Performance Improvement Design and Algorithm on Database Tuning Using Systolic Architecture

Dr. Faizur Rashid
Asst. Professor, Department of Computer Science
College of Computing and Informatics, Haramaya University, Ethiopia

Abstract

The need of the society has been transformed in the automated system, which requires more intelligent execution. The performance of the database is not easy task. Investigation has been made in different issues of performance improvement. The development in the performance of the business is required every day and accordingly machine requires improving its performance on the database. Here 'Database Processor' is being introduced to improve the performance. It provides a different segment of the processing using Systolic Architecture to perform millions of operation per second. The systolic architecture is being integrated with its execution area where cells of data are connected to each other and perform a sequence of operation on the data that flown between them. In this paper, a performance improvement algorithm based on systolic architecture estimated tuning parameters is presented. During processing elements (PE) into its unit, it will diagnose the affected process and alleviate them. The proposed method shows, it is effective in improving query response time as speed up ratio.

Key Words*: Performance Improvement, Systolic Architecture, Database Processor, DBA, Speed up ratio, Processing Element (PE).*

I. Introduction

In the era of web positivity, system performance is more important to businesses than ever before. A poorly performing web site is just as bad as one that is actually down, because users get frustrated and are unlikely to return. The DBA are expecting to solve the problem in as less in market cost. They like to see performance of the machine in quick responding way.

Many systems provide the performance tuning in database application. Among them, many separate components measure the database processes. The many sets of measurements can lead to data overload for the DBA who is tasked with analyzing a problem. It is common for the DBA to measure the task for long hours into their system and analyze the performance [1].

To make a performance better of the database is not in the user hand and a user to perform an operation again, to requiring the building of a copy of the database, which may take weeks to set-up. Many times

performance issues go unresolved because it is not regarded as cost effective to spend the time and resources to reproduce the problem. Once the speed-up ratio of the database increase [5] the performance of the database can be motivated. Such unrelated metrics make it infeasible to weigh the performance impact of one component of the database with the impact of others components. The design is being presented to show the performance of the database using systolic architecture.

II. Related Work

Many framework and methodologies have been proposed that visualize the system performance indicators, analyze the symptoms and auto tune the DBMS to deliver enhanced performance. Use of Operating System views and Indexes, Pruning table and column sets[2], Spatial database and tuning, Use of self healing techniques [3], use of physical design tuning are among the proposed solutions. The classical control is modified and a three stage control involving Monitor, Analyze and Tune [4] is employed to ensure system stability. The Automated tuning systems have been proposed that employ a feedback control mechanism layered on top of a target system. Oracle's previous solution for performance tuning was Stats pack. Stats pack is a tool that takes snapshots of database performance statistics and generates reports across a pair of snapshots [6].

Analyze and Tune [6] is employed to ensure system stability. These time instances are responsible for saving the current status information or a snapshot of the server to the log. This architecture has high monitoring overhead, due to the fact that when large number of parameters to be monitored, almost every module's status information has to be stored on to the log and if done frequently may eat up a less CPU time. The performance of the parameter is on the amount of impact they produce by their workload. A formal knowledge framework for self tuning database system is presented in [7] that define several knowledge components. The knowledge components include Policy knowledge, Workload knowledge, Problem diagnosis knowledge, Problem Resolution Knowledge, Effect or knowledge, and Dependency knowledge.

The above collected information influence to analyze through the systolic architecture by giving time to individual instances as processing element (PE) or processing data for a desired output speed up time. The Systolic architecture would then estimate the extent of correction to be applied to the key system parameters that help scale up the system performance.

III. Performance Improvement

To increase the efficiency of read or write system log files into DBMS in shortest time is commonly called as performance tuning. The objective is to collect the system log files and apply information extraction techniques and also gathers key system parameters like speed up ratio, number of active processes and the data that are processing way. The architecture presented in this paper, only one parameter namely, the computing memory is tuned. Using the statistical information of these three parameters to train the Systolic Architecture and generate an output that gives an estimate of the optimal system speed up time. Since, the DBMS are dynamic and continuously running around the clock, the above information must be extracted without causing any significant system overhead. Being a human being a DBA cannot measure the memory consumption (flow of the task) in the memory area. A small algorithm is being introduced to measure the require capacity of the memory [8].

Performance improvement, by its nature, considered again and again. The identification of the part of the system which makes barrier in the performance improvement is called bottleneck. For this reason, removing the first bottleneck might not lead to performance improvement immediately, because another bottleneck might be revealed. Also, in some cases, if serialization points move to a more inefficient sharing mechanism, then performance could degrade. With experience, and by following a rigorous method of bottleneck elimination, applications can be debugged and made scalable [9].

Systematic tuning follows these steps:

- Assess the problem and establish numeric values that categorize acceptable behavior.
- Measure the performance of the system before modification.
- Identify the part of the system that is critical for improving the performance. This is called the bottleneck.
- Modify that part of the system to remove the bottleneck.

A performance problem may be identified by slow or unresponsive systems. This usually occurs because high system loading, causing some part of the system to reach a limit in its ability to respond. This limit within the system is referred to as a bottleneck. A handful of techniques are used to improve performance.

Data drives today's businesses, and managing databases often involves complex planning, time management and system wide routine task implementation. Database automation helps enterprises better manage their database operations, reducing down-times as well as the overall time taken in the system.

IV. Systolic Architecture for database performance

This architecture consists of an array of PEs as cell. Each cell performs its operation and passes it to the next neighbor. The Systolic array process element is being focused in the figure-1 (a) and (b) showing to incorporated way with the database. The PE in the memory will increase the efficiency of the database.

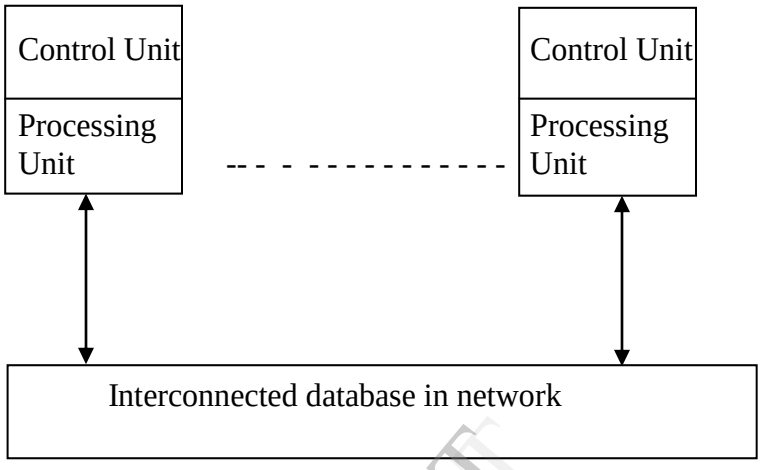

Figure 1, (a)

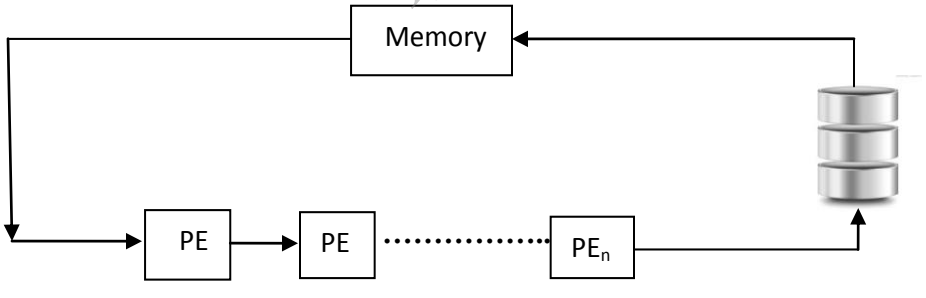

Figure 1, (b)

In above figure each PEs as log files assume to assign the computation time t=0 in beginning stage. Projection vector of data

$$d = \begin{bmatrix} d1 \\ d2 \end{bmatrix}$$

Two or more of d are executed by the same processor, where processor space vector $P^T = (p_1, p_2)$. Scheduling task in vector, $S^T = (s1, s2)$ and any PE with index I, would be executed at time, $S^T I$. The memory utilization efficiency can be assumed by

$$HUE = 1 / | S^T d |$$

Hence, because two PEs are executed by the same processor are spaced $| S^T d |$ time unit apart.

V. Proposed Database Recovery and Improvement Model using Systolic Architecture

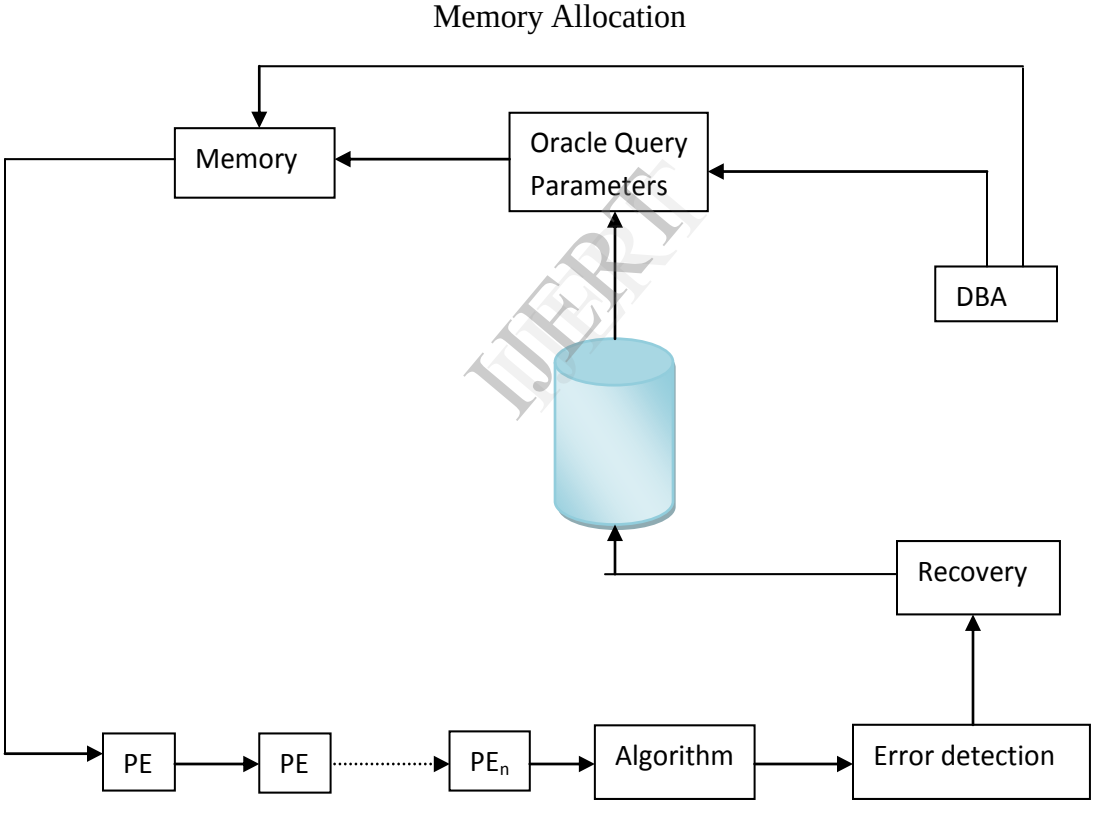

Figure 2- Database design using systolic architecture to recover error automatically

VI. Algorithm

1. db tuner MEMORY_SIZE
2. Start
3. Execute Application and Algorithm
4. Calculate HUE = 1 / | S^T d|
5. IF (HUE > 0)

{
Error detection in PEs
}
{
a. Start
b. Recovery Process
c. End
}
ELSE
{
Proceed Next PEs
}

6. Exit Application
7. End

VII. Conclusion and Scope

The design of algorithm, script or any architecture to improve the database is hard to get success. Many design for automated data has been proposed in the past using physical or logical tuning by the experts of databases. Here it has been proposed to improve its performance using a systolic architecture to accelerate the DBA work. An algorithm will be design to sort the error and recover them automatically. The DBA can monitor the situation of memory and speed the ratio of memory according to the need.

References

[2] Chaudhuri, S.; Weikum G, "Foundations of Automated Database Tuning", Data Engineering, April 2006.

[3] Peng Liu, "Design and Implementation of Self healing Database system", IEEE Conference, 2005.

[4] Rimma V. Nehme, "Dtabase, Heal Thyself", Data Engg. Workshop April 2008.

[5] Debnath, B.K.; Lilja, D.J.; Mokbel, M.F., "SARD: A statistical approach for ranking database tuning parameters", Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference, April 2008 .

[6] Foundations of Automated Database Tuning, VLDB '06, September 12–15, 2006, Seoul, Korea.Copyright 2006 VLDB Endowment, ACM

[7] Soror, A.A.; Aboulnaga, A.; Salem, K., "Database Virtualization: A New Frontier for Database Tuning.

[8] Gerhar Weikum, Axel Moenkerngerg et. al., Self-tuning Database Technology and Information Services : From wishful thing to viable Engineering", Parallel and Distributed Information System 1993.

[9] I. Alagiannis, *Towards Adaptive, Flexible, and Self-tuned Database Systems*, (DIAS, I&C, EPFL) in EDIC-ru/05.05.2009