

Query Optimization Using Fuzzy Logic in Integrated Database

Susana*, Suharjito

Magister in Information Technology, Binus Graduate Program, Bina Nusantara University,
Jakarta, Indonesia

*Corresponding author, e-mail: elf.shinee@gmail.com

Abstract

Query optimization in integrated database can't be separated from data processing method. In order to have faster query response time, a method to optimize queries is required. One of many methods that can be used for query optimization is using fuzzy logic with Tsukamoto inference system. Value set on each variable is defined membership functions and Tsukamoto inference system used in determining these rules or the terms of query results, then apply it into query method or query line structure. The application of fuzzy logic inference systems with Tsukamoto can accelerate query response time, and will have more significant difference when the amount of selected data is greater.

Keywords: query optimization, fuzzy logic, database

Copyright © 2016 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Technology and data is becoming a very important part in the survival of a company's business. As the company grows, the greater the (amount of) data that must be processed. Such data must be analyzed to be used to support business decision making. Analysis for any type of decisions must also be done by considering different factors.

For certain types of transactions made by specific customers, negotiation must be done with the grantor, which slows down the transaction process, to accelerate this process; a categorical list of consumers is made. With the list of categories of consumers, trading decisions can be directly carried out by the system. But the list of categories of consumers obtained from querying the current data provide considerable comparative results with a list of categories created by the grantor's decision. This means that the decision must still be made manually, because the results from the system cannot be used to make decisions. The significant difference between the results of the list of categories of consumers of the system and a list of categories of consumers of the decision maker occurs because the system makes decisions based only on the exact figure (firmly) from the factors used as reference, while decision maker (human) makes a decision based on estimates mainly on factors that are considered more important. Therefore, it needs to be optimized in the query that is used to obtain the list of categories of consumers.

Use factors that affect query response time, among others are: missing indexes, inexact statistics, badly written queries, deadlocks, cursors, excessive fragmentation of indexes, and frequent recompilation of queries [1]. Optimization of query response time, especially on a large database with the number of joins that a lot can be done by using a genetic algorithm to obtain the best selection methods and the fitness function used in processing and the process of chromosomal mutations will shorten the response time [2].

In this study, query optimization will be carried out by using fuzzy logic. The use (of) query with fuzzy logic will accelerate query processing time because of the limitations use on the data that will be selected [3]. The use of fuzzy logic can not only improve the response time, but can also improve the selection results data obtained primarily related to the fuzzy logic so as to produce a system that is flexible and can handle general human demand [4]. The use of fuzzy logic will be a solution to problems in this study because fuzzy logic provides results that are can also use uncertain or it can be said to be approaching the human mind [5].

2. Related Works

Siallagan utilizes genetic algorithms to optimize the database query by using the virtual tables represented in the query tree to generate new parse tree used left-deep strategy with two methods of crossover, namely: M2S crossover and crossover Chunk [6]. Raipukar perform optimization in distributed databases by using fuzzy logic role. Queries on distributed data are expressed in the form of relational calculus relations resulting fuzzy queries [3]. Tiwari optimizes queries in a distributed database with some evolutionary algorithms such as Ant Colony Optimization (ACO), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO). Each optimization results obtained are then compared, in order to obtain that ACO provides the most optimum results [7]. Syaifudin implement online analytical Hybrid processing (HOLAP) for a distributed database query optimization by using genetic algorithms on database clustering process [8]. Barbhuiya create architectural models of optimal query using a genetic algorithm with genetic programming approach to optimize complex queries [9].

3. Research Method

The first step of this research is literature study, collect data into the test sample, analyze the condition of the current database, determine variables, create membership function, and determine the rules, process optimization, evaluation of optimization results and then conclusions and suggestions. The research will begin by describing the background, and then determine the purpose and scope of this study. The study of literature becomes necessary to deepen the understanding of the optimization of fuzzy membership functions, fuzzy logic and Fuzzy Inference System Tsukamoto.

Second step, collect the data needed to conduct trials, and ensure the company is granted permission to use the data.

Third step, analyze the condition of the existing database. The next is determined variables into a factor in query optimization, making the membership function in each set at a predetermined variable, then define the rules in query optimization. After that, using the fuzzification membership functions that have been made and to do query optimization by inference using Fuzzy Inference System Tsukamoto. From the results of inference, defuzzification will be conducted to get the result in crisp set.

Fourth step, query optimization results will be evaluated to see a comparison of the query before and after optimization.

In the process of fuzzification Fuzzy Inference System Tsukamoto, use the following steps:

- 1) Fuzzification the crisp data. Fuzzification carried out in accordance with the membership function of each variable predetermined.
- 2) Results fuzzification inference will be conducted by the rules (rules) that have been determined.
- 3) After the process of inference, the next step is to do defuzzification of the results of the inference obtained, in order to obtain a query that will produce data category of consumers which is the output variable in this study.

4. Results and Analysis

4.1. Comparison before and after Optimization

The test results are made will be compared to determine the difference between the response time before optimization with fuzzy logic and the response time after optimization with fuzzy logic. Comparison of average response time required for query can be seen in the Figure 1. Based on Figure1 it can be seen that the response time after the optimization with fuzzy logic is faster than the response time prior to optimization. This is because with the use of fuzzy logic in modeling rules then there is a limitation of data query region so that the query is run on a more specific area. In addition, with the use of valuable linguistic variables, the search query is easier because the query only search the data by comparing the word on every decision variables. Response time difference obvious when doing quite a lot of data selection. The lesser amount of data to be selected, the fewer differences in response time.

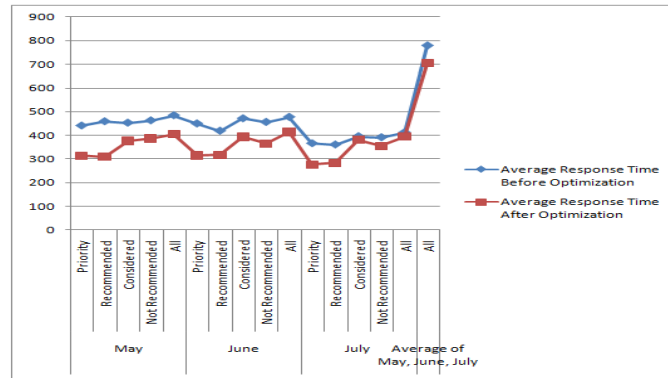


Figure 1. Comparison of Before and After Optimization

Furthermore, the T-Test determines whether there are significant differences between the response time before optimization and response time after optimization. Here are the results Paired T-Test before and after optimization with fuzzy logic:

T-Test

[DataSet0]

	Mean	N	Std. Deviation	Std. Error Mean
Pair 1 Sebelum_Optimasi	454.9000	16	94.45244	23.61311
Setelah_Optimasi	375.0250	16	98.40292	24.60073

	N	Correlation	Sig.
Pair 1 Sebelum_Optimasi & Setelah_Optimasi	16	.925	.000

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
				Lower	Upper			
Pair 1 Sebelum_Optimasi - Setelah_Optimasi	79.87500	37.46293	9.36573	59.91241	99.83759	8.528	15	.000

Figure 2. Paired T-Test between Response Time before Optimization and After Optimization

In testing using Paired T-Test between response time before optimization and response time after optimization, with a significance level of 5%, the value of the t-table obtained amounted to 2.1315 and t value by 8.528. $T_{count} > t_{table}$, it shows that there is a significant difference between the response time before and after optimization with fuzzy logic.

4.2. Comparison of Optimization with Fuzzy Logic and Genetic Algorithms

Comparisons are also made between the query optimization using fuzzy logic with optimizations performed by using a genetic algorithm. Comparison of average response time between optimization with genetic algorithms and fuzzy logic can be seen in the Figure 2. Figure 3 shows that the response time comparison of fuzzy logic and genetic algorithms is approximately 0.763 to 1.065. The amount of data that is selected does not significantly affect the query that is optimized with genetic algorithm, but the results of optimization with genetic algorithm similar to results obtained from queries prior to optimization, while the number of data selected on queries that are optimized with fuzzy logic is more optimal for close to the results determined manually. Table 1 compares of the number of records that are selected manually (by humans), before the query optimization, query-optimized fuzzy logic and queries are optimized with genetic algorithm.

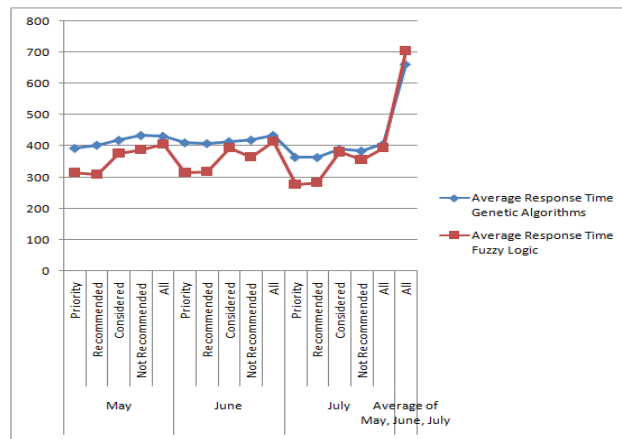


Figure 3. Comparison of Optimization with Fuzzy Logic and Genetic Algorithm

Furthermore, the Paired T-Test to determine whether there are significant differences between the response time optimized with genetic algorithm and response time optimized with fuzzy logic (Figure 4). Here are the results Paired T-Test query with genetic algorithms and fuzzy logic:

T-Test

[DataSet1]

	Mean	N	Std. Deviation	Std. Error Mean
Pair 1 - Algoritma_Genetika	420.5500	16	67.93191	16.98298
Fuzzy_Logica	375.0250	16	98.40292	24.60073

	N	Correlation	Sig.
Pair 1 - Algoritma_Genetika & Fuzzy_Logica	16	.954	.000

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
				Lower	Upper			
Pair 1 - Algoritma_Genetika - Fuzzy_Logica	45.52500	39.21264	9.80316	24.63006	66.41994	4.644	15	.000

Figure 4. Paired T-Test between Response time with Genetic Algorithm and Fuzzy Logic

In testing using Paired T-Test between the response time of queries that are optimized with genetic algorithm and the response time of queries that are optimized with fuzzy logic, with the same test that the significance level of 5%, the value t-table obtained amounted to 2.1315 and the value of t amounted to 4.644. $T_{count} > t_{table}$, it shows that there are significant differences between the response time before and after optimization with fuzzy logic.

The next evaluation is to compare the selection results obtained from the man-made selection, before the query optimization, query after optimization with fuzzy logic and query with genetic algorithms. The following table is a comparison of the number of records that are selected manually (by humans), before the query optimization, query-optimized fuzzy logic and queries are optimized with genetic algorithm.

Table 1 show that the amount of data the result of a query that is optimized using a fuzzy logic approach results manually determined by a ratio of 0.706 to 1.136. This is because the results of fuzzy logic approach using human logic membership functions and rules given.

The next step is to do Precision, Recall and Accuracy Test to determine the level of accuracy of the information requested using the answers given by the system (precision), the success rate in finding information back (recall), as well as the degree of closeness between the predicted value compare with the actual value (accuracy). The data that will be used to

determine the accuracy would be priority consumer data from each month. The formula used for calculating the value of precision and recall can be described as follows Table 1.

Table 1. Comparison of the Number of Records

Month	Consumer Data	Number of Records			
		Manual	Before Optimization	Fuzzy Logic Optimization	Genetic Algorithms Optimization
May	Priority	19	4	17	4
	Rekomended	81	38	73	38
	Considered	1127	909	1149	909
June	Not Recommended	885	1161	873	1161
	Priority	17	7	12	7
	Rekomended	109	34	94	34
July	Considered	1603	1224	1531	1224
	Not Recommended	691	1155	783	1155
	Priority	11	4	8	4
	Rekomended	63	29	50	29
	Considered	1488	1525	1690	1525
	Not Recommended	1104	1108	918	1108

Table 2. Confusion Matrix Precision, Recall and Accuracy Test

		Human (Manually)	
		Not a Priority	Priority
Fuzzy Logic	Not a Priority	TN (<i>True Negative</i>) <i>Correct Absence of Result</i>	FN (<i>False Negative</i>) <i>Missing Result</i>
	Priority	FP (<i>False Positive</i>) <i>Unexpected Result</i>	TP (<i>True Positive</i>) <i>Correct Result</i>

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision, recall and accuracy test will be conducted on data obtained from the query before optimization and after optimization.

Table 3. Comparison of the Average Percentage of Precision, Recall and Accuracy Test

	Before Optimization with Fuzzy Logic	After Optimization with Fuzzy Logic
Precision	100%	100%
Recall	32.86%	77.60%
Accuracy	99.54%	99.86%

Table 3 shows the significant difference lies in the degree of success in finding the back of information (recall). Query that is optimized using fuzzy logic provides a higher yield compared to the prior query optimization, while the results of precision and accuracy tests did not have significant difference.

This test shows that the query with fuzzy logic can be used on the system to accelerate the processes in the system to provide accurate results.

5. Conclusion

In this study, it appears that the query optimization with fuzzy logic can improve the speed of query response time. The amount of data selected still affect the speed of query response time, but the time of selection of the data with the same number of records, it appears that query with fuzzy logic faster than the query without fuzzy logic. A significant difference occurred while processing the query for selecting data in a large number.

The number and resulting data's are selected from the query using fuzzy logic is closer to the result of selection by humans with an accuracy rate of over 90%. Thus the results of

query optimizations are useable decision making without going through the process of analysis so as to shorten the processing time.

References

- [1] Corlatan CG, Lazar MM, Luca V, Petricica OT. Query Optimization Techniques in Microsoft SQL Server. *Database System Journal*. 2014; 5(2): 33-48.
- [2] Butey PK, Meshram S, Sonolikar RL. Query Optimization by Genetic Algorithm. *Journal of Information Technology and Engineering*. 2012; 3(1).
- [3] Raipurkar AR, Bamnote GR. Fuzzy Logic Based Query Optimization in Distributed Database. *International Journal of Innovatibe Research in Computer and Communication Engineering*. 2013; 1(2): 422-426.
- [4] Sharma P. Retrieval of Information Using Fuzzy Queries. *International Journal of Engineering and Techniques*. 2016; 2(3).
- [5] Mohiuddin MS. Performance comparison of conventional controller with Fuzzy logic controller using chopper circuit and Fuzzy tuned PID controller. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*. 2014; 2(4): 149-164.
- [6] Siallagan M, Sabariah MK, Sontya M. *Optimasi Query Database Menggunakan Algoritma Genetik*. In Seminar Nasional Aplikasi Teknologi Informasi. 2008: 53-57.
- [7] Tiwari MP, Chande SV. Query Optimization Strategies in Distributed Databases. *International Journal of Advanced in Engineering Sciences*. 2013; 3: 23-29.
- [8] Syaifudin R, S, Hartanto R. Review: Implementasi Holap Untuk Optimasi Query Sistem Basis Data Terdistribusi Dengan Pendekatan Algoritma Genetik. *Jurnal Sistem Informasi Bisnis*. 03. 2014: 164-171.
- [9] Barbhuiya SZ, Nandi MG. Query Optimization Using a Genetic Programming Approach. *ADBU-Journal of Engineering Technology*. 2016: 158-163.