expressing Navigational Queries," in The Computer Journal, vol. 58, no. 11, pp. 2841-2851, Nov. 2015, doi: 10.1093/comjnl/bxu128.
8. F. Bamashmoos, I. Holyer, T. Tryfonas and P. Woznowski, "Towards Secure SPARQL Queries in Semantic Web Applications Using PHP," 2017 IEEE 11th International Conference on Semantic Computing (ICSC), 2017, pp. 276-277, doi: 10.1109/ICSC.2017.29.
9. L. Lv, H. Jiang and L. Ju, "Research and Implementation of the SPARQL-TO-SQL Query Translation Based on Restrict RDF View," 2010 International Conference on Web Information Systems and Mining, 2010, pp. 309-313, doi: 10.1109/WISM.2010.44.
10. H. Wang, Z. M. Ma and J. Cheng, "fp-Sparql: An RDF fuzzy retrieval mechanism supporting user preference," 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012, pp. 443-447, doi: 10.1109/FSKD.2012.6234114.

11. X. Chen, T. Wu, Q. Xie and J. He, "Data Flow-Oriented Multi-Path Semantic Web Service Composition Using Extended SPARQL," 2017 IEEE International Conference on Web Services (ICWS), 2017, pp. 882-885, doi: 10.1109/ICWS.2017.112.
12. J. M. Almendros-Jiménez, A. Becerra-Terón and G. Moreno, "A fuzzy extension of SPARQL based on fuzzy sets and aggregators," 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2017, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2017.8015411.
13. R. Gupta and S. K. Malik, "SPARQL Semantics and Execution Analysis in Semantic Web Using Various Tools," 2011 International Conference on Communication Systems and Network Technologies, 2011, pp. 278-282, doi: 10.1109/CSNT.2011.67.
14. H. Jabeen, E. Haziiev, G. Sejdiu and J. Lehmann, "DISE: A Distributed in-Memory SPARQL Processing Engine over Tensor Data," 2020 IEEE 14th International Conference on Semantic Computing (ICSC), 2020, pp. 400-407, doi: 10.1109/ICSC.2020.00079.
15. D. A. C. Amat, C. Buil-Aranda and C. Valle-Vidal, "A Neural Networks Approach to SPARQL Query Performance Prediction," 2021 XLVII Latin American Computing Conference (CLEI), 2021, pp. 1-9, doi: 10.1109/CLEI53233.2021.9639899.
16. D. S. Ferru and M. Atzori, "Write-Once Run-Anywhere Custom SPARQL Functions," 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), 2016, pp. 176-178, doi: 10.1109/ICSC.2016.64.

A. Kashlev and A. Chebotko, "SPARQL-to-SQL Query Translation: Bottom-Up or Top-Down?," 2011 IEEE International Conference on Services Computing, 2011, pp. 757-758, doi: 10.1109/SCC.2011.79.

17. X. Zhai, L. Huang and Z. Xiao, "Geo-spatial query based on extended SPARQL," 2010 18th International Conference on Geoinformatics, 2010, pp. 1-4, doi: 10.1109/GEOINFORMATICS.2010.5567605.
18. M. Krommyda and V. Kantere, "Understanding SPARQL Endpoints through Targeted Exploration and Visualization," 2019 First International Conference on Graph Computing (GC), 2019, pp. 21-28, doi: 10.1109/GC46384.2019.00012.
19. J. M. Almendros-Jiménez, A. Becerra-Terón, G. Moreno and J. A. Riaza, "Tuning Fuzzy SPARQL Queries in a Fuzzy Logic Programming Environment," 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2019, pp. 1-7, doi: 10.1109/FUZZ-IEEE.2019.8858958.
20. K. M. Nguyen, T. -H. Nguyen and X. H. Huynh, "Automated translation between RESTful/JSON and SPARQL messages for accessing semantic data," 2016 International Conference on Electronics, Information, and Communications (ICEIC), 2016, pp. 1-4, doi: 10.1109/ELINFOCOM.2016.7562981.
21. J. Euzenat, A. Polleres and F. Scharffe, "Processing Ontology Alignments with SPARQL," 2008 International Conference on Complex, Intelligent and Software Intensive Systems, 2008, pp. 913-917, doi: 10.1109/CISIS.2008.126.
22. S. Chun, S. Seo, W. Ro and K. -H. Lee, "Proactive Plan-Based Continuous Query Processing over Diverse SPARQL Endpoints," 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015, pp. 161-164, doi: 10.1109/WI-IAT.2015.168.
23. T. Chawla, G. Singh and E. S. Pilli, "A shortest path approach to SPARQL chain query optimisation," 2017 International Conference on Advances in Computing,

Communications and Informatics (ICACCI), 2017, pp. 1778-1778, doi: 10.1109/ICACCI.2017.8126102.

24. N. Kumar and S. Kumar, "Querying RDF and OWL data source using SPARQL," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-6, doi: 10.1109/ICCCNT.2013.6726698.
25. P Ramprakash, M Sakthivadivel, N Krishnaraj, J Ramprasath. ”Host-based Intrusion Detection System using Sequence of System Calls” International Journal of Engineering and Management Research, Vandana Publications, Volume 4, Issue 2, 241-247, 2014
26. N Krishnaraj, S Smys.”A multihoming ACO-MDV routing for maximum power efficiency in an IoT environment” Wireless Personal Communications 109 (1), 243-256, 2019.
27. N Krishnaraj, R Bhuvanesh Kumar, D Rajeshwar, T Sanjay Kumar, Implementation of energy aware modified distance vector routing protocol for energy efficiency in wireless sensor networks, 2020 International Conference on Inventive Computation Technologies (ICICT),201-204
28. Ibrahim, S. Jafar Ali, and M. Thangamani. "Enhanced singular value decomposition for prediction of drugs and diseases with hepatocellular carcinoma based on multi-source bat algorithm based random walk." Measurement 141 (2019): 176-183. https://doi.org/10.1016/j.measurement.2019.02.056
29. Ibrahim, Jafar Ali S., S. Rajasekar, Varsha, M. Karunakaran, K. Kasirajan, Kalyan NS Chakravarthy, V. Kumar, and K. J. Kaur. "Recent advances in performance and effect of Zr doping with ZnO thin film sensor in ammonia vapour sensing." GLOBAL NEST JOURNAL 23, no. 4 (2021): 526-531. https://doi.org/10.30955/gnj.004020 , https://journal.gnest.org/publication/gnest_04020
30. N.S. Kalyan Chakravarthy, B. Karthikeyan, K. Alhaf Malik, D.Bujji Babbu,. K. Nithya S.Jafar Ali Ibrahim , Survey of Cooperative Routing Algorithms in Wireless Sensor Networks, Journal of Annals of the Romanian Society for Cell Biology ,5316-5320, 2021
31. Rajmohan, G, Chinnappan, CV, John William, AD, Chandrakrishan Balakrishnan, S, Anand Muthu, B, Manogaran, G. Revamping land coverage analysis using aerial satellite image mapping. Trans Emerging Tel Tech. 2021; 32:e3927. https://doi.org/10.1002/ett.3927
32. Vignesh, C.C., Sivaparthipan, C.B., Daniel, J.A. et al. Adjacent Node based Energetic Association Factor Routing Protocol in Wireless Sensor Networks. Wireless Pers Commun 119, 3255–3270 (2021). https://doi.org/10.1007/s11277-021-08397-0.
33. C Chandru Vignesh, S Karthik, Predicting the position of adjacent nodes with QoS in mobile ad hoc networks, Journal of Multimedia Tools and Applications, Springer US,Vol 79, 8445-8457,2020