International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 1|| January 2018 || PP. 67-74

Query Optimization of A Distributed Database System Using Fuzzy Logic

[*1]O. T. Timi-Johnsonand [2]P. O. Asagba.
Department of Computer ScienceUniversity of Port Harcourt
Rivers State, Nigeria
Corresponding Author: O. T. Timi-Johnsonand

Abstract: *Query optimization of a system is the process of modifying or enhancing that system, so that some aspect of the system works more efficiently and makes use of fewer resources. Generally, a query is optimized so that it can execute more rapidly, operate with less memory or resources. The objectives of this paper are to design a distributed system that optimizes queries and processors, and to develop effective query execution plans for mining distributed database with the use of fuzzy logic. The system was implemented by developing a suitable framework for query processing in a distributed system that is time dependent, and a fuzzy logic approach was employed based on linguistic terminology for flexible user query. The result obtained shows fuzzy queries will always take less response time as compared to normal SQL queries. As observed, this is due to the fact that query evaluation engine does not have to search all the records in a database, it searches the records within the fuzzy range value alone.*

Keywords*: Fuzzy logic, Query optimization, Distributed database*.

Date of Submission: 01-01-2018 Date of acceptance: 20-01-2018

I. INTRODUCTION

Optimization comes to play when the design of a system needs to be improved on. The basic objective of optimization is to either minimize the cost of production or maximize the efficiency of production. Query optimization of a system is the process of modifying or enhancing that system so that some aspect of the system works more efficiently and makes use of fewer resources. Query optimization is a difficult task in distributed environment because of numerous factors like data allocation, speed of communication channel, indexing, availability of memory size of data transmission. A distributed database system is a collection of multiple, logically interrelated databases distributed over a computer network.it makes use of two different technologies used for data processing which includes database systems and computer networks. The role of a query optimization is to produce query execution plan which represent an execution strategy of the query with minimum cost (Taylor, 2010). In this study fuzzy logic will be used to produce a query execution plan (QEP) because of its capacity to interpret natural language, it also demonstrates a level of vagueness on the request for queries from a database. The use of fuzzy logic in querying the system will accelerate query processing time because of the limitation used on the data that will be selected.

II. RELATED WORK

Raipukar and Bannote (2013), integrated a query processing system into a distributed employee management database. In their work, they applied the distributed concept of fragmentation and relational calculus on three database base schemas and integrated the conventional SQL technique in a query optimizer application. Ranges were defined in terms of minimum and maximum salary limits and were supplied by users during a query session. They discovered that the third partition or database gave the least execution time.Tiwari and Chande (2013), reviewed several evolutionary algorithms for query optimization in a distributed database management system with a particular emphasis on the number of joins and relations in a query. Their findings showed that the performance of distributed query optimization can be greatly facilitated if hybrids of the Ant Colony Optimizers are used.Hu and Zhang (2006), proposed a model using the XML standard for the execution of distributed queries in a web environment. Based on their model, the user's query is decomposed into sub-queries using schemas of remote databases specified in XML, and processed by the database management systems on the sites. The results of local processing are joined and delivered to the user's site in a way similar to XJoin (Urhan and Franklin, 2000). Seamless data sharing was achieved using XML standard and in-memory data structure. In order to allow for easy implementation and flexible adaptation XML services were incorporated to lunch local processing in the asynchronous mode.Sunita and Jadhav (2012), described a generalized framework for the optimization of distributed queries. They suggested the heuristics approach to

find the best query evaluation plan and semi-join strategy for reducing communication cost in a distributed database.

Combi et al (2011), proposed a general framework for the description and management of temporal trends by considering specific temporal features with respect to the chosen time granularity in a database. They studied the temporal aspects of clinical data within temporal relational databases, by using a temporal extension of the relational calculus, and subsequently by showing how to map these relational expressions to plain SQL queries. They studied inter-granule trends with a developed Trend Analyzer for on-line analytical processing and discovered the need to consider efficiency issues in the query process. However, little emphasis was placed on the optimization of the SQL queries.Sukheja and Singh (2013), developed a novel hybrid Query optimization model and algorithm for a distributed database. Their novel algorithm or model is a step ahead of previous techniques in the query optimization by employing recursive techniques in the query map using relational algebra calculus. To verify the feasibility of "hybrid query optimization algorithm" they developed simulation models in a dynamic environment and experiments were conducted in a distributed database environment using the functionality of database environment, multithreading, networking and JDBC concept.

Ryeng (2011), focused on improving performance of query processing in large DBMS where co-ordination between sites has been reduced in order to increase scalability. Methods to increase failure resilience of aggregation queries were used. The study of failure resilience of aggregation queries presented shows that different aggregation functions react differently to failures and measures must be put in place to counter each of the adopted function. These methods are listed below;

1. A low cost method to increase accuracy is proposed.
2. A dynamic data placement method.
3. The lock up method.
4. Catching of intermediate result.

These methods have been evaluated by doing simulations and experiments on implementation in a distributed DBMS prototype.Bezdek (2011), in his work dealt with fuzzy logic and its possible use in database systems. He illustrates fuzzy thinking style in a simple example, then emphasizes the advantage of fuzzy approach to database searching.

III. METHODOLOGY

The agile model is the proposed methodology used for this work because of its ability to accommodate changes at any stage in the development cycle.

3.1 Analysis of Existing System

The existing system was implemented by (Raipurkar,2013). In his work, he integrated a query language to increase the capability of data retrieval based on human perception. He also made emphasis on the difficulty/challenges encountered in query optimization of a distributed database system, and suggested that integration of a query processing subsystem into a distributed database management system can be used to analyze query response time across fragmentations of global relations.

3.2 Architecture of Existing System

The architecture of the existing system consists of a trusted client as well as two or more servers that provide a database service. Client will execute the query on various fragments and the algorithm will find out the best minimized query processing time. The database is partitioned into three partitions: dbfpartition1,dbfpartition2 and dbfpartition3. In the database design, dbfpartition1 consists of two vertically fragmented relations on the basis of tuple Id (tid) from which original global relation can be recovered by applying simple join algorithm. Dbfpartition2 consists of three vertically fragmented relations on the basis of tuple id and dbfpartition3 consists of original global relation. In the query processing and query optimization process, query response time is compared against these two vertically fragmented relations with global relation and fragments with minimum response time for read only and update application is selected for developing distributed database applications.The architecture of the existing system is shown in Figure 1.

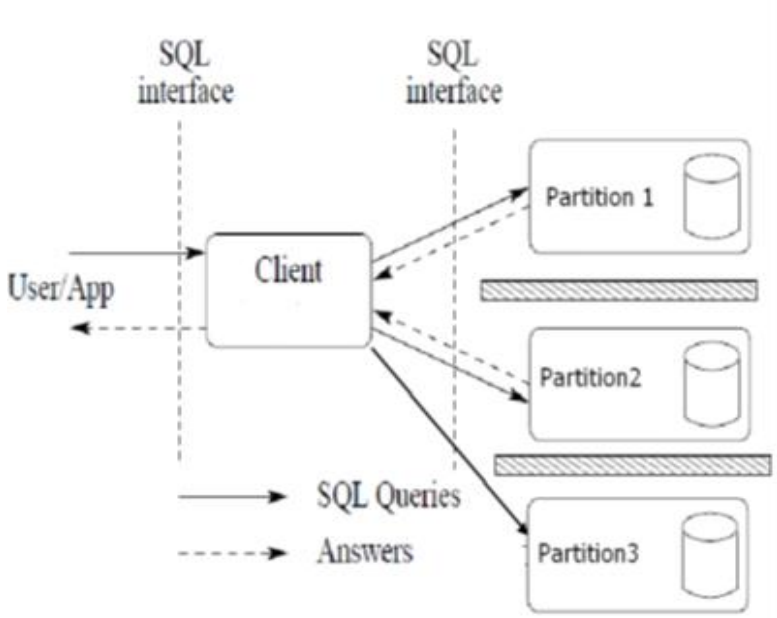

Figure. 1: Architecture of existing system (Raipurkar, 2013)

3.3 Disadvantages of Existing System

1. Microsoft visual studio 2010 was used to design the user interface, which makes the interface non-receptive to fuzzy entries.
2. The architecture of the existing system does not clearly state the application of fuzzy range in its design.
3. The system is rigid, because it focuses on only one attribute (salary) amongst all other represented in the user-interface.
4. The system's architecture does not have an explicit framework, in other words the framework was not clearly stated.

3.3 Implementation of Existing System

The existing system scheme was implemented using Microsoft Visual Studio 2010 and Microsoft SQL server 2010. The Working scheme is:

1. Create a different partition for the database (here three databases are created with different schemas).
2. Apply workload i.e. queries on all databases and estimate the cost of each query in terms of response time.
3. The number of queries applied on the partitions is varied from 1 to 100.
4. Different types of query operations such as Insert, Update, delete and fetch are applied on all the partitions.
5. A range is defined for the salary attributed defined in all the partitions
6. If user is fetching records of ranges then query is classified into range1, 2 or 3 as per the user input.
7. Then the comparison is done on the basis of cost of queries i.e. response time of the query.

For queries which we had simulated on salary attribute of the partitions, partition3 is the best candidate as it results in least response time. Ranges are defined in terms of minimum and maximum salary which will be entered by user in main window of project and according salary will be assigned proper ranges as defined in system. Simulating normal query on all partitions shows that queries will always take less response time. Reason for the least response time is that query evaluation engine does not have to search all the records in database; it has to search records in range values. Hence SQL queries are most optimized queries in distributed databases.

3.5 Design of Proposed System

In the proposed system, a fuzzy based model for optimal mining of a DDBMS will be developed. This will also present the underlying principles and analytical studies for a proper understanding of the fuzzy mining process. A step by step algorithm and Use Case Model will be provided to show how this model can be realized. An effective query execution plan for mining distributed database system will also be developed.

The systems component architecture of an Optimal DDBMS for a layered approach is captured in Figure3 and its shows the key component parts of the DDBMS.

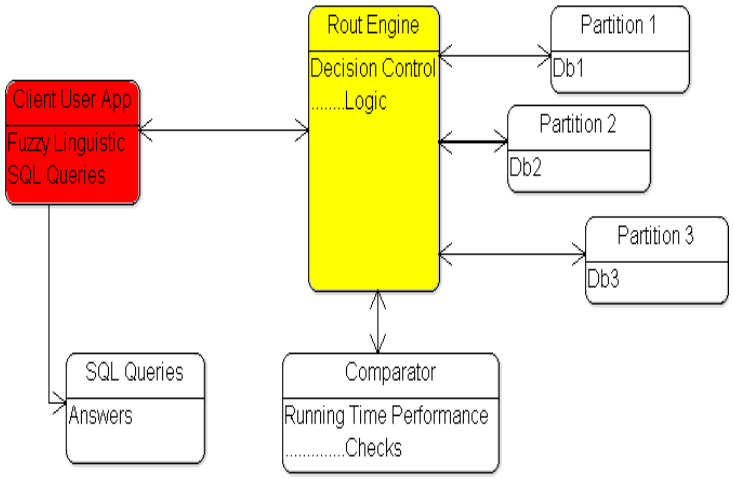

Figure.2: Proposed system Architecture

3.5.1 Advantages of Proposed System

Fuzzy logic which is one of the most successful artificial intelligence technique is been used in the implementation of the proposed system.

1. It enables the system to be easily understood because of its use of natural language.
2. The proposed system applies distributed database ideology to a DBMS and implements efficient tunneling using a routing control logic technique, this is advantageous as this can reduce query over load in data mining a system and encourage a more efficient way of transferring client queries in a time dependent manner. It also prevents single point failure from occurring.
3. The system also encourages end-user activity that is it has a friendly user interface.
4. The proposed system is implemented using java language which is an object oriented language, thereby making the codes reusable.

3.5.2 Use Case Analysis of Proposed System

The use case model typically includes the follow key actors:

1. The database user
2. The database engineer

Typical use cases in a Use Case Model for a DDBMS will include:

1. Database Updates
2. Optimal Processor
3. Maintain Database
4. Database Site Considerations
5. The Use Case Model for the test DDBMS should be implemented as shown in Figure3. It can be seen that car finder is separate from the Database user and Expert because a find operation does not require update or maintenance functionality.
6.

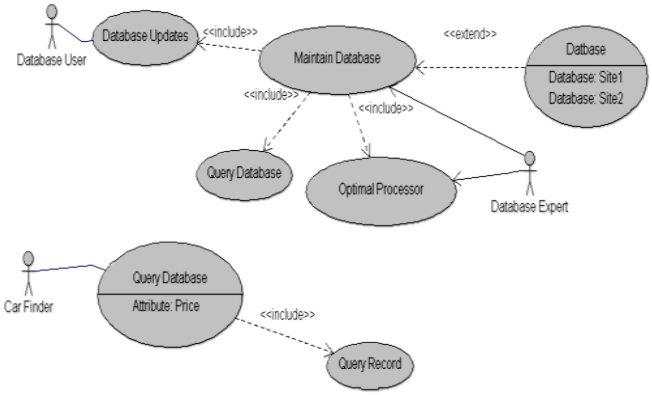

Figure 3: Use Case Model Capture

3.5.3 Systems Structure and Domain of Proposed System

The systems block structure for a DDBMS is captured in Figure 4. It shows the Information System (IS) which comprises of people and computers that process and interprets information. JDBC is a Java database connectivity technology.This technology is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. The query processor in a database management system receives as input a query request in the form of SQL text, parses it, generates an execution plan, and completes the processing by executing the plan and returning the results to the client.

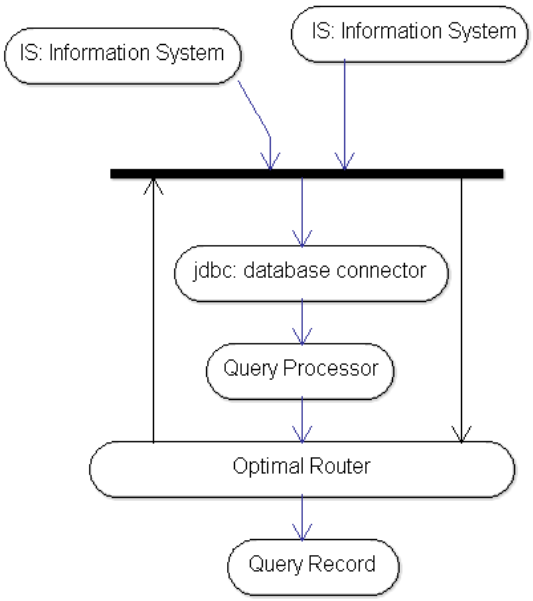

Figure 4: Fuzzy DDBMS Systems Model

IV. Database Design

The database is designed using MS-Access.

The design considers the following structures specifications to minimize risks in the implementation:

1. Six Attribute field types – ID, Price, Year, Km
2. Maximum field length of 50
3. Text Field (String) Data Types except for ID which is Auto-number (Long Integer)
4. Use of Duplicate keying strategy for other field types excluding id.

4.1 Input Specification

Input specifications include specifications that govern the operation and reporting mechanism of the DBMS. Some of the key are;

1. We expect numeric output with upper and lower bounds for each attribute considered:
   a. Price range: 5000 – 1, 000, 000
   b. Year: 2003 – 2009
   c. Km: 0 – 100, 000
2. Maximum number of input attributes – 3 for Price, Year and Km
3. Fuzzy Input Expressions– Lingual (low, moderate, high). The input specification of the proposed system is shown in Table 1.

Table 1: Input Specification

ID	ATTRIBUTE		LINGUISTIC TERM		
			LOW	MEDIUM	HIGH
1	PRICE	TERM	Low price	Medium Price	High Price
		RANGE			
2	KM	TERM	Low km	Medium km	High Km
		RANGE			

3	YEARS	TERM RANGE	Old years	Middle years	Current years

4.2 Output Specification

The output specification of the proposed system is shown in Table.2

Table 2: Output Specification

ID	Query-Time	Partition	Database
1	<75ms	Partition 1	Car Db1
2	>75ms	Partition 2 Partition 3	Car Db2 Car Db3

4.3 Analysis Of Result

The system has been simulated based on input specifications and the results of simulation are as shown in Figure 4.1 and 4.2 for the price, km and years attributes, more than one attribute can be queried at a time as compared to the existing system where only one attribute was considered.Next, we test our system for routing functionality using our time-dependency algorithm. This is captured in Figure 4.3 using the java console in NetBeans. As can be seen the system seeks to query a smaller database when query time is large. From the results, it is clear that comparison of simulating normal SQL query as carried out in the existing system and fuzzy query on all partitions as executed in the proposed system, shows that fuzzy queries will always take less response time as compared to normal SQL query. Reason for the least response time is that query evaluation engine does not have to search all the records in database or partition; it has to search records in fuzzy range values alone. Hence fuzzy queries are most optimized in distributed databases.Figure 4.1 shows the fuzzy query outputs for attributes of price/kilometer. Frm the display we see that the system gives a specified range to suit the user's request. Thereby eliminating cumbersome search for users and hence fast decision making.Figure 4.2 shows the fuzzy query outputs for attributes of kilometer and year. From the display we see that the system gives a specified range to suit the user's request. Thereby eliminating cumbersome search for users and hence fast decision making. We can also see that more than one attribute can be queried at a time, unlike the existing system which focuses on only one attribute.

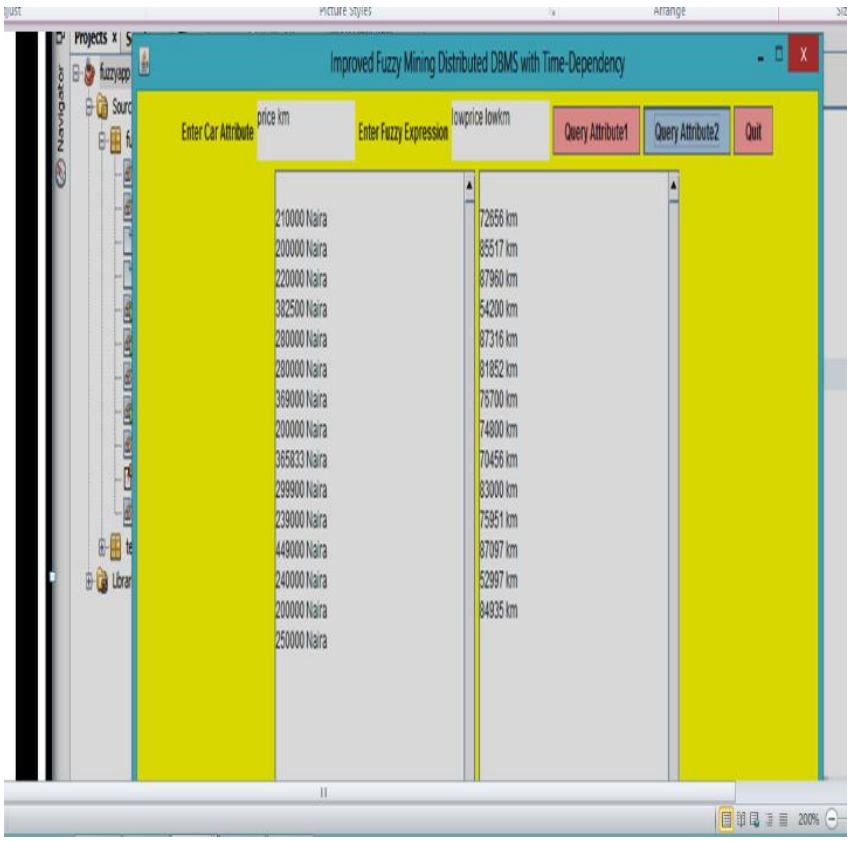

Figure 4.1: Fuzzy query output for price and km

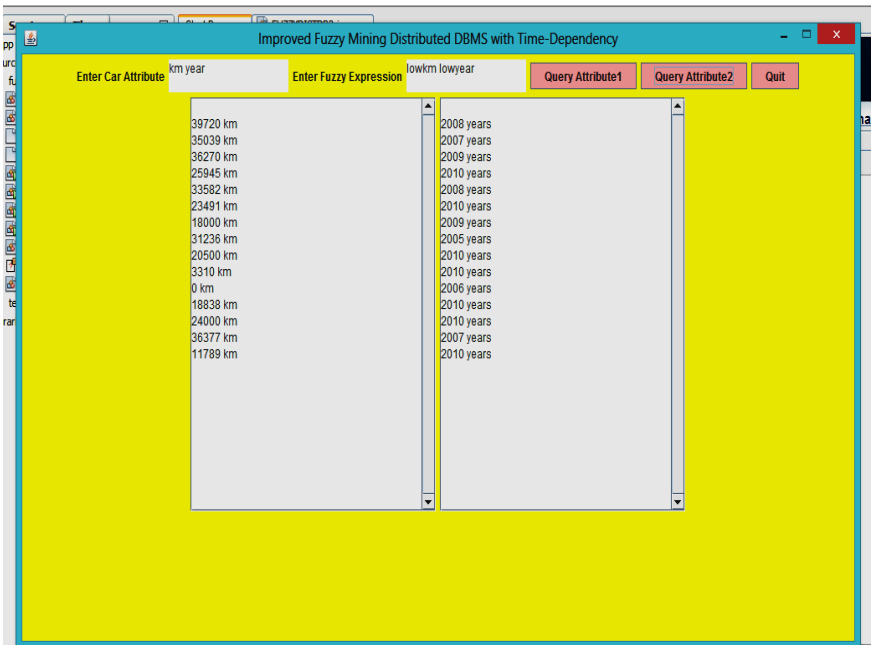

Figure 4.2: Fuzzy query output for kilometer and year

V. Conclusion And Recommendation

In This Study, We Have Developed A Suitable Framework For Query Processing In A Distributed Database Management System In A Time Dependent Way. We Have Also Employed A Fuzzy Logic Approach Based On A Linguistic Terminology For Flexible User Query For More Than One Attribute Case Using The Java Language. In Addition, A User Interface Has Been Developed To Make System Easier To Use. Generally, Querying A Database Involves Sql-Like Statements. We Have Succeeded In Integrating Sql In Our Fuzzy Approach Using Java Class And This Has Greatly Improved The Querying Process.1.

REFERENCES

[1]. Bennett, K., Ferris, M. C. And Ioannidis, Y. E. (2000). A Genetic Algorithm For Database Query Optimization. In Proceedings Of The Fourth International Conference On Genetic Algorithms, 400–407.
[2]. Coccombi, C., Pozzi, G. And Rossato, R. (2012). Querying Temporal Clinical Databases On Granular Trends, Journal Of Biomedical Informatics, Elsevier, 273-291.
[3]. Dubois, D. And Prade, H. (1990). Fuzzy Sets And System: Theory And Application ,Academic Press, New York.
[4]. Hu, G. And Zhang, D. (2006). Execution Of Distributed Queries In Web Environment, Journal Of Computational Methods In Science And Engineering, Ios Press, 243-254.
[5]. Mamdani, E.H. And Assillian, S. (2000). An Experiment In Linguistic Synthesis With A Fuzzy Logic Controller. International Journal Of Man Machine Studies, Vol.7, No.1,1-13.
[6]. Mamdani, E.H. (1990). Advances In The Linguistic Synthesis Of Fuzzy Controllers. International Journal Of Man-Machine Studies, Vol.8, 669-678.
[7]. Perrizo, W. Li, J. Y. And Hoffman, W. (1989). "Algorithms For Distributed Query Processing In Broadcasting Local Area Networks," Ieee Trans. Knowledge And Data Eng., Vol. 1:2, 215-225.
[8]. Rao, S.S. (2009). Engineering Optimization Theory And Practice, John Wiley &Sons, Inc.4th Edition, 693-722.
[9]. Raipukar, A.R. And Bamnote, G.R. (2013). Query Optimization In Distributed Database, International Journal Of Innovation Research In Computer And Communication Engineering, Issn (Print): 2320-9798, Vol.1, Issue 2, 422-426.
[10]. Sukheja, D. And Singh, U.K. (2013). Novel Distributed Query Optimization Model And Hybrid Query Optimization Algorithm Vol.8 22-32.
[11]. Sunita, M. And Jadhav, V. (2012). General Frame Work For Optimization Of Distributed Queries, International Journal Of Database Management System; Vol.4, No.3,35-47.
[12]. Schweitzer, B. And Skylar, A. (2001). Associative Functions And Abstract Semi-Groups, Publ. Math Debrecen, Vol.10, 69-81.
[13]. Selinger, P. G., Astrakhan, M. M., Chamberlin, D. D., And Price, T. G. (1999) Access Path Selection In A Relational Database Management System. In Acm Sigmod International Conference On Management Of Data., 23–34.
[14]. Sugeno, M. (2003). Fuzzy Measures And Fuzzy Integrals. Fuzzy Automata And Decision Processes,89-102, North-Holland, Ny.
[15]. Swami, A. And Iyer, B. (1993). A Polynomial time algorithm for optimizing join queries. In proceedings of the 9th international conference on data engineering (vienna, austria, apr.), ieee Computer Society, Washington, DC, 345–354.
[16]. Tiwari, P. And Chande, S.V. (2013).Query Optimization Strategies In Distributed Databases, International Journal Of Advances In Engineering Sciences, (Print) Issn: 2231-0347, Vol. Issue 3, 23-29.
[17]. Urhan, T. And Franklin, M.J. (2000). X Join: A Reactively Scheduled Pipelined Join Operator, Ieee Data Engineering Bulletin Vol.5, 27-33.
[18]. Yao, Y.Y. (2012). Granular Computing For The Design Of Information Retrieval Support Systems, Kluwer Academic,299-329.
[19]. Zadeh, L.A. (1997). Towards A Theory Of Fuzzy Information Granulation And Its Centrality In Human Reasoning And Fuzzy Logic, Fuzzy Sets And Systems.Vol.19, 111-127.

Zadeh, L.A. (1995). The Concept Of A Linguistic Variable And Its Application To Approximate Reasoning, Parts 1,2 And 3.Information Sciences,Vol.8, 199- 210.

[20]. Zhang, W. And Wang, K. (2000). An Efficient Evaluation Of A Fuzzy Equi-Join Using Fuzzy Equality Indicators. Knowledge And Data Engineering, Ieee Transactions On, Volume: 12 Issue: 2, 225 –237.

O. T. Timi-Johnsonand."Query Optimization of A Distributed Database System Using Fuzzy Logic." International Journal of Engineering Science Invention(IJESI), vol. 7, no. 01, 2018, pp. 67-74.