International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Implementing Query Terms Linked to Virtual Databases by Metaheuristic Techniques

Suman Sourav Prasad, Sambit Kumar Mishra

Abstract: ***In general case, the database trigger may be quite applicable to signified queries to validate the database requests. Specifically, these may be essential to adopt search mechanisms to identify the query terms. In such cases it may also be required to eradicate ambiguities during updates by checking consistencies, durability. Many database systems support aggregate functions as it may be really linked to statistical analysis of large scale data. Again as per the requirement and schedule, multilevel aggregation may be thought of towards report generation and implementation of join predicates. In case of complexity, direct requests may be accessed to schedule the entire database operations. While optimizing the database queries, alternative query plans may be thought of implementing specific routines to eradicate the duplicity of query terms. It may be quite possible to containerize the query plans linked to several data servers exploring the inter operator parallelism. Also the assemblers linked to the query plans in the servers may steer the process accordingly. Considering the implementation mechanisms of database query plans inside a cloud storage system, the data may be automatically partitioned and replicated. The servers may change dynamically the existing load in response to the query plans. The queries as well as the transactions may be uncommon during optimization process and applications may be communicated following standard activity protocols linked to the database servers. Linking the query terms to the databases, it may also be required to incorporate metadata towards plan execution. Many times transactional database applications linked to relational cloud may have the provision of configuring and accessing the data and may face the challenges like scalability and privacy. To overcome these issues, the tasks may be relocated and rearranged linked to database servers by which better performance may be achieved dealing with complex transactions. Also the aggregation methods or techniques linked to data partitioning may enable the structured queries to yield better performance. In this paper it is intended to obtain query terms along with the threshold values linked to virtual databases.***

Keywords: ***Query terms, Join indices, Virtual machine, Query plans, Metaheuristic, Threshold value***

I. INTRODUCTION

It may be very clear that towards optimum utilization and application, the databases may be required to be virtualized. In return, it may help the user to have realization and conceptualization of databases. In many cases, the database virtualization may have the option to hide the locations linked to the databases while querying the databases. Sometimes it may also be very essential to calibrate he databases towards achieving perfection. In this regard, the performance of database in cloud platform may be achieved by initiating quick match techniques to single queried databases or group of queries linked to multi databases. In such cases, performances may be measured through several parameters like throughput, CPU utilization. In general, the query execution plans may be generated by choosing the optimal query plans with minimal execution cost. In the same time, the inference engine linked to execute queries may utilize the query execution plans towards evaluation and estimation of storage complexities. Also the increment of databases also in virtual environments along with queries may be directly proportional to the number of alternative execution plans. To provide the realistic mechanisms and solution linked to the optimization, it may be required to integrate the part and portion of the databases. In such cases, the optimizers may yield query execution plans to obtain the optimality. In all the cases, it may be very essential to obtain the better execution plans with minimum execution time. After that it may be required to examine the total evaluation time in the multiple databases in virtual platforms implementing multi join indices.

II. REVIEW OF LITERATURE

Marcus et al.[1] in their work have highlighted on reinforcement learning techniques to optimize the search spaces linked to join order enumeration along with query plan execution. It has been observed that the technique may require 9000 iterations towards optimization along with join order enumeration. Hester T et al.[2] in their work have adopted the mechanisms to optimize queries relinquishing the optimizers. In their work, they focused on actions initiated with traditional query optimizers and tried to estimate the query performance. Again also they tried to improve the performance of traditional query optimizers and reflected the same in their work. Taylor M. E et al.[3] in their work have elaborated on techniques linked to query optimization through training models and focusing on complex queries. In the technique the search space may be divided into smaller segments and the techniques linked to reinforcement learning may again be adopted. Leis V et al.[4] in their work have focused on the issues linked to cardinality estimation for all types of cost models. They tried in their work to match the match the query latency along with the optimizer cost. Poess M et al.[5] in their work have adopted the mechanisms to estimate the candidate queries. They observed that practically the workloads may contain less no. of queries over a single database.

Revised Manuscript Received on December 30, 2019.

* Correspondence Author

Suman Sourav Prasad*, Department of Computer Sc.&Engg., Ajay Binay Institute of Technology, Cuttack, affiliated to Biju Patnaik University of Technology, Rourkela, E_Mail : prasadsuman800@rediffmail.com

Sambit Kumar Mishra , Department of Computer Sc.& Engg.Gandhi Institute for Education and Technology, Baniatangi, affiliated to Biju Patnaik University of Technology, Rourkela, E_Mail : sambitmishra@gietbbsr.com

A. Kivity et al.[6] in their work have focused on working principle of virtual machine in different environments other host machine. They observed that the significant hypervisor may be essential for the virtual machine environment, as for a virtual machine, it may be essential to control all the resources.

D. M. Jacobsen et al.[7] during their studies, have suggested that the flexibility and the reproducibility of the containers sometimes may be adopted in HPC environments. They have also focused the advantages of using the container technology in cloud computing.

F. Zane et al.[8] in their work have focused on match action unit which may be more essential to spend a fixed amount of time per packet only, as it must sustain the determined bit rate of the protocols. It may be more essential to realize table matches in constant time. It may be obvious that if the database may require only exact matches, the match action unit may report to static RAM storage and match via hashing.

R. Avnur et al.[9] in their work again focused on the alternative strategy for utilization of match action unit. Accordingly it may be obvious that, there may not be any patterns. So the programs switched to match action unit may have the abilities to execute instructions. Also the programs may not be parameterized and may be required to modify the query patterns.

P. Bosshart et al.[10] in their work have proposed the architecture and the design alternatives linked to match action units. In general, the common query pattern as well as the join-and-group-by implementation may be done sequentially.

Duggan et al.[11] in their work have focused on mechanisms to design different data stores and measured their performance. They observed that the relational databases may store and process structured data efficiently but its performance decreases with read heavy queries. Accordingly the columnar databases may be used for analytical query processing along with handling unstructured data.

Bijit Hore et al.[12] in their work have signified the computing capacity and the high speed incoming data. It may be obvious that the cloud services for managing and exploiting submitted data may yield solutions to privacy issues linked to cloud.

Cetin Sahin et al.[13] in their work have focused on Index-based schemes to increase query performance while ensuring strong security. They observed that the prior schemes may not cope with the high rate of incoming data that occurs in a wide range of monitoring applications.

Ioannis Demertzis et al.[14] in their work have also proposed index-based strategies for answering range queries over outsourced data.

T. Dokeroglu et al.[15] during their work have focused on clusters linked to virtual machines cloud computing. They observed that it may enable users rent large amount of resources for short duration in order to run complex queries efficiently on large scale data. The rent duration may be decreased more with the help of better query optimization technique. So there may be need of investigating efficient query optimization techniques to minimize query evaluation time and response time.

J. Goldstein et al.[16] in their work have focused on materialized views to improve the query processing time for aggregation query over large relations. It may be obvious that the technique may be dependent upon efficient view matching algorithm. Also they have devised a technique to match materialized view derived from Select-Project-Join queries.

K. Anyanwu et al.[17] in their work have focused on nested triple group data model and algebra to minimize the general processing cost by implementing different techniques. The techniques may also be needed to minimize the generation of intermediate results.

S. Wu et al.[18] in their work have focused on implementation of MapReduce to minimize generation of intermediate results to manage the storage and network cost. The join operators may be implemented to evaluate the query.

I. Elghandour et al.[19] in their work have focused on physical operations linked to cache and minimized the redundancies applying matching techniques to query optimizers. They have also analyzed the performance of MapReduce linked to analytical query languages.

L. L. Perez et al.[20] during their studies have focused on history aware query optimizer associated with intermediate results. They observed that the matching views may be quite capable linked to previously estimated queries and subsequently to improve the query execution time.

III. MECHANISMS TO MANAGE DATA VIRTUALIZATION

Usually, in virtual environment, there is a provision of linking and masking of databases. Actually, the databases may be residing in the different heterogeneous servers but may be seemed to be in one platform. Accordingly, the techniques adopted in this case may help in managing the databases also in the virtual platforms. This may be implemented through virtual machine box which may be requiring minimum 32-bit as well as 64-bit host operating systems.

IV. IMPLEMENTATION OF METAHEURISTIC APPROACH TOWARDS EVALUATION OF QUERY TERMS

In very specific cases it may be essential to adopt metaheuristic approach to find solution towards optimization problems. In this situation, particle swarm optimization technique has been thought of to evaluate query terms. In many cases, the technique may be adopted through task scheduling mechanisms to minimize the delay in process execution. The criteria of optimization in general may be non liner and non consistent towards complex constraints. Also sometimes, the solution of optimality may not be achieved in all situations. To obtain the optimal solution in such scenario though difficult but is not impossible.

V. STEPS TO OBTAIN QUERY TERMS

The query terms linked to individual database may be confined within the data servers. Accordingly the records / transactions may be carried out to proportionate the data sets. In such cases, it may be required to obtain the threshold values to act as checkpoints and control the collected data as well as to measure data dependencies.

```
Input : Data servers $Ds_i$ , Query term $Qt_i$
Step 1 : Obtain the data item set from database
Step 2 : Measure the frequency of Query term , $Qt_i$
Step 3 : Nominate the set of Query terms ,$J_i$ = {Set of Query terms};
Step 4 : Initiate the search and regenerate the query terms
    for (i = 1; $J_i$ !=0; i++) do begin
$Q_{ti}$+1 = Query terms generated from Data item set;
Step 5 : for each Query term ,$Q_t$ in database do
Update the Query terms of each database linked with Data item sets
```

Table 1 : Query terms linked to Data item sets with threshold value

Sl.No.	Query terms	Data item sets	Threshold value
1	100-500	Ds1,Ds6,Ds9	6.7
2	500-1000	Ds10, Ds19, Ds29	6.4
3	1000-1500	Ds30, Ds37,Ds46	5.6
4	1500-2000	Ds50,Ds57,Ds67	4.6

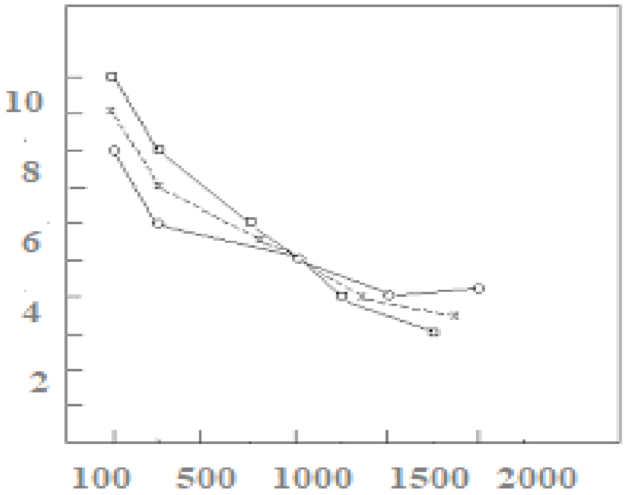

Figure 1 : Query terms vs Threshold value

As shown in figure-1, the reduction of threshold values may be indirectly proportional to the query terms linked to data item sets. It has been observed that the threshold values may be minimal with high range of query terms associated with the data item sets.

VI. ALGORITHM TO PROCESS QUERY TERMS

```
No. of iterations = 100;
Inertia = 1.0;
Check_measure = 2.0;
Maximum Query terms/Database = 500;
Step1 : Positioning and update the Query terms
for i = 1 : Maximum Query terms
    Query term(i, 1) = Query term(i, 1) + Query term(i, 5)/1.2
    Query term(i, 2) = Query term(i, 2) + Query term(i, 6)/1.2 ;
    x= Query term(i, 1);
    y = Query term(i, 2);
Step 2 : Obtain the parametric value of query terms
    Qterm_val = (x- 20)^2 + (y - 10)^2;
Step 3 : Compare and update the parametric value with query terms
    if Qterm_val < Query term (i, 7)
        Query term (i, 3) = Query term (i, 1);
        Query term (i, 4) = Query term (i, 2);
Step 4 : Obtain the updated minimal values of Query terms/Database
        Query term (i, 7) = Qterm_val;
Step 5 : Positioning the Query terms and update the Qterm_val

for i = 1 : Query terms
    Query terms(i, 5) = rand*inertia*Query terms (i, 5) + Check_measure *rand*( Query terms(i, 3) - Query terms (i, 1)) + Check_measure *rand*( Query terms(global optima, 3) - Query terms (i, 1));
    Query terms(i, 6) = rand*inertia* Query terms(i, 6) + Check_measure *rand*( Query terms (i, 4)- - Query terms(i, 2)) + Check_measure *rand*( Query terms(global optima, 4) - Query terms (i, 2));
End
```

Table 2 : Evaluation of query terms with check measures

Sl. No.	Query terms	Check_measure	Qterm_val
1	250	2.0	340.9
2	370	2.0	477.4
3	460	2.0	493.7
4	475	2.0	499.6

As reflected in table-2, the values of query terms are directly proportional to the query plans associated with the databases. Based upon the check measures, the amount of query terms may be more responsible towards estimation of query plans.

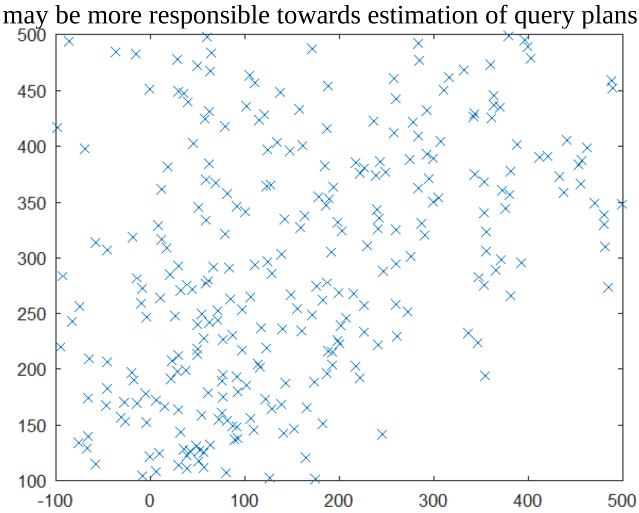

Figure 2 : No. of Databases linked to Data Serversvs Query terms accessed

Table 3 : CPU utilization linked to Query terms in Database

Sl.No.	Maximum Query term/Database	% of usage of CPU

1	577	46
2	924	55
3	1167	74
4	1469	83

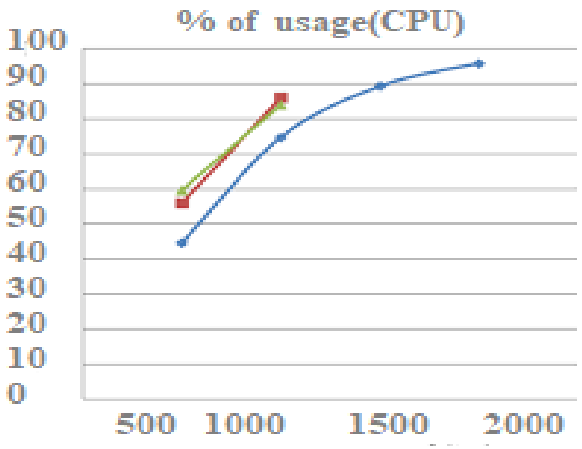

Figure 3: Maximum Query term perDatabase

As shown in figure 3, the percentage of utilization of CPU is based upon the amount of query terms associated with the database. Also the attainment of query plans linked to virtual platforms may be directly proportional to the utility factors linked to CPU utilization.

VII. EXPERIMENTAL ANALYSIS

It has been observed that the values of query terms are directly proportional to the query plans associated with the databases. The values of query terms are increased as per the queries linked to the databases.

Query terms	Queries in database	Threshold value
370	924	6.7
460	1167	5.6
475	1469	5.6

Having consistent check measures, the percentage of utilization of CPU is definitely associated with the maximum attainment of query terms within the databases. Therefore the query terms along with the query plans have direct significance while linked to virtual platforms.

VIII. DISCUSSION AND FUTURE DIRECTION

Many analytical steps have been carried out to evaluate the performance of queries linked to virtual databases. It has been observed that the parameters linked to measure the performance may not be so effective and impressive as compared with non virtual databases. In such scenario, it may be very much essential to implement the techniques to tune the database linked to cloud as well as to signify the query indices and optimize the query performance.

IX. CONCLUSION

The query execution plans may be evaluated by gathering the statistics linked to the databases in the servers either virtual or non virtual. To optimize the query performance, it may be very essential to focus on indexing strategies and effectiveness on the queries associated with the databases. There are many aspects to supervise the techniques and improve the performance through system resource integration and phases linked with database queries in virtual environments. The systems linked with these queries may be responsible to optimize the query performance. In this regard, thorough observation may be required towards representation of dynamism of schema and improvements in query patterns.

REFERENCES

1. Marcus, R., et al. Deep Reinforcement Learning for Join Order Enumeration, aiDM '18. (International Conference on Exploiting Artificial Intelligence Techniques for Data Management, June 2018).
2. Hester, T., et al. Deep Q-learning from Demonstrations, AAAI '18.(AAAI Conference on Artificial Intelligence, February , 2018).
3. Taylor, M. E., et al. Transfer Learning for Reinforcement Learning Domains: A Survey. JMLR '09.(Journal of Machine learning, 2009).
4. Leis, V., et al. How Good Are Query Optimizers, Really? VLDB '15(Very large Database Conference, August,2015).
5. Poess, M., et al. New TPC Benchmarks for Decision Support and Web Commerce. ACM SIGMOD International Conference ,2000.
6. A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. kvm: the linux virtual machine monitor. In Proceedings of the Linux symposium, volume 1, pages 225–230, 2007.
7. D. M. Jacobsen and R. S. Canon. Contain this, unleashing docker for hpc. Proceedings of the Cray User Group, 2015.
8. F. Zane, G. Narlikar, and A. Basu. Coolcams: power-efficient tcams for forwarding engines. INFOCOM, 2003.
9. R. Avnur and J. M. Hellerstein. Eddies: Continuously adaptive query processing. SIGMOD Rec., 29(2), 2000.
10. P. Bosshart, G. Gibb, H.-S. Kim, G. Varghese, N. McKeown, M. Izzard, F. Mujica, and M. Horowitz. Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. SIGCOMM, 2013.
11. Duggan, J., Elmore, A. J., Stonebraker, M., Balazinska, M., Howe, B., Kepner, J., et al. The BigDAWG Polystore System. ACM Sigmod Record, 44(3), 2015.
12. Bijit Hore, Sharad Mehrotra, Mustafa Canim, and Murat Kantarcioglu.Secure Multidimensional Range Queries over Outsourced Data. In VLDB. 333– 358, 2012.
13. Cetin Sahin, Tristan Allard, Reza Akbarina, Amr El Abbadi, and Esther Pacitti. A Differentially Private Index for Range Query Processing in Clouds. In ICDE. 857–868, 2018.
14. Ioannis Demertzis, Stavros Papadopoulos, Odysseas Papapetrou, Antonios Deligiannakis, and Minos Garofalakis. Practical Private Range Search Revisited. In SIGMOD. 185–198, 2016.
15. T. Dokeroglu, S. A. Sert and M. S. Cinar, "Evolutionary multi-objective query workload optimization of Cloud data warehouses", The Scientific World Journal, 2014.
16. J. Goldstein and P. A. Larson, "Optimizing queries using materialized views: a practical, scalable solution", In ACM SIGMOD Record, ACM, vol. 30, no. 2, pp. 331-342, 2001.
17. K. Anyanwu, H. Kim and P. Ravindra, "Algebraic Optimization for Processing Graph Pattern Queries in the Cloud", Internet Computing, IEEE, vol. 17, no. 2, pp. 52-61, 2013.
18. K. Anyanwu, H. Kim and P. Ravindra, "Algebraic Optimization for Processing Graph Pattern Queries in the
19. S. Wu, F. Li, S. Mehrotra and B. C. Ooi, "Query optimization for massively parallel data processing", In Proceedings of the 2nd ACM Symposium on Cloud Computing, ACM, October, pp. 12, 2011.
20. I. Elghandour and A. Aboulnaga, "ReStore: reusing results of MapReduce jobs", Proceedings of the VLDB Endowment, vol. 5, no. 6, pp. 586-597, 2012.
21. L. L. Perez and C. M. Jermaine, "History-aware query optimization with materialized intermediate views", In Data Engineering (ICDE), IEEE 30th International Conference on IEEE, March, pp. 520-531, 2014.

AUTHORS PROFILE

Er.Suman Sourav Prasad is pursuing Ph.D. in Computer Sc.&Engg. in the area Cloud computing and big data.. Currently he is working as faculty in ABIT Engineering College, Cuttack. He has published good number of papers in reputed journals..

Dr.Sambit Kumar Mishra is having more than 22 years of experience in different AICTE approved institutions. He has more than 29 publications in different peer reviewed International Journals and editorial board member of different peer reviewed indexed Journals.