JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
SICAIET
2024
Research Article
Collection: Artificial Intelligence and Emerging Technologies

Involuntarily Trained and Analytical Query Optimization Model

M. Abdul Qadoos Bilal[1*], Baoning[1], Muzammil-ur-Rehman[2], and Nazir Ahmad[2]

[1]College of information and Computer, TYUT, Taiyuan, 03000, China.
[2]Department of Information Technology, IUB, Bahawalpur, 6300, Pakistan.
*Corresponding Autor: Abdul Qadoos Bilal. Email: aqkhan_iub@yahoo.com

Academic Editor: Salman Qadri **Published:** February 01, 2024

Abstract: In the management of a database system, it is essential to present the outcomes of any query mix with high throughput and low execution time. The experiment conducts on the Postgresql queries on the TPC-H benchmark for the determination of response time of query mixes MPL3, while they are running concurrently. This research uses the deterministic approach for attaining the response time of the query mixes in conjunction with other query mixes. The selected query mix executes in three threads simultaneously. This makes query set(s) in each thread containing query mixes having the same ending of execution time as well as having the same starting time. It makes sure that each query set consists of at least one query mix executing at a time, the number of query mixes may vary in the query set. For attaining the research targets, it is necessary to maintain a catalog to save such query sets which have the same execution time. Due to achieving a high level of parallel processing, this research presents a query mix response time.

Keywords: Multi-crops; Plant Disease; Alexnet; GoogleNet; Fuzzy Logic; Edge Detection.

1. Introduction

The basic feature of the computer science field is parallel processing. It allows a computer machine to do several jobs simultaneously, and this feature is called multi-tasking. This opens a new area of research in database management systems. Usually, a single query executes its time slot and fetches data. It is quite demanding to develop such solutions which can enhance the performance of a dabase system through huge database size and meet the high demand for data. It's good to run more than one query simultaneously to avail multi-tasking feature of the computer system.

The sequence of arriving of any query for both concurrent executing along with other batch queries and its execution affects its response time [1,2]. Response time of a query is the usage of resources for that particular query at a specific time. The same scenario applies to the query mix but at a high level because more than two queries run concurrently at the same time in the query mix. Therefore, those concurrent queries need more resources than a single query [3,4]. The workload for any database develops at the client's request by adopting several types of queries. For example, q1,q2,..qT. In it, T is the types of databases queries for multi-programming level for representing the several queries running concurrently. Ni refers to the total number of queries used in developing the workload for any database management system as given in equation 1.

$$|w| = \sum_{j=1}^{r} N\text{j} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

For calculation the recital of database system, this is desired to treat with management of workload as there are several demands of completionof processing of the multiple queries, so that different queries can be executed in any time slot. There are also some other causess like allocation of resource, storing of data on scatterered locations on the disk and availability of resource on a disk. With the pessage of time, the size of databases grows with very high rate. This is the main cause for slowing down of the database. For any human, this becomes impractical to handle such huge and miscellaneous data. As mentioned earlier, it produced the requirement to evelop such databases that can address this issue, as discussed earlier [1, 2]. Prediction of execution time of database queries plays a vtal task while managing gigantic database systems, particularly related to the workload, this is the result of execution of different queries concurrently on the appeal of users because it may allow Database Administer (DBA) to know system behaviour, which helps him in coordination with system and provides some assessment about its performance [3]. Waiting of resources for execution of concurrent queries is based on resource contention, which depends on queuing theory. Any computational graph containing computing units on its edges and numerical information transmitted from its directed edges after calculation in a sequence node to node [4].In the database wworload, this is reported that a experience of interaction exists among the available queries. This means that any query may be run isolatedly or in the combination of more than onequeries. As far as interaction of query permits for more than one queries running simulteneousy, it also represented that their execution might be positively or negatively affect the implementation of the single query. If execution of batch queries effect positively, then buffer pool data is used by the query without waiting, this will call to another query for processing and time for completion on that data is saved. On the other hand, if the execution negatively effect, then one query becomes cause of increasing execution time of another query,because both require dissimilar resources, which may cause locking. Literature reports an important phenomenon named interaction among the database queries while running on the system. Therefore, it may enable the query to execute in isolation or in combination with other queries. The query interaction becomes the cause of positive or negative effects on those queries that are running simultaneously. The interaction of query may direct towards the interaction of query mixes. It may be possible that the interaction of query mixes provides interesting results. If it is needed to measure the performance of the database system, then managing the workload of that specific database becomes essential. The database's workload develops with the combination of several queries randomly running and several times with each other. Other parameters like availability of resources, resource allocation, and geographical storage location of data on disk. The volume of data in databases has increased rapidly with each passing day. Therefore, databases perform poorly. The fact that human beings cannot handle such vast and diverse data creates the need to build database-based applications that address these issues [5-10]. If resources are occupied by another task then waiting state is the only option to avoid locking conditions. The system also must wait for the required resources to execute the query. In order to execute the desired operation, the CPU, RAM, cache, and I/O devices may become the subject of competition [11-13]. Response time of the query affect because of multiple reasons like, imaging of query processing, query ordering, and management of capacity [14]. Query response time takes high importance at the time of dealing with huge database systems, specifically the workload, it shows the results of the execution of a lot of queries simulteneously. it helps the database administer (DBA) to do the assignment proficiently because of synchronization within the system and increase the performance [15-20]. A queuing theory determines how long it takes to execute queries based on the necessary resources [21-24]. A balances information system, for example search engines, is mainly related to data which they can represent in less time (effectiveness). Still, The focus is not on required results, which are the the user's requirement(efficiency). Users oftenly study the data speed received (time needed for representing results of search which user uses), not related to repetition (number of cycles for attaining desired results) in which data is received [25]. Data is retrieved from the database system by executing query execution plans which specify how data is accessed from the database system's source tables. The selection of any query execution plan among several other possible plans is query optimization [26-30].

2. Materials and Methods

This section explains the methodology of the research. This research chooses the heuristic and deterministic approaches to acquire the desired results. QN denotes the nine TPC-H queries {q1,q3…..q19}

used in the developing of the workload of the research. In contrast, QAVG represents the average compilation of each query {qavg1,qavg3….qavg19} and T shows the threads in which qmix running in the parallel way{t1,t2,t3}. QMIX denotes the query mixes developed in the result of the running of queries concurrently {qmix1,qmix2,qmix3……..Qmixn}. This research adopted permutation for calculation of the maximum number of query mixes formed, where n is the number of query mixes formed r represents the MPL level, and MPL shows the number of queries present in that particular query mi type.

Query mix combination execution based on one or more query mixes execute simultaneously using parallel processing because there are three separate threads containing different query mix combinations. This research applies the heuristic approach for predicting the response time of the sample space of the query mixes for categorization of query mixes concerning the response time of query mix or mixes in separate threads. Below is the pictorial representation of the methodology of query mix adoption for parallel processing.

Figure 1. Pictorial representation of the query mix adoption in parallel threads

2.1 Experimental Evaluation

The details of the hardware and software used in this experiment are given below. Dell core i7 2.0 GHz, two physical CPUs, 4 logical cores each (Total 8 cores). 6 GB RAM has 250 solid-state drives. It is a 6th generation system. Modelling performance has a variety of scenarios and features used to predict database performance. The objectives of performance optimization are parameters tuning query scheduling, and configuration of the system. Seeing response time as a crucial point for focusing provides efficiency in database queries operations.

2.2 Database Workload

This research took 01 GB data for execution. It used nine queries for developing query mixes and developing a workload of nine queries qi from TPCH's twenty-two queries qi. There 345 query mixes are developed. In this research, formerly query mix(s) were selected from the already set query mixes. For determining the selected query mix qi/p behaviour in terms of response time ti/p, the query mix qi/p executes in such a way that all query mix(s) qi/p of all the three threads have the same starting and ending time i/p of execution. This process remains the same in the whole experiment.

2.3 Heuristic Approach

This research applies the heuristic approach to the sample space of the 345 query mixes qi/p. These query mixes qi/p are the results of the former study in which 350 queries qi are used to develop these query mixes qi/p from the workload of nine TPC-H selected OLAP queries qi from the 22 TPC-H queries qi.

First of all, the individual query mixes qi/p or more than one query mixes qi/p in three separate threads Ti. From three threads Ti, each separate thread Ti has a single query mix qi/p or a combination of query mix qi/p but has the same response time ti/p in all the three threads because this research uses the random heuristic approach for the response time ti/p. After getting each combination of all three threads, Ti maintains the record of query mix qi/p participants. Also, calculate each participant query qi separately from the participant query mix qi/p and count them in descending order for maintaining the record from the allocated workload. If any query qi completed its allocated repetition in workload, then the query mixes qi/p, which have finished query qi, are not applicable for the next steps of the experiment. This experiment was used only for those query mixes qi/p, which have those queries qi that has not completed their repetition. The below-shown flow chart represents the step-by-step procedure of this research.

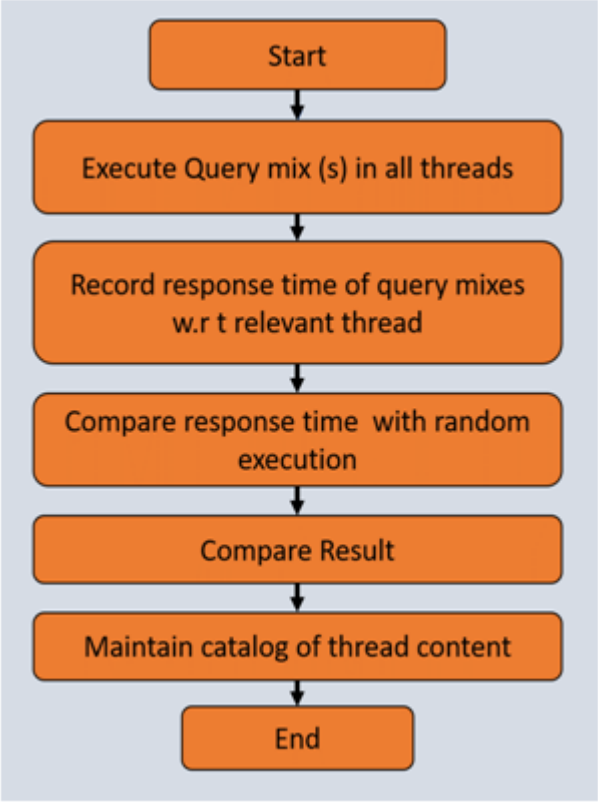

Figure 2. Query mix adoption flow chart

This research develops the query mix i/pfrom the workload queries qi, after developing all query mixes i/p, sort-out all query mixes i/p of the sample space of this research with their response timei/p. Later on, select any expected response time i/p or targeted finishing time i/p for experiment module. This process will remain the same until all the three threads Ti have one or more query mixi/p separately and acquire the targeted response time i/p. After completing each cycle of the experiment, the former process repeats again and again. The targeted response time and the contents of all threads Ti maintain in a catalog for future use. The following framework represents the query mix adoption scheme for this research.

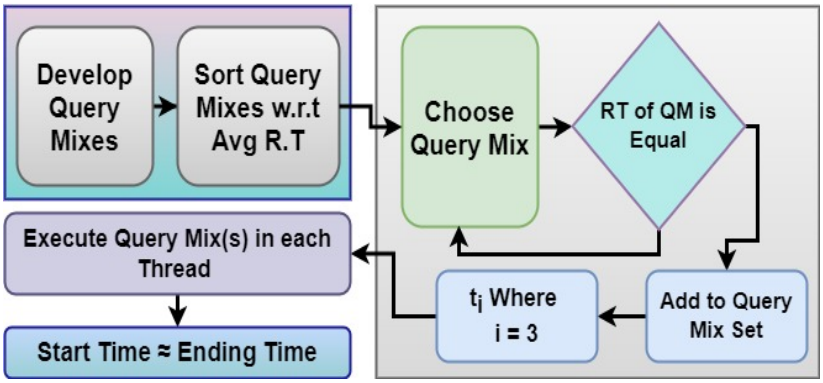

Figure 3. Query mix adoption framework

3.Results

This table1 shows the developed query mixesi/p running concurrently in three separate threads. This table follows a randomly heuristic prediction approach for representing the query mixi/p combinations. This table shows all those query mix (s)i/p in each thread Ti. The execution timei/p threshold for all query mix combinations is 63 milliseconds. These time threshold tables perform two tasks (1) show the ending time of all query mixi/p or their combinations (2) counting of each query qi in each table in descending order from the available sample space of workload of query mixesi/p. After this table, these queries qi are remaining for use in the next tables. Q1=49, Q3=47, Q5=35, Q6= 39,Q9=14,Q10= 30,Q12=35, Q14=43,Q 19=39.

Table1. Illustration developed query mixes$_{i/p}$

Thread	**T1**		**T2**		**T3**	
Query	Qid	Qrt	Qid	Qrt	Qid	Qrt
Instance 1	Q1	63	Q3	16	Q9	001≈0
Instance 2	Q 9	.001≈0	Q9	. 001≈0	Q6	15
Instance 3	Q 9	.001≈0	Q3	16	Q9	001≈0
Instance 4			Q14	15	Q19	18
Instance 5			Q9	. 001≈0	Q9	001≈0
Instance 6			Q3	16	Q9	001≈0
Instance 7					Q6	15
Instance 8					Q14	15
Instance 9					Q9	001≈0
Query mix Combination Execution Time = 63						

4.Discussion

This research is conducting for the representation of the benefits of parallel processing by execution of the queries of database in a query mix concurrently. This research opens an ne research area in the field of query optimization by discussing the batch queries concurrently.

5. Conclusions

This paper shows the response time of such query mixes. The findings of this research are that query mixes are a promising approach for implementing parallelism. It shows that all three threads comprise those query mixes or a combination of query mixes that finish the execution of all threads simultaneously, and all threads have the overall same response time.

References

1. A. Mateen, B. Raza, M. Sher, M. M. Awais, N. Mustapha, Workload Management: a technological perspective with respect to self-characteristics, Artificial Intelligence Rev.41(4)(2014) 463-489
2. B. Raza, A. Mateen, M. Sher, M. M. Awais, Survey on Automatic workload management: Algorithms, Techniques and Models, J.Compute, 3(7) (2011) 29-38.
3.]J. Zhang, N. Baoning., "A Similarity Model for Prediction Response Time of Concurrent Queries",
4. J. Zhang, N. Baoning.," A Clustering-based Sampling Method for Building Query Response Time Models", International Journal of Computer Systems Science & Engineering, 2017.
5. M. Amjad, J. Zhang., "GScheduler: A Query Scheduler Based on Query Interaction",
6. S. Guirguis, M. A. Ashraf, P. K. Chrysanthis, and A. Labrinidis, "Adaptive Scheduling of Web Transactions," in Proc. ICDC' 09, 2009, pp. 357-368.
7. A. C. Koing, B. Ding, S. Chaudhri, and V. R. Narasayya, "A Statistical Approach toward Robust progress Estimation", in Proc. VLDB' 11, 2011, pp. 382-393.
8. J. Duggan, U. Cetintemel, O. Papaemmenouil, and E. Upfal, "Performace prediction for concurrent Database workload", in Proc. SIGMOD' 11, 2011, pp.337-348.
9. M. B . Sheikh, U. F. Minhas, O. Z. Khan, and A. Aboulnaga, "A Bayesian Approach to online performance Modeling for Database Appliance using Gaussian Models", in Proc. ICAC'11, 2011, pp. 121-130.
10. M. Ahmed, S. Duan, A. Aboulnaga and S. Babu, "Predicting Prediction Time of Batch Queries Workload using Interaction-Aware Models and Simulation," in Proc. EDBT'11, 2011, pp 449-460.
11. M. Ahmed, A.Aboulnaga and S. Babu, and K. Munagala, "Modeling and Exploiting Query Interaction in Database system," in Proc. CKIM'08, 2008, pp. 183-192.
12. M. Ahmed, A. Aboulnaga and S. Babu, and K. Munagala, "Interaction-Aware Scheduling of Report Generation Workload," VLDB Journal, vol.24, pp. 589-615, Aug. 2011.
13. M.A. Q.Bilal1, N. Baoning1, M. Rehman2, N. Ahmed2, A. Hussain1, B. Ahmed1, M. Amjad1, S. Kanwal3., "Creation and Comparison of Query Mix," IJCA Journals, Vol.177, No. 21, pp. 0975 – 8887, December 2019.
14. A. Ganapathi, H. A. Kuno, U. Dayal, and J. L. Wiener, "Predicting Multiple Metrics for Queries: Better Decisions Enabled by Machine Learning," in Proc. ICDE'09, 2009, pp. 592-603.
15. M.Akdere, U. Cetintemel, M. Roindata, and E. Upfal, "Learning-Based Query Performance Modeling and Prediction," in Proc. ICDE'12, 2012, pp. 390-401.
16. B.Tozer, T. Brecht, and A. Aboulnaga,"Q-Cop: Avoiding Bad Query Mixes to minimize client time out under heavy load," in Proc. ICDE'10, 2010, pp. 397-408.
17. J. Duggan, O. Papaemmanouil, U. Cetintemel, and E. Upfal, "Contender: A resource Modeling Approach For Concurrent Query performance prediction," in Proc. EDBT'14,2014, pp. 109-120.
18. C.Macdonald, N.Tonellotto, I. Ounis., "Learning to Predict Response Times for Online Query Scheduling", in proc. SIGIR',2012,Portland,Oregon,USA.
19. S. M. Mahajan, V. P.Jadhav., "Analysis of Execution Plans in Query Optimization," IJS&ER, Vol.3, Issue 2, February-2012.
20. M.Joshi, P.Rajasthan, "Query Optimization: An Intelligent Hybrid Approach using Cuckoo and Tabu Search," IJIIT, 9(1), 40-55, January-March 2013.
21. Chaudhuri,S., Narasayya,V.R., and Ramamurthy, R. (2004). Estimating Proggress of Execution for SQL Queries. Proc. of the 2004 ACM SIGMOD/PODS conference, Paris, France, ACM, pp.803-814.
22. Luo, G., Naughton, J.F., Ellmann, C.J., and Watze, M. (2004). Towards a Progress Indicator for Database Queries. Proc. of the 2004 ACM SIGMOD/PODS conference, Paris, France, ACM, pp.791-802.
23. Konig, A.C., Ding, B., Chaudhuri, S., and Narasayya, V. R. (2011). A Statistical Approach Toward Robust Progress Estimation. Proceeding of VLDB Endoment, Volume 5, No 4, pp. 382-393.
24. Li, J., Nehme, R.V., Naughton, J. F. (2012). GSLPI: A Cost-based Query Progress Indicator. Proc. of the 28th International Conference of Data Engineering, Washington DC, USA, IEEE Computer Society, pp. 678-689.
25. Wasserman, T.J., Martin, P., Skillicorn, D. B., and Rizvi , H. (2004). Developing a Characterization of Business Intelligence Workload For Sizing New Database System. Proc. of the 7th International Workshop on Data Warehousing and OLAP, Washington DC, USA, ACM, pp. 7-13.
26. Schaffner, J., Eckart, B., Jacobs, D., and Schwarz, C., et al. (2011). Predicting In-Memory Database Performance for Automating Cluster management Task. Proc. of the 27th International Conference of Data Engineering, Hannover, Germany, IEEE Computer Society, pp. 1264-1275.

27. Luo, G., Naughton, J.F., Yu, P.S., (2006). Multi Query SQL Progress Indicator. Proc. of the 10th International Conference on Extending Database Technology, Munich, Germany, ACM, pp. 921-941.
28. Wu, W., Wu, X., Hacigumus, H., and Naughton, J.F., (2014). Un-certainity Aware Query Eecution Time Prediction. Proceeding of the VLDB Endowment, Volume 7, No.14, pp. 1857-1868.
29. Wu, W., Wu, X., Hacigumus, H., and Naughton, J.F., (2013). Toward Query Execution Time for concurrent and Dynamic Database Workload. Proceeding of the VLDB Endowment, Volume 6, No.10, pp. 925-936.
30. Marcus, R., Papaemmanouil, O., (2016). WiSe DB: A Learning-based Worload Management Advisor for Cloud Database, Proceeding of the VLDB Endowmwwnt, Volume 9, No. 10, pp. 780-791.