Effective Scheduling and Parallel Database System

R.Ananth Narayanan
Department of Information Technology
Madras Institute of Technology, Chennai, India
ananth.rkn@gmail.com

Abstract—Database Management Systems (DBMS) come across several types of workloads, more precisely User workloads and System workloads. User workload consists of query statements and their execution and while system workload consists of transactions, their details, sessions involved etc. Different parameters like CPU cycles, memory storage etc., are definitely an integral part of their successful execution. Considering the nature of the database as such, throws different aspects of improvement in different fields pertaining to it. The research here introduces the concept of an efficient technique for the management of the database in all activities pertaining to it. The proposed approach strives for efficient scheduling of jobs, in other words, data in database, using a few specialized logic, detection and minimization of idleness while handling queries and also for proper and sophisticated manipulation of databases through Parallel Database Systems(PDBS). The PDBS strives to exploit modern processor architectures using software-oriented solutions for data management. The ultimate aim of this approach is to provide a way for efficient processing of data in an adoptive fashion such that it doesn't involve any human assistance/intervention.

Index Terms - Database Management Systems, efficient scheduling, minimization of idleness, parallel database systems.

I. INTRODUCTION TO DATABASE SYSTEMS

Database Management Systems(DBMS) usually work with very large volumes of data. DBMS is almost involved in all modern day applications. The growing technologies has resulted in the birth of so many advancements in this field. These include Online Transactions, Bulk processing and other Data Mining Activities that efficiently deal with very large volumes of data. There are indeed several characteristics that determine the performance of database systems. They include cost of maintaining the database, Interconnectivity between database, Safety of the database and its recovery in case of crash. Apart from all these the speed of response of Database(DB) and accuracy in data maintenance determine how efficient the DB is.

Online Transaction Processing(OLTP) is a class of information systems that facilitate and manage transaction oriented applications[1]. OLTP is used to refer to processing in which the system responds immediately to user requests. An Automatic Teller Machine(ATM) is a perfect example for this.

In early 2001 the concept of Autonomous Computing was introduced that aimed at building self-sufficient systems that can perform tasks effectively without the intervention of humans[2]. This is of prime importance in the processing and management of DB systems mainly due to the processing and manipulation of large user data, handling of complex user requests. The OLTP workload of DB doesn't have enormous data but has more users with more write activity involved in it. In addition to OLTP, DB has Data Warehouse/Decision Support Systems Workload which is enormous in terms of data, on comparison with OLTP.

The Transaction Processing Council(TPC) defines processing and database benchmarks and delivers trusted results back to the industry. The two benchmarks defined by TPC are TPC-C for OLTP and TPC-H for Decision Support Systems[3]. The former consists of read-only type of queries incl., insert, delete and update queries while the latter consists of goal oriented queries. This research proposes a scheduler for manipulation of DB to manage these workloads and also for the minimization of idleness of the DB in processing the data. Also the PDBS concept of database is introduced here to effectively remove the main drawbacks of conventional DB. High disk access time and the non supporting of large databases in a single system are two such drawbacks. PDBS is the only solution for increasing the I/O bandwidth through parallelism and for storing huge databases in a single system. High performance, High availability and extensibility are the three main advantages of PDBS.

The paper is divided as follows: Section II of the paper deals with the previous research work and the existing techniques in DBMS. Section III of the paper contains the proposed approach for efficient management of DB along with its features and experimental results involving the comparison of the existing models with the proposed approach, finally Section IV concludes the paper along with future directions.

II. RELATED WORK

Management of Database workload is the most notable aspect of DBMS. A large number of research activities are being done in this domain mainly because its growing need and importance.
This section deals about the previous researches and the existing technologies pertaining to DBMS. A few models have been already proposed to classify the workload into OLTP and Decision Support Systems[4]. One such model makes uses of two important algorithms, namely Hierarchical Clustering and Classification and Regression Tree Algorithm (CRT). Hierarchical Clustering finds the relationship between them while CRT splits the workload of the database system into the two parts categories above. The classification is performed by considering all the data items as one main group and then splitting them up into two sub-groups in such a manner that assigns high values to one subgroup and low values to another. This classification is done based on value of first data item. After this, depending upon the value of second data item again it is split into two sub-groups. This process is carried out till the last data item's value.

A database management model has been proposed by Menasce and Colleagues which takes into account the nature of tasks performed by the users. More precisely it can group similar tasks that manipulate the database and demonstrates it by the use of a graph on which similar patterns are produced for similar tasks[5]. This models works well for managing a decent amount of data while it fails abruptly when it comes to deal with enormous amount of data in the DB.

In addition to OLTP and Decision Support System workloads the database also comes across Business Intelligence Workloads (BIW). Intelligence Workload Management is a growing paradigm for database systems management[6]. It enables optimization and management of resources in a secure and compliant manner across physical, virtual and cloud environments to deliver business services for customers. A model was proposed to manage BIW that has the ability to characterize BIW by taking into account a few parameters like CPU utilization, I/O rate etc.. This model was experimented over TPC-H benchmark and was found to be effective. Though this model proved to manage and deal with BIW of DB, it suffers a drawback because this proposed model considers only the user demand as it's only parameter and doesn't take into account any parameters from the Database point of view.

A framework for DBMS was proposed in the mid 2000's that brought in a very different approach to DB manipulation. This is a stop and resume framework in DB where suspension is proposed for some queries to manage DB workload efficiently[7]. When the processing is suspended only a selected queries from the list of operations is saved and on resuming these queries start executing from the point they were saved and not from the beginning. Another similar model for query processing was proposed a few years later that introduced the concept of checkpoints. Checkpoints are placed after each cardinality and the query execution starts from the most recently saved checkpoint on restarting, incase of any suspension in execution of queries.

A new scheduler was introduced later that executed the similar queries together in batches[10]. The queries in a batch had similar characteristics like same memory space, same CPU utilization etc., This divides the workload of database into different classes based on time of execution, cost involved etc. This model was verified by a list of experiments and was validated. All these database management techniques, though they are verified, they do have some limitations. The vast developing technology in the modern era calls for more efficiency in DBMS and hence this research here aims to provide them and also these related works mentioned above take assumptions relying on the values provided by query manipulators and the results obtained based on assumed values can never be the same results as obtained on true values. The proposed module aims at providing better characterization in classifying DB workload, efficient scheduling of workload in such a manner to detect and minimize idleness while processing and also the introduction of PDBS that would totally provide a new approach on how data can be stored and managed.

### III. Proposed Module For Efficient Database Management

This research covers both the management of database workload in a single database as well as the concept of having a parallel database system.

![Architecture diagram of proposed model](image-url)
A. Classification of workload

This module effectively classifies workload of the DB system. It gets the tasks to be performed from the user and based on few defining parameters it classifies the workload into the two categories of DB workload mentioned above. The user types the workload in a task editor and by the means of which DB classifies the task into a category before the starting its execution. Case base reasoning technique is adopted here that consists of the key parameters and the type of workload.

Both OLTP and DSS workloads are initially executed in isolation for obtaining their corresponding parameters and other relevant details and the values are stored effectively in a log (name of location where it is stored). Thus whenever a new workload enters for execution the parameters and details like memory space etc., of the workload are matched with the values present in log and hence they are classified into either OLTP or DSS depending upon which values the incoming parameters match with. If the parameters of the new workload do not match with any of the two types present in the log then Bayesian algorithm is used to identify the type of workload.

Incoming workload

<table>
<thead>
<tr>
<th>Workload classification</th>
<th>OLTP</th>
<th>DSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(classified based on CBR)</td>
<td>(Both executed)</td>
<td>(In isolation)</td>
</tr>
</tbody>
</table>

Key parameters obtained

Key parameters

(key parameters stored in log)

LOG

This classification consists of assigning class labels to a set of unclassified cases. Bayesian classifier is defined by set C of classes and set A of attributes. A generic class belonging C is denoted by Cj and generic attribute belonging to A is denoted by Ai.

Let n(aijk/cj) be the no. of cases in which A appear with the value Ai_k in class C then,

\[ p(ai_k/c_j) = \frac{n(ai_k/c_j)}{\sum n(ai_k/c_j)} \]  

(1)

where p is the conditional probability and it is arrived only based on the frequency.

B. Efficient Scheduling

Once the incoming workload is categorized into OLTP and DSS, it's effect on the DB can be accurately determined. Depending upon the workload's parameters and the type of workload this proposed model schedules the workload efficiently by using a scheduler that is based on a logic which comes as a variation of the logic that is used in Boolean Algebra where there is only one truth value for a state. For e.g. in Boolean Algebra we assume 0 for false and 1 for true.

The logic we use here has truth values ranging from 0 to 1. Hence it works on the concept for partial truth. This logic is known as the Fuzzy Logic. Gaussian function is used to evaluate the input and they generate a fuzzy value ranging from 0 to 1. There are a set of Fuzzy rules that check the fuzzy value and publish the expert decision in the form of fuzzy words. The result in each stage is converted into fuzzy set and finally it is converted to output values using maximum methods (FOM, LOM, MOM). Just like what is done earlier for finding the type of workload, here we again run the different workloads in an isolated environment and store the parameters like workload_id, type, execution time, no. of queries, response time etc., for each workload in a separate log.

By execution of these workloads the idle time of the DB can be calculated from the parameters mentioned above and are stored for each workload in the log for future use. When a new workload is executed the idleness is calculated and is matched with the idleness values in the log. If the exact value is not found then a new log entry is created for the particular workload and idleness is stored along with its parameters. With vastly improving technology once the idle time of database systems are tapped they can be effectively used to perform other activity that will increase the efficiency of the DB. One such activity is the tapping of idleness of DB and performing data backup in that stipulated time.
One such way to minimizing DB idleness is the use of the concept of Parallel Database Systems (PDBS). The increased throughput achieved through inter-query parallelism and decreased response time makes it most efficient to use[11]. PDBS architecture facilitates the concept of shared memory that is very much known for it’s simplicity. Load balancing is also achieved here. However the cost involved is a bit on the higher side.

### C. Parallel Database Systems

Shared Disk, Shared Memory and Shared nothing Architectures exist in Parallel Database Systems. Shared Disk Architectures are cost effective. If nothing is shared between the databases then high availability is found and also cost is drastically reduced. This is a form of architecture known as Shared Nothing Architecture.

There are two types of query parallelism involved here. They are namely inter and intra query parallelism[13]. The former deals with execution of multiple queries generated by concurrent transactions while the latter includes two types of operators. Inter operator executes several operators of the query tree on several processors in parallel[14], while intra operator executes the same operator in different processors, each one working on a subset of data.

Given below is a simple algorithm involving the concept of PDBS.

**Algorithm PNL**

```
input: R1,R2,...,Rm //Fragments of relation R
      S1,S2,....,Sn // Fragments of relation S
      JP //Join predicate
output: T1,T2,....,Tn //Fragments of result
begin
  for i from 1 to n do in parallel
    (Send R entirely to each S node)
    send Ri to each node containing a fragment of S
  end for
  for j from 1 to n do in parallel
    (perform the join at each S node)
    begin
      Tj=(R,Sj,JP)
    end for
END(PNL)
```

**Fig.3 Fuzzy based Scheduler**

In this architecture each device has its own disk as well as specified allocated memory space. Shared Memory Architecture take care of memory allocation constraints by allowing sharing of memory space through a common bus.

Data allocation is one of the key techniques in PDBS. Distribution of data across different database system efficiently maintains the consistency and accuracy of the DB[12]. Data fragmentation technique utilizes the use of three strategies. Round Robin partitions the i th tuple for n partitions(i mod n) while Hashing utilizes hash functions to assign partition number to some attributes , Range partition distribute tuples based on the range of values for some attributes. This maximizes the system performance minimizing the DB idleness time and maximizing the throughput.

There are two types of query parallelism involved here. They are namely inter and intra query parallelism[13]. The former deals with execution of multiple queries generated by concurrent transactions while the latter includes two types of operators. Inter operator executes several operators of the query tree on several processors in parallel[14], while intra operator executes the same operator in different processors, each one working on a subset of data.

The above graph shows the various existing DB models, their corresponding percentage of efficiency and their comparison with the efficiency of PDBS model. CM stands for Characterization Model, QSHM stands for Query Shuffler model, ECM stands for E-Commerce Model, SSM stands for Start Stop Model, MLS stands for Mixed Workload Scheduler and PDBS is the proposed model. PDBS has the highest efficiency of 81.2%. Efficiency is obtained by the number of transactions a model executes in unit time. Efficiency is given by: 

\[ E = \frac{\text{number of transactions}}{\text{unit time (sec)}} \]
each model represents the number of transactions that are executed in one second. It is very evident from the above graph that the proposed PDBS model has the highest efficiency among all the currently existing models and by the means of utilization of DB idleness for data backup, it ensures data security as well as data integrity.

From the above figure it is seen that as the number of CPU's increases the transactions per second of the parallel database system also abruptly increases which makes the system more efficient and effective. All these results are based on their execution in Oracle environment. These results signify the drastic increase in performance by the usage of PDBS and also show the effect of increasing performance of PDBS due to increase in the number of CPU's.

![Graph showing the effect of no. of CPUs on PDBS](image)

**IV. CONCLUSION**

This research provides effective and very efficient management of the database without any human intervention. The detection of DB idleness gives a very large scope for development in this field considering the fast growing technology. The research first classifies the database workload and then based on the type of workload, effectively schedules it for processing and detects the idle time of the DB while it's being processed. The tapped idleness, by the can be used to do wonders in this field. One such way of having a very fast response time even while processing very large data is by making use of PDBS.

The availability of fast microprocessors and small inexpensive disks is an ideal platform for PDBS. By employing data partitioning, intra-operator parallelism etc., converting an existing DB to PDBS is a really straightforward one. This architecture is called Hybrid architecture which includes Non Uniform Memory Architecture (NUMA) that employs non uniform memory access and Cluster which consists of combination of Shared Nothing Architecture along with Shared Disk Architecture(SAN/NAS) and is subsequently formed by a group of connected computers. All the modules described above are implemented in Oracle environment. This implementation aims to detect the DB idleness in the existing DB models and thereby minimizing it by using PDBS. Its powerfulness lies in keeping the database active all the time without any sorts of idleness.

**REFERENCES**


