STUDENT PORTAL USING PARALLEL COMPUTING ENVIRONMENT

[1]Shubhangi Rawal, [2]Mansi, [3]Priya Tyagi, [4]Gaurav Agarwal
[1]Student, [2]Student, [3]Student, [4]Assistant Professor
Department of Computer Science & Engineering,
ABES Institute of Technology, Ghaziabad, Uttar Pradesh, India

Abstract: Simultaneous execution of processes and the calculations are done in parallel computing for the computation. The problems which are large can be divided into smaller problems, and then they can be solved at the same time instead of solving at different times. A DBMS is a system which is closed and has its own data, which must be in its corrective format and slows down the improvement for data-hungry applications such as Machine Learning. [2] Parallelism has been used for high performance computing for long. This could also be computed by using Hash Partitioning. Dividing the logical database and its component elements into different parts which are independent. Manageability, availability reasons, performance, or load balancing are normally done using database partitioning. We can also build separate smaller databases for performing partitioning**.**

I. INTRODUCTION

In parallel computing, many processors can execute simultaneously. Traditionally, computer software has been written on a single computer which is executed one at a time, once it is done the next one starts which was very time consuming when implemented.

Parallel computing executes many processes together as it has multiple processors. Parallel computing also takes less time in searching as compared to sequential search which reduces its complexity.

The project Student Portal Using Parallel Computing Environment deals with the facility which will help the colleges as well as the students of the colleges to get the information about the activities which are taking place in the nearby college. The student or colleges have to first register on the portal in order to access the portal. Once the student or the college is registered the students or college can search for which ever college or location they want to, who have registered on the portal. As we have data in this work, instead of searching data sequentially which is very time consuming and increases the time complexity of the algorithm, we will do the searching using parallel computation which will reduce the time taken to search the data as well as will reduce the time complexity of the algorithm by dividing a task into small partitions.

II. BRIEF LITERATURE SURVEY

Traditionally, software has been written for sequential computation, a problem is broken into a discrete sequence of instructions. Instructions are executed in a pre-defined sequence one after another on single processor. At a moment, one instruction is executed.

In parallel computation unlike serial computation, some instructions which are repetitive in nature are divided on different processors for fast and efficient result. An algorithm cannot be completely executed simultaneously, only some parts are executed.

These days processing of data is done by text data mining. The input text are structured using the text data mining (frequently by parsing, adding derived lexical features, abolish others and putting into a database), patterns are obtained within the structured data and the output is evaluated and interpreted finally. [6]

Work is done on mapping of data which decreases the determined attempt of evaluating large size of data. The two-dimensional mapping needs lesser communication operations (only broadcast and decreasing operations with the rows and columns of the processor grid) in every coarsening step measured with one-dimensional mapping. [5]

This project is working on parallel implementation of finding part of student portal by using hash partitioning. In this horizontal fragmentation of table is performed. For the range, list and hash strategies, an attribute called as partitioning key, should be taken within the table attributes. The partition of the table rows will be based on the value of the partitioning key. In the range strategy, a given range of values is given to a partition. The data is divided within the nodes in such a manner that every partition consist of rows required for mapping between the partitioning key values and its nodes rely on the outcome of a hash function. [4]

Load balancing –The main issue of fragmentation of table is balancing load conforming to size of data.

As the resources are vital in nature, the load balancing of the jobs in a grid environment can greatly influence the grid's performance. [3]

By assuring that *load* (overall work) is balanced, we assure fast and efficient computation. We have to keep in mind both **communication time** and **computation time.**

Load=lower bound (n/p)

Where, n is size of data and p is number of partitions

If the size of data is 10,000 and we create 10 partitions then load =1000. In this computation will be less but communication time will increase, so it will not be efficient. But if we create 5 partitions then both computation and communication time will be minimum. That's how load need to be balanced.

Algorithms vary significantly in how parallelizable they are, ranging from easily parallelizable to completely unparallelizable based on the efficient working.

Work and Depth- In parallel algorithm, the running time and complexity determined by work and depth of graph determined by a problem. **Estimating** performance is a major part of analysing algorithms. Although these analysis are not used to determine the precise running time of an algorithm on a specific machine, it is necessary in evaluating how the running time increases as a function of the input size. To evaluate performance, a formal model is required to account for the costs. [1]

In parallel computing, the most common models are based on a set of processors connected either by a shared memory, as in the Parallel Random Access Machines (PRAM)), or through a network, as with the hypercube or grid models. In such processor-based models, performance is calculated in terms of the number of instruction cycles a computation takes (its running time) and expressed as a function of input size and number of processors. In this project, we are dividing data rather than using multiple processors but as it is still executing work in parallel form, so we can compute efficiency in same way.

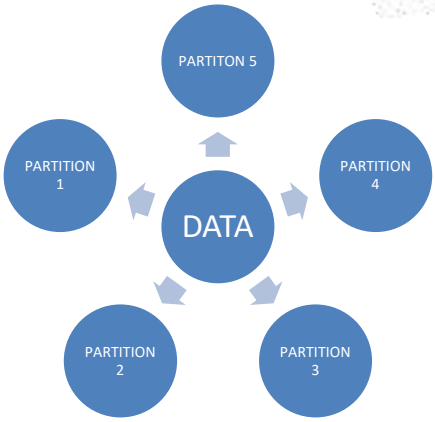
Amdahl's law- is a formula which gives the theoretical speedup in latency of the execution of a task at fixed workload that can be expected of a system whose resources are improved.

Amdahl's law can be formulated in the following way:

$$S_{latency}(s) = 1/((1-p)+(p/s))$$

Where

- $S_{latency}$ is the theoretical speedup of the execution of the whole task;
- s is the speedup of the part of the task that benefits from improved system resources;
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.
- In this project set of data will be divided in 5 partitions for efficient working on single processor and to achieve parallel computation through single processor. Through hash partitioning, data structure for all partitions will remain same but data will be divided which is horizontal segmentation.



This project is using oracle virtual box for executing multiple instructions simultaneously. Through oracle virtual box each partition is mapped with virtual machines which are given a specific memory to perform search. This helps in attaining parallel environment over single processor.

III. PROPOSED METHODOLOGY

Algorithm-

Step 1- Take input from student portal.

Step 2- Divide data in table through hash partitioning in five partitions.

Step 3- Map each hash partition to virtual machines (VM) through oracle virtual box.

Step 4-Search data through student portal in two ways:

a) By college name

b) By checkpoint

Step 5- Read output produced.

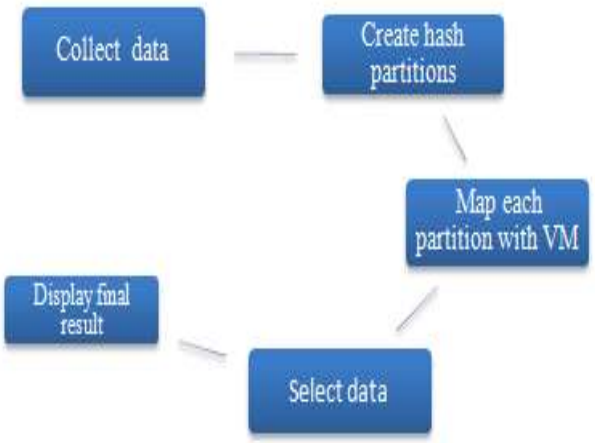
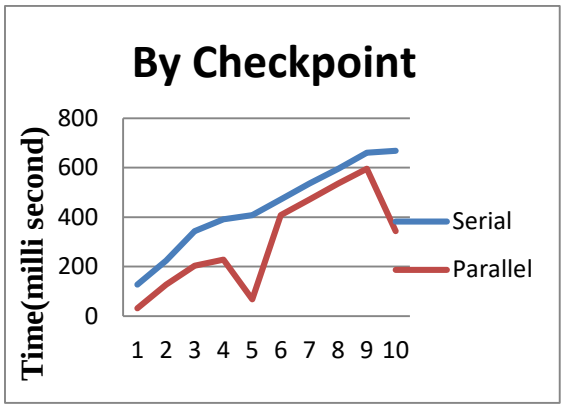


Figure: Block diagram

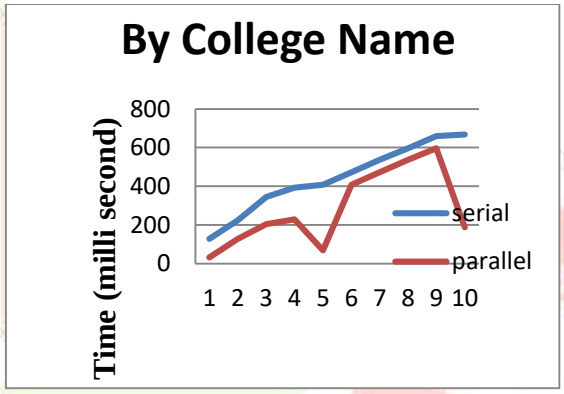
Point	Serial time (ms)	Parallel time (ms)
1.	100	28
2.	180	108
3.	224	72
4.	224	136
5.	264	132
6.	280	180
7.	288	188
8.	324	160
9.	328	204
10.	560	188

IV. RESULT:

By Checkpoint:



By College Name:



Point	Serial time (ms)	Parallel time (ms)
1.	128	32
2.	224	128
3.	344	204
4.	392	228
5.	408	68
6.	472	408
7.	536	472
8.	596	536
9.	660	596
10.	668	344

Average Time:

Modes	Serial time (ms)	Parallel time (ms)
Checkpoint	442.8	301.6
College name	277.2	139.6

V. CONCLUSION

Now a days, there are many ways to deal with large amount of data across the world such as using distributed systems for dealing with problem and parallel computing but these processes demands large setup price and for handling data which is comparatively less than data across world, it is somewhere inefficient. In those cases you can use partitioning techniques over data for efficient search. To make it more efficient partition data based on it' size like this project. This technique of partitioning data provides parallel computing over data on a single processor device which is cost and time efficient. In this project, hash partitioning is used but there are many other types of partitioning provided by oracle such as range partitioning, list partitioning, columns partitioning,

key partitioning and sub partitioning. Based on the requirement these different kind of partitioning can be used and data searching can be made for time efficient.

VI. REFERENCES

1. Guy E. Blelloch, Bruce M. Maggs "Parallel Algorithms". USA: School of Computer Science, Carnegie Mellon University, 2015.
2. Ronald Barber, Christian Garcia-Arellano, Grosman Ronen, Rene Mueller, Vijayshankar Raman, Richard Sidle, Matt Spilchen, Adam Storm, Yuanyuan Tian, PinarTozun, Daniel Zilio, Matt Huras, Guy Lohman, C. Mohan, Fatma Ozcan, Hamid Pirahesh "Evolving Databases for New-Gen Big Data Applications". IBM Research–Almaden, 2017.
3. Kapil B. Morey, Sachin B. Jadhav "A New Approach for Dynamic Load Balancing using Simulation in Grid Computing". IRJET, Volume-3, Issue-1, 2016.
4. Caio H. Costa,Joao Vianney B.M. Filho, Paulo Henrique M. Maia, Francisco Carlos M. B. Oliveire "Sharding by Hash partitioning: A datatbase scalability pattern to achieve evenly sharded database clusters". Conference Paper, 2015.
5. George Karypis, Vipin Kumar "A parallel algorithm for multilevel graph partitioning and sparse matrix ordering" Journal of parallel and distributed computing.1998.
6. Muneo Kushima,Kenji Araki, Tomoyoshi Yamazaki,Sanae Araki,Taisuke Ogawa,Noboru Sonehara "Text data mining of care life log by the level of cared required using key graph" IMECS,2017.