Efficient Database Risk Management Using BSP and Fuzzy Logic

Pranav D. Talwar
Department of Computer Engineering
Smt.Kashibai Navale College of Engineering
Pune, India
pdtalwar@gmail.com

Prof.D.H.Kulkarni
Department of Computer Engineering
Smt.Kashibai Navale College of Engineering
Pune,India
deepakulkarniskn@gmail.com

***Abstract*—** Increasing passion for internet throughout the world causes the exponential growth in the web applications and distributed techniques. Due to high usage of web applications massive transactions are happening at the database server side. Even though databases are well equipped with powerful tools, most of the times they are unable to fulfill the user's demand and resulting in longer waiting queues or crashing of databases. So many methods and systems are existing to handle the overflowing database queries, but most of them again take longer time to get rid of the situation. This paper put forwards an idea of handling this risk situation of the database by collecting all the queries in a Queue and thereby evaluating risk aware situation by fuzzy classification. Once the risk awareness is notified then these queries in the queue are committing quickly using batch stream processing technique to avoid longer waiting queues of the queries for execution.

***Keywords**- Batch stream processing; Fuzzy logic; Risk awareness; load*

*****

I. INTRODUCTION

In today's era of globalization the number of businesses is increasing tremendously. For any company, the secret data of company plays an important role. No company wants to leak their secret data as it can bring the company on road. Since growing rate of industries gaining attention of the size of data.

With the increase in data the overhead of maintaining such data is also increases. A recent study shows that most of the companies are not interested in putting lots of money on IT infrastructure. So to have feasible maintenance of data, cloud emerged as one of the best suit. In short words cloud is a storage pool to which remote access can be done. As the cloud offers a great advantage in maintaining and accessing the data it becomes as one of the area of research. Along with maintaining the confidentiality of the data, it reduces the hardware cost to the great extent.

Numbers of databases are used by the companies to put their secret data. So in order to keep the confidentiality database should pass some of the privacy policies. One of such policy is access control mechanism. Access control restricts the unauthorized users to access the data by giving some secret credentials to the authorized users. To do so RABC (Role base access control) is a famous policy.

Query optimization is a technique of increasing efficiency of the query so that processing time taken by the query should be less. Query optimization is important but not must part of the query processing systems. Varieties of algorithms are proposed to perform query optimization. Query optimization consists of 4 main steps as.

- Scanning and parsing
- Optimization
- Query Code Generator
- Query processor

Parsing

Whenever query is submitted it is very important part to check whether it consists of important syntax or not. Scanning is a process of converting plain text to the text in form having well syntax. Once well formatted text is generated it checked for its correct syntax. This syntax checking is done by the parsers. On successful parsing the query fed to the next optimization module.

Optimization

For any given query different ways are available to shoot the query. E.g. SELECT, JOIN operations can be completed by different other ways. Query optimization is a process of finding the best suitable way to execute query with less time overhead. E.g. there are some queries that can be executed in sequential and parallel manner also. So if the query optimization is done using parallel manner then the required time is less compared to the sequential manner. For this optimization set of rules are applied on the internal structure of the query. Query optimization can be done by using two ways

1. Heuristic optimization
2. Cost based optimization

Heuristic optimization makes use of some of set of rules to accomplish the task whereas cost based optimization makes use of different execution ways to find out the way that gives the minimum execution cost.

Query Code Generator

Query optimization find out the best execution plan to be executed by the query. Once this done, query code generator try to find the access routines to be followed by the query. DBMS giving a provision of saving the access routine for later use.

Query processor

Here actual execution of the query is done by using the access routine selected by the query code generator step as shown in the figure 1.

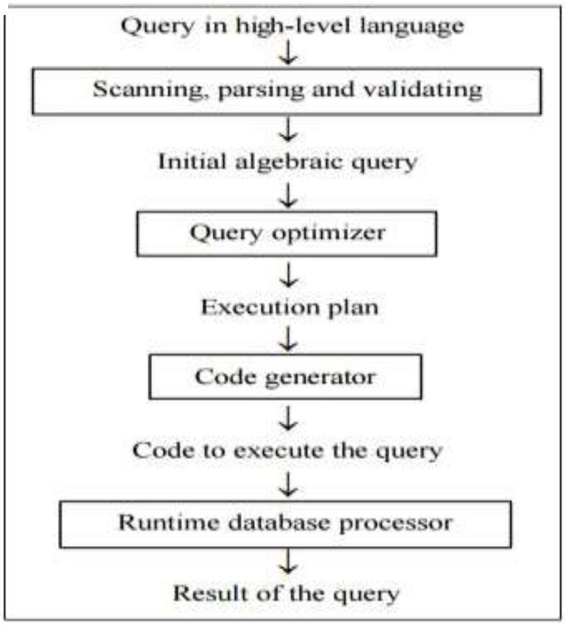
Query in high-level language
Scanning, parsing and validating
Initial algebraic query
Query optimizer
Execution plan
Code generator
Code to execute the query
Runtime database processor
Result of the query

Figure 1: Query processing process

Batch stream processing is a technique of executing the tasks without manual interruptions. Task is set up at start; system will completes the task as per the scenario and load with no involvement of the user. Batch stream processing offers list of advantages, some of these advantages are as:

- It shifts the task once task is done.
- It does not require human interruption
- It increases the efficiency as system doesn't have to wait much.
- Due to these advantages BSP used in variety of applications like databases, image processing, conversion operations etc. where time is at core part of the operation

The rest of the paper is organized as follows. Section 2 discusses some related work and section 3 presents the design of our approach. The details of the results and some discussions we have conducted on this approach are presented in section 4 as Results and Discussions. Sections 5 provide hints of some extension of our approach as future work and conclusion.

II. LITERATURE SURVEY

[1] Shows how the query optimization is performed on non-aggregate sub queries. Authors states that till there research numbers of methods are proposed to perform query optimization on aggregate sub queries but not on non-aggregate sub queries. They proposed a Nested relation approach taking nested relational algebra as base .Here optimization rules are also explored by the developers that can be used for the purpose of issue solving, which is normally observed in relational database management system.

[2] Proposed a Kairos a system for finding the hardware requirement as well as finding the resource utilization. Kairos is designed in such way that first of all it finds out the work load of the system and based on that workload it finds out the complete resource utilization. Also the problem of database consolidation is addressed. To find out the performance evaluation of the system they make use of TPC-C like benchmark. On this benchmark they shows that system obtains 12 times higher throughput. Also the real word dataset is collected from the wikia.com to check the efficiency of the system in real scenarios. Often databases are misused by the third parties. This misused cost a lot for the any business.

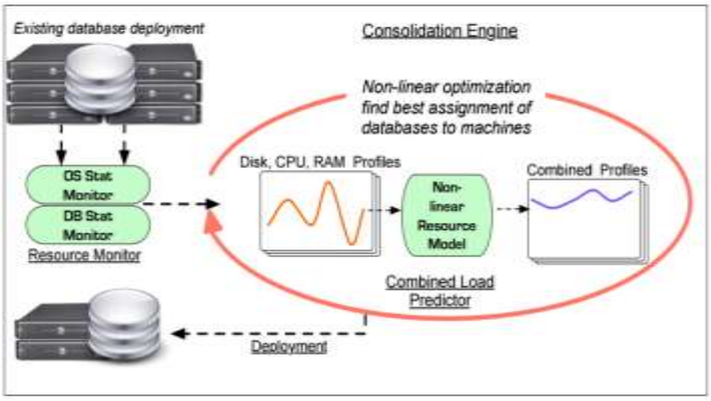

Figure 2: Kairos system architecture

[3] Proposed a DEMID a database misuse detection system which efficiently identify the intruders which misuse the databases. DEMIDS mainly focused on relational database management systems. In order to find out the misuse, DEMIDS makes use of audit logs which contains the behavior of the user actions. Domain knowledge of the database is used for the same.

[4] Tries to find out the normal query optimization problems observed in databases. To address the problem three factors are considered i.e. space used for searching, strategy used for searching and the model used to find out the cost estimation of the system. In this paper different search strategies are studied with their individual searching algorithms. For better understanding of the method, they did survey in two categories i.e. deterministic strategy and randomized strategy. From this deep study author concludes that randomized algorithms are not well suited for small queries but best suit for the large queries.

[5] Gives a query optimization technique in two simple ways. In first step authors just simple identifies the different optimization algorithms with deep study. And in second step an operational model is proposed to optimize the query. At the end they conclude that the proposed method can be effectively used for the dynamic demands of the user.

Sql queries can be effectively used to retrieve the data from the huge database. But when the input provider is human being the chances of getting more precise data will get reduced. To overcome the problem number of researches are proposed. Fuzzy database is one of such research. Fuzzy database makes use of fuzzy logic; fuzzy logic helps in finding the missing attributes from the query entered by the user. Also it can be used very effectively for maintaining the access control policies of the database.

[6] Presents MNS, a fuzzy based controller used to maintain the access control policies of the database. MNS is

developed to operate in centralized database system. In this paper the risk of the system is managed by finding the need of the user. However the fuzzy logic based systems can be effectively used for the distributed system. [7] explains the fuzzy logic based approach for distributed databases. Author tells that the system obtains good degree of confidentially.

[8] DStream processing is a batch stream processing, intended to overcome the disadvantage of the stream processing system. DStream refers to discrete stream where tasks are divided into small streams for the processing. Compared to traditional methods of stream processing, the said system requires less time and less fault tolerance ratio. Lineage information of each of the separate process is used to find out the re-computation in case of faults. Because of this reason faults tolerance ratio can be easily reduced without duplication of the nodes. For optimization purpose SQL is used as a base.

[9] Proposed a concept of synthetically created calibrating database. For the experimental evaluation of the system they make use of three traditional databases viz: Allbase, DB2 and Informix which gives 80% of accuracy. [10] Illustrates all the open issue of query optimization in relational database management system. They identify the three open problems in said field for the purpose of research.

III. PROPOSED METHODOLOGY

In this section, we describe our framework for efficient database management using fuzzy logic and batch stream prcessing with the below mentioned steps as shown below

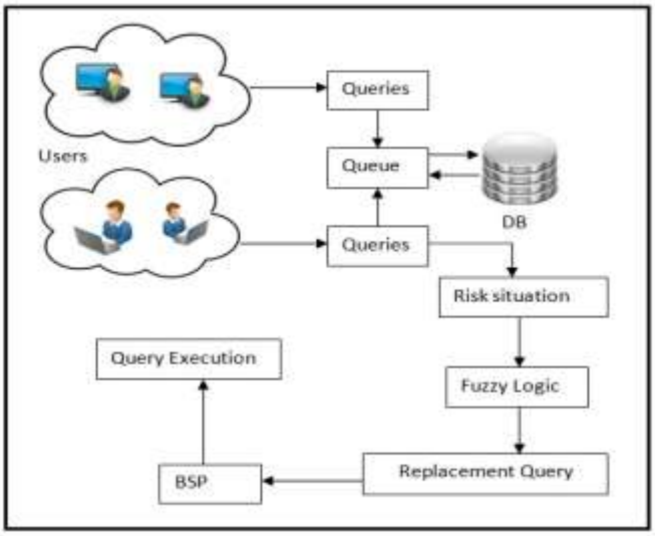

Figure 3: Overview of proposed approach

Step 1: Here in this step system is designed to accept huge number of database queries in a given instance which is stored Microsoft excels sheets. Once these queries are fed to the system, System receives all the queries in a processing queue and another parallel instance of the program removes these queries from the processing queue and sends for the execution to make the system continues busy for processing of the database queries.

Step 2: In this step more queries are fed to the busy system and again system receives all the queries and add into the processing queue.

Step 3:- Here once the system find itself in longer queue of the committing database queries, then it sense the risk situation awareness scenario and calls the fuzzy classification system to evaluate the situation in more preciseness.

Step 4: In this step performance parameter of the processing queue which is handling the query flow to the MYSQL database is calculated using the fuzzy classification.
Fuzzy logic works on the basis of following steps

- *Fuzzyfier Inference Engine*- Fuzzy system prepares the crisp values based on the number of the queries remaining in the processing queue and estimation time to complete the process. After evaluation of the time and number of queries crisp values are generated as very Low, Low, Medium, High and Very High ranges.
- *Fuzzification* – Here on the basis of crisp values the risk awareness is identified based on the predefined protocol for the fuzzy logic.
- *If then Rules* – Here all the essential parameters are evaluated by taking appropriate decision using if then style. Then risk awareness is quite well judged to prepare the query replacement process.

Step 5: Here once the risk is identified on the basis of database queue manager performance then immediately query replacement process triggers to collect all the remaining queries in the processing queue in a vector.

Step 6: Batch stream processing (BSP) system actually takes the vector of data queries from the previous step in linear manner and finally provides a batch of data having same characteristics to commit the transaction to reduce the time drastically with the following steps

• Query Series: Here the Queries are collected in many vectors to form a complex query handling structure.

• Normalization: All the vectors are normalized with a single or multiple characters with the tags like insert, update, delete and select

• Query Optimization: All the normalized vectors are batched and tagged to different recursive procedures to conclude the committing transactions.

The complete process of efficient database Risk management system can be representing by the following pseudo code.

OVERALL SYSTEM PSEUDEO CODE

```
Input: Database Queries
Step 1: Read database queries
Step 2: Add all queries into the queue Q.
Step 3: Remove $Q_i$from Q and Insert $Q_i$into the database
Step 4: Receive more queries in the queue $E_Q$
Step 5: Risk situation identification by measuring Queue size
Step 6: Fuzzy logic to classify the Risk situation intensity
Step 7: Preparing replacement queries by removing all remaining queries from the Queue into vector.
Step 8: Passing query vector to Batch stream process
Step 9: Batching the queries
Step 10: Execution of queries in a go to save time
Step 11: Stop
```

IV. RESULTS AND DISCUSSIONS

To show the effectiveness of proposed system some experiments are conducted on java based windows machine using netbeans as IDE and MYSQL 5.0 as database.

To measure the performance of the system we set the bench by putting our system in many tests as mentioned below

4.1 Performance Time Measurement

In our proposed system we used about 5000 database tuples to make our system busy in executing these queries by collectively feeding the data to our system in a given instance. As the system is enriched by the BSP system it drastically reduces the time to execute records in a common database respiratory. The performance of our system has shown below in the figure no 4.

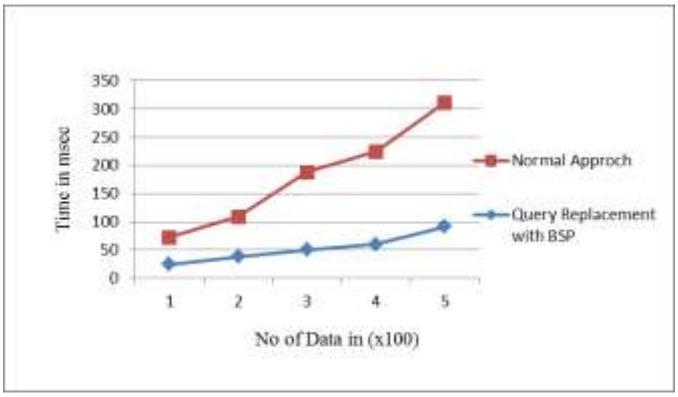

Figure 4: Performance analysis

Where X – axis indicates the number of data records in thousands and Y-axis represents the time required to integrate in millisecond. As graph is clearly declares our system is quietly enriched in performance time.

4.2 RMSE (Root mean square Error)

3. RMSE is a quite usable common parameter to judge the general purpose error metric for numerical predictions.
4. The use of RMSE as a measuring parameter amplifies and severely punishes large errors. And it can be represent with the below shown equation

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (1)$$

Where y_i represents the predicted output and y_i^ represents the outcome from the system.

On plotting of the RMSE values for different number of runs in between normal approach of query processing and our proposed model of Risk aware management system using fuzzy logic and BSP we found the graph as shown in figure 5.

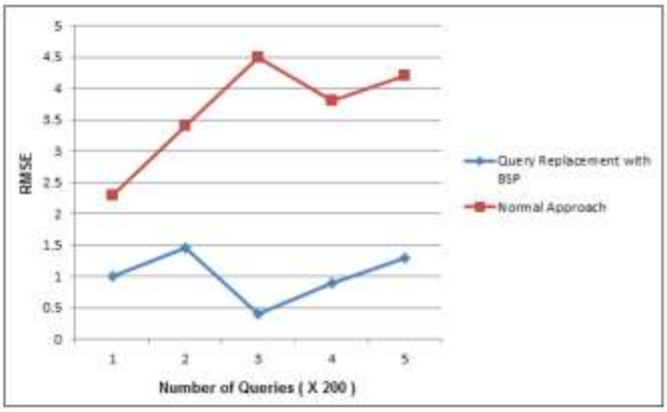

Figure 5: RMSE accuracy of the Proposed approach

Above graph indicates that proposed system is having less RMSE value than the normal approach .This shows the better performance of our idea of using fuzzy logic with batch stream processing for database risk management system using query replacement methodology.

V. CONCLUSION AND FUTURE SCOPE

In the proposed approach by using fuzzy logic System successfully identifies the database server performance parameter and it break down points. Based on these things our system identifies the high priority queries from the priority queue deepened on the extracted features of fuzzy classification.

System introduced an idea of clustering these queries from the priority queue based on the fuzzy feature score. Then these queries are batched based on the query type to commit in single go to yield best performance of the database even in the on the high risk scenario using batch stream process technique.

The proposed system can be enhance to work in distributed scenario to increase the performance time of the query Committing by batch stream processing technique .

ACKNOWLEDGEMENT

I am very grateful to my guide Prof. D. H. Kulkarni and staff members for their generous help for their many thoughtful and insightful comments that have greatly influenced this paper.I would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. We are also thankful to my colleagues and parents for their invaluable support.

REFERENCES

[1] Cao, Bin, and Antonio Badia. "SQL query optimization through nested relational algebra." *ACM Transactions on Database Systems (TODS)* 32.3 (2007): 18.

[2] Curino, Carlo, et al. "Workload-aware database monitoring and consolidation."*Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 2011.

[3] Chung, Christina Yip, Michael Gertz, and Karl Levitt. "Demids: A misuse detection system for database systems." *Integrity and Internal Control in Information Systems*. Springer US, 2000. 159-178.

[4] "A Survey of Distributed Query Optimization " The International Arab Journal of Information Technology, Vol. 2, No. 1, January 2005

[5] Freytag, Johann Christoph. "The Basic Principles of Query Optimization in Relational Database Management Systems." *IFIP Congress*. 1989.

[6] Cheng, Pau Chen, et al. "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control." *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007.

[7] Q. Ni, E. Bertino, and J. Lobo, "Risk-based access control systems built on fuzzy inferences," in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '10. New York, NY, USA: ACM, 2010, pp. 250–260.

[8] Gao, Peter Xiang, and Di Wang. "Optimizing Query Processing in Batch Streaming System."

[9] "Query Optimization in Heterogeneous DBMS" Weimin Du Ravi Krishnamurthy Ming-Chien Shan

[10] Genet, Bryan Howard, and Annika Hinze. "Open issues in semantic query optimization in relational DBMS." (2004).