Creation and Comparison of Query Mix

M. Abdul Qadoos Bilal
College of Information and Computer
Taiyuan University of Technology, China

Baoning Niu
College of Information and Computer
Taiyuan University of Technology, China

Muzammil-ur-Rehman
Department of Computer Science and Information Technology
The Islamia University Bahawalpur, Pakistan

Nazir Ahmed
Department of Computer Science and Information Technology
The Islamia University Bahawalpur, Pakistan

Akbar Hussain
College of Information and Computer
Taiyuan University of Technology, China

Bilal Ahmed
College of Information and Computer
Taiyuan University of Technology, China

Muhammad Amjad
College of Information and Computer
Taiyuan University of Technology, China

Saima Kanwal
School of Computer and Communication
Lanzhou University of Technology, China

ABSTRACT
In the database management system, it is significant to show the results of any query with the lowest response time. This research conduct on the Postgresql database queries on the TPC-H benchmark for calculating the response time of the multiple queries. For acquiring this phenomenon, it is essential to adapt such practice from which more than one queries at the same time may be executed to get benefits of parallel processing. By inspiring this idea, this paper presents the query mix, which is a combination of more than two queries for multiprogramming level 3 (MPL-3), which shows that there are three randomly selected queries in each query mix. This research contained two different experiments, in the former experiment, each query executed in isolation, and in the second experiment, the combination of three queries executed simultaneously for acquiring MPL3. The results show that there is a strong correlation between the individual query and the query mix.

General Terms
Batch Queries, Multi-level Programming

Keywords
Query Mix, Execution Time

1. INTRODUCTION
Parallel processing is a primary feature of the computer system. It is our daily life observation that computer systems do several jobs at the same time. It all becomes possible because of multitasking. This revealed a new research era in the field of the database. It is the usual practice of database systems to fetch data of a single query at a time. It is needed to search such a solution that can improve the performance of the overall system with the vast size databases as well as the demand for data. It is good to execute more than one query concurrently.

It is also reported that the order in which query for execution is called either in isolation or in query mix has a strong effect on the response time [1,2]. Query response time is to deal with resources used to execute that specific query. This mechanism is also used in query mix but at a high level because, in the query mix, it is needed to measure the response time of more than one query running simultaneously [3][4].

Table 1. Notation of paper

Symbols	Description
MPL	Multi-programming level
T	Types of Queries for Multi-programming level M
q_j	Type of query j running in isolation
$q_{j/p}$	Query type j running concurrently with query type p
t_j	Time of query j running in isolation
$t_{j/p}$	Time of query j and query p running concurrently

The workload of any database system generates automatically by any client if he requests to execute any query type(s). Let $q_1,q_2,q_3.....q_T$. These are from T types of queries of database system where multi-programming level represents the multiple queries types running concurrently. N_j represents the total number of queries included in any workload, as shown in the following.

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

2. PROBLEM SIGNIFICANCE
In the workload of any database, it is come to know that there is a phenomenon of interaction that exists among the queries. This means that query can be executed in isolation as well as in the combination of more than one. As far as query interaction permit for query execution in conjunction, it also showed that their execution might affect positively or negatively on the execution of the individual query. If execution effect positively, it means that one query utilizes the data in buffer pool directly without waiting which is called by another query for execution and time for computation on that data is saved but if execution effect is negative then one query interfere in other query execution and both required different resources which may become cause of locking.

3. LITERATURE REVIEW

For measuring the performance of any database system, it is needed to deal with workload management because there are many requests for execution of database or several queries, from which different queries executed at different times. There are many other factors like resource allocation, resource availability, and data store on different geographical locations on disk. As with the passing of each day, the data volume of databases increases with a very high rate. Therefore, this becomes the primary reason for decreasing the performance of the database. For any human, it is impossible to manage such large and diverse data. This reveals the need to build such databases that can handle the issue mentioned above [5-10]. Waiting for the required resources for execution of the query is also an issue because if resources are already in use, then the system has to wait for those resources from avoiding locking condition. For getting the required resources to fulfill the desired operation, it may create competition for accessing these resources which are (CPU, RAM, Cache and I/O devices) [11,12]. Query response time may also affect due to these reasons, which are, query progress imaging, query arranging, and capacity management [13]. Database queries response time plays crucial role while managing massive database systems, especially in the workload which may shows the result of running bulk of queries at the same time on the request of users because, it may help the database administer (DBA) to whole work efficiently because of coordination within system and increase overall performance [14,15]. Waiting for the required resources for the execution of queries depend upon queuing theory [16]. A scale information system just like a search engine, those are mostly concerned with the amount of data which they can show the less amount of time (effectiveness), but they don't focus on the relevant results, which user want to get (efficiency). Users mostly analyze the speed of data (time for showing results of any search which any user uses) received not the repetition (number of repetitions for acquiring desired results) in which data is obtained [17].

4. METHODOLOGY

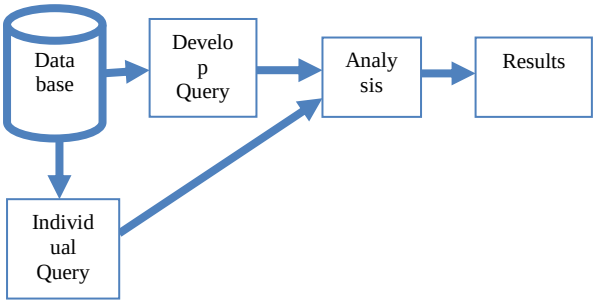

Fig 1: Methodology of the Research

This section elaborates about the methodology of current research. In the former step of experiment response time of individual queries is measured. In the later step queries are executed in the form of query mix containing at least three queries in each query mix. After getting response time of queries from both experiments a detailed analysis is performed on the results. In the end, finding of research is presented. As shown in figure 1 above.

5. EXPERIMENT EVALUATION

The detail of hardware and software uses in this experiment 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. Modeling of performance has a variety of scenarios and features used in the prediction of database performance. The objectives of performance optimization are parameters tuning query scheduling and configuration of the system. To see response time as a crucial point for focusing, it provides efficiency in database queries operations.

6. DATABASE WORKLOAD

We take 01 GB data for execution and having a workload of nine queries q_i from TPCH's twenty-two queries q_i. Queries arrangement for execution in workload plays very important for determining the query mix $q_{i/p}$ behavior in terms of response time $t_{i/p}$. In this research, queries q_i is executed isolated in the first experiment for collecting their response time t_i in isolation execution then, In the second trial, queries q_i is selected randomly for making a query mix $q_{i/p}$ or batch of query, then executed and their response time $t_{i/p}$ is noted down along with their execution.

7. RESULTS

This section tells about the results of this research. This research revealed that there is a strong correlation of 60 % between individual query and query mix.

At first, queries q_i executed individually. Results show that the response time t_i of each query is in millisecond which exists on the y-axis of Figure.2 and queries q_i exists on the x-axis. Results showed that Q1 is the most costly query q_i in the workload; its response time t_i is 6300.2 milliseconds. On the contrary, Q9 is the cheapest query q_i; its response time t_i is 2.32 millisecond. Whereas remaining other seven queries q_i have moderate response time t_i between (1360-1980) milliseconds response time t_i as shown in Figure 2.

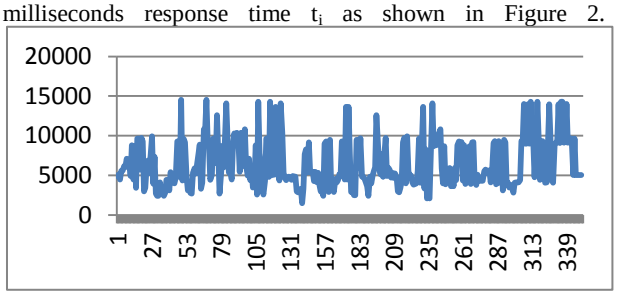

Fig 2: Individual Query response time

In the second part of the experiment, it analyzes query mix $q_{i/p}$ of level MPL3. These results are of the query mixes $q_{i/p}$. Here is the time $t_{i/p}$ on milliseconds on y-axis and batch queries $q_{i/p}$ are on the x-axis. The results show that query mix $q_{i/p}$ containing queries (Q10, Q9, Q6) is the most costly query mix $q_{i/p}$ because of its high response time $t_{i/p}$ which 16099 milliseconds. On the other hand, it is an exciting finding that there is 33 such query mixes $q_{i/p}$ exist, which has less than ten milliseconds $t_{i/p}$ response time which is shown in Figure 3.

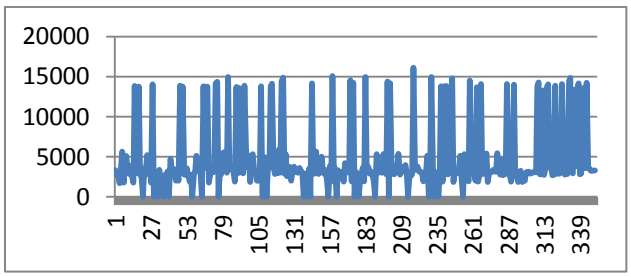

Fig 3: Query Mix response time.

By combining the analysis of both query mix $q_{i/p}$ and individual queries q_i, it is come to know that query mixes $q_{i/p}$ are executed not more than individual query q_i execution time $t_{i/p}$ as shown in Figure 4. Most of the query mix $q_{i/p}$ response time $t_{i/p}$ is approximately the same as the individual query q_i. In contrast, some single query q_i shows higher response time t_i than query mixes $q_{i/p}$, which can be observed in the resultant Figure 4.

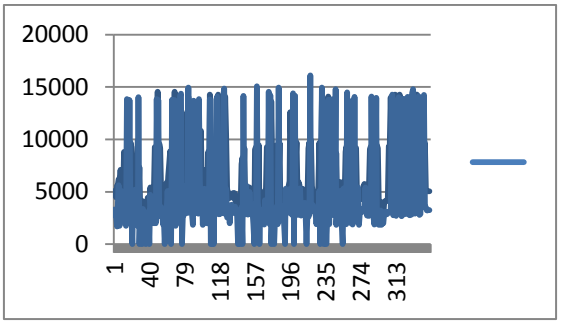

Fig 4: Individual and Query Mix Response time

By discussing those query mixes $q_{i/p}$ which has the least response time $t_{i/p}$, results show that that query mixes $q_{i/p}$ must have one query q_i nine (Q9) which reduces their execution time t_i as shown in Figure 5. In the Figure 5, fluctuating line tells the response time $t_{i/p}$ of Query mixes $q_{i/p}$.

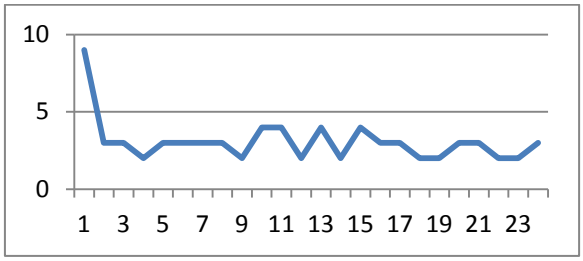

Fig 5: Least Response Time Query Mixes

The Figure 6 tells about those query mixes qi/p, which has the highest response time ti/p. Here one query mix qi/p, which discussed earlier, has the highest response time ti/p. But some other query mixes qi/p, which has high response time ti/p like (Q14, Q12, Q5) and (Q5, Q19, Q12). According to this research fifty worst query mixes qi/p have been found. As shown in Figure 6.

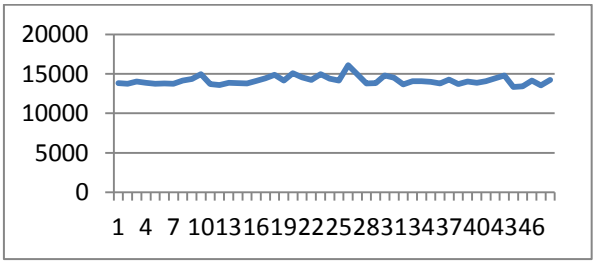

Fig 6: Highest Query Response Time

In the Figure 7, those query mixes qi/p are shown, which have medium response time ti/p. i.e., neither they have the least response time ti/p nor the highest response time ti/p. These query mixes qi/p are more in numbers, as shown in Figure 7.

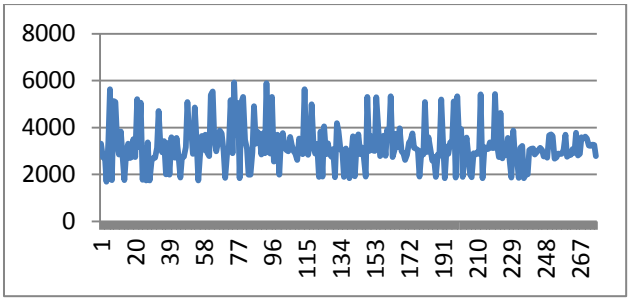

Fig 7: Moderate Query mix

8. CONCLUSION

This paper shows the response time of individual queries as well as the response time of query mixes. The findings of this research are that query mixes are the right approach for implementing parallelism. It shows that there is no so much difference between individual query and query mix response time. It also divides the query mixes into groups of least, middle and highest response time query mixes.

9. REFERENCES

[1] A. Mateen, B. Raza, M. Sher, M. M. Awais, N. Mustapha, Workload Management: a technological perspective with respect to self-characteristics, Artif. Intell.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] 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.

[14] M.Akdere, U. Cetintemel, M. Roindata, and E. Upfal, "Learning-Based Query Performance Modeling and Prediction," in *Proc. ICDE'12*, 2012, pp. 390-401.

[15] 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.

[16] .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.

[17] C.Macdonald, N.Tonellotto, I. Ounis., "Learning to Predict Response Times for Online Query Scheduling", in proc. *SIGIR*',2012,Portland,Oregon, USA.