G307

Parallel Programming to Increase Performance of Binary Search Tree

Cipta Ramadhani
JurusanTeknikElektro
FakultasTeknikUniversitasMataram
Nusa Tenggara Barat, Indonesia
cipta@te.ftunram.ac.id

Wahyu Ramadhani
STMIK SyaikhZainuddin NW
NadhatulWathanAnjani
Nusa Tenggara Barat, Indonesia
Wahyurama2@gmail.com

Abstract—**Binary Search Tree is one of the most useful fundamental data structure for dynamic datasets. It can be used to sort numerical data, character string and any other data in a sophisticated way. However, the performance in term of running time for Binary Search Tree (BST) increases when a data become large, especially for data mining and detecting pattern in DNA sequences. To increase speed performance of Binary Search Tree, there are several strategies were applied such as using super computing or conducting parallel programming. In this research we applied parallel programming by using Message Passing Interface (MPI) to sort numerical data. This method is more efficient and less cost compare with using super computing. The basic idea of parallel programming is dividing root into sub-root and executing sub-root simultaneously on multiple cores. MPI provides a set of send and receive function that allow data to be processed on multiple cores. This method is very useful to find data faster than in a sequential way, but the consequence is the coding time takes little bit longer than usual.**

Keywords—Binary search tree, Message passing interface, Parallel programming.

I. INTRODUCTION

Binary search tree (BST) is a binary tree where the root node values is greater than its left children and smaller than its right children. We can represent tree search by a linked data structure in which each node is an object. In addition to a key value, each node contains attributes left, right and parent that point to the nodes corresponding to its left child, right child and its parent respectively. The operations that can be performed on BST including searching, insertion, deletion and other queries. Basic operations on a binary search tree take time proportional to the height of the tree. For a complete binary tree with n nodes, such operations run in O($\log_n$) for the worst-case time. However, the performance in terms of running time for BST increases when data become larger and that case becomes a big deal if we run BST to process data mining which contain different data type. To overcome this problem, parallel programing can be applied as a solution to reduce time consuming problem of BST.

A. *Why Parallelization*

Parallelization is another optimization technique to further enhance the performance. The goal is to reduce the total execution time proportionally to the number of cores used. If the serial execution time is 20 seconds for example, executing the parallel version on a quad core system ideally reduces this to 20/4 = 5 seconds. [1]

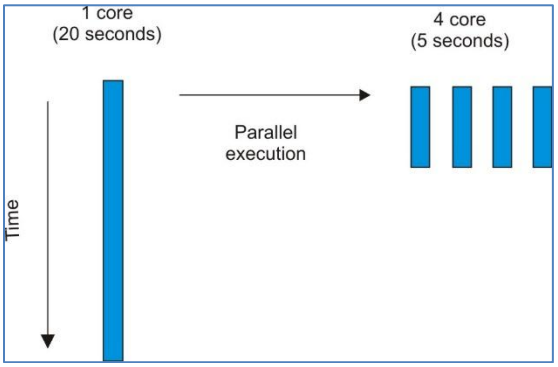

Figure 1 Parallelization reduces the execution timer

The significant difference here is that unlike tuning for serial performance, with parallelization more than one core is executing the program. So the goal must use these additional cores as efficiently as possible to reduce the execution time. Butperformance improvement is not guaranted. Depending on the application characteristics, the parallel programming model selected, the implementation of the parallelism in the application, the system software and the hardware used, one might see little or no performance improvement when adding cores. in some cases enabling more cores can degrade performance when compared to single core execution.

B. What is Parallel Programming

Parallel programming is a computer paradigm where multiple processors attempt to cooperate in the completion of a single task. The basic idea is dividing task into sub-task and executing sub-tasks simultaneously on the multiple cores. In the parallel programing paradigm, there are two memory models: shared memory and distributed memory. Shared memory is a piece of memory that can be allocated and attached to an address space. Thus, process that have this memory segment attached will have access to it. Otherwise, distributed memory has its own rule. On distributed memory processing nodes have access only to their local memory, and access to remote data was accomplished by request and reply messages[2].

C. The Basic Idea of MPI

MPI is ran on distributed memory paradigm. Several computers on the network are used to process a large amount of data. Before conducting the parallelism, the data on the server must be divided into parts and send it to a particularly processor. After processing a part of the data, particular processor will send it back to the server. By using this concept, the problem of running time can be reduce.

The MPI programming requires that the parallelism is coded explicitly by the programmer. Therefore, the programmer is responsible for analyzing and identifying the algorithm/application. The programmer must know how to conduct mechanism of send and receive the data. In addition to send and receive mechanism, communication process must be identified specifically. As a result, programming using MPI tends to be hard and intellectually demanding. However, on the other hand, properly written the MPI programming can often achieve very high performance and scale to a very large number of process[3].

D. Contribution

This research is dedicated to researcher who want to process a large amount of data which bounded by time consuming problem.The aim of this paper is to evaluate performance in term of running time of BST by using parallel programming. Several kind of strategy such as searchings and deletion problem will show the benefit of using MPI. The other main goal is also to reduce device cost. Instead of using super computer, scientist may use several computer with a reduce price.

II. RELATED WORKS

MPI has been widely used in many research field, for example in Local Area Network. It constructed distributed parallel computing based on LAN to solve the problem of scientific computing of geophysical exploration explaining. It can distribute massive complex scientific computing tasks to multiples mainframes in the LAN, and use the idle CPU resources from workstation to complete scientific computing problem cooperatively[4]. In another field, parallel programming is used to detect pattern in DNA sequence. By using parallelized paradigm, a better performance in term of running time can be achieved[5]. In BST concept, some researcher has also conducted. Kakako tree A BST with caching is added. the time complexity of BST varies between O(n) and O(log n) depends upon the number of nodes that are present in left and right children. Kakao Tree is a new data structure which is a variant of BST with some initial nodes act like caching.[6]

In this paper, numerical data will be used to show the benefit of parallel programming by using BST structure.

III. METHODOLOGY

On this paper, we will measure and analyze running time of BST using MPI. Several procedure, such as searching and deletion data is conducted to show the advantage of using parallel programming.

A. Materials

During the experiment, we used several computer for running the parallelizing. In addition to several computer, switch and cables will be used to support traffic data among the computer. One computer is set up as a server and the other as slave. Because every slave will process a part of data with their own resource, computer server will divide the data in to a part and send it to every single slave. Figure 3.1 below describe a computational model of parallel programming.

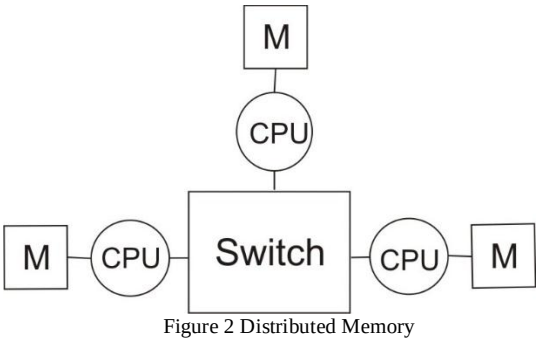

Figure 2 Distributed Memory Multiprocessor

Each computer slave will use their own memory to process the data which already send by the server. After process on each slave has been completed, output data is sent back to the server. And the process will be terminated.

B. Description of experimental scenario.

The parallel operations of BST are implemented using Linux Debian. Our experiment will be divided into two part. First scenario is to find the lowest value in the numerical data by using Domain decomposition method. Step one, computer server has to divide the data into parts and send each part to the slave. Each of computer slave will find the lowest value in its data BST algorithm and send it back to the server. Furthermore, on the computer server, all data which derived from the slave will be process again on the computer server to get the lowest value. Finally the process is terminated. figure 3 below show the procedure of domain decomposition method

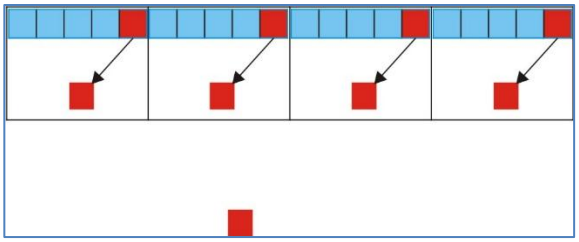

Figure 3Domain Composition method

Slightly different to the first scenario, the second scenario will find a single value in BST using the same procedure with the first scenario. It used a Domain decomposition method.

Parallelizing model used Domain decomposition model to spread all of data to the computer slave. By using Domain decomposition model, every computer slave must maintain its own data which is already sent by computer server. On each scenario, different data size will be used to show the trend of speed performance in BST. In this experiment, we used dual core processor. Therefore, each computer will have two cores for running the parallelizing. For example, if we have four computers in the experiment, we will have eight cores which can be used to process each part of the data.

The parallel operations are implemented using c++. For this experiment, we conducted two cases such as searching and finding the minimum value of the data. In each case, scanning data must be conducted at the beginning. After completing the scanning process, whole data must be divided by a total number of computer so that each core will get the same number of data to process. Then, each core will conduct insertion process with its own data. after all scanning and insertion process have done, searching and finding value can be conducted.

IV. RESULTS

The focus in this research is to know about the performance of BST in term of the running time by using parallel programming idea. In order to achieve the result, we used different number of computers in each experiment so that the differences of running time in each process can be found clearly.

In the first experiment, we will find the lowest value in a large amount of data. The target value will be placed manually in the code program. Figure 4 shows the benefit of using parallel programming.

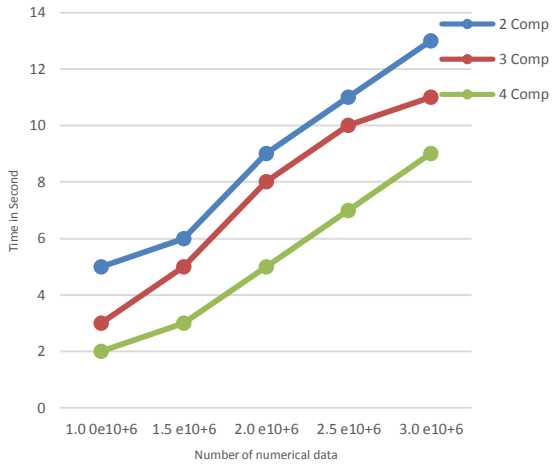

Figure 4Find the lowest value's scenario

in figure 1 shown above, The best performance of BST derived by using eight cores (four computer). If we used just two or three computers the running time becomes larger. It means that the more computer we used the better performance we have. From these research we known that by utilizing a parallel programming, we can reduce the running time of BST very significant rather than using just one computer. by using one computer, it takes more than 100 second to get the lowest value. But if we used two or three computer, the running time of BST can be reduced significantly.

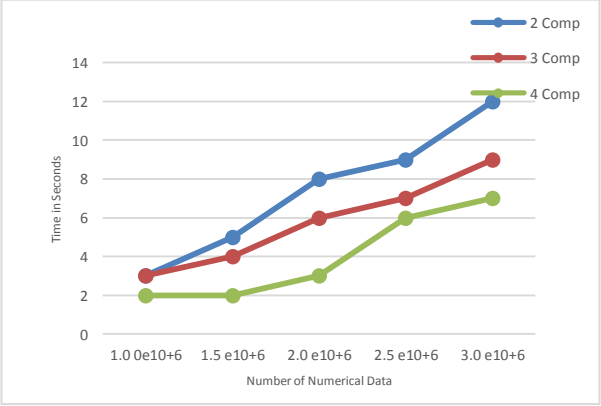

Figure 5Searching's scenario

With the same procedure, the searching process is conducted in the second experiment. We used different number of computer show the performance of BST. In a searching process, the worst case problem should be used to see the maximum process in BST. By using this idea, the changing of time speed in parallel programming can be found clearly. Figure 2 shows that by adding a number of computer continuously, the time value for the searching process can be reduced

V. CONCLUSION

This paper introduced the benefit of using parallel programming. In this research, we have shown that parallel programming can be used to increase the performance of BST in term of time speed. The best performance of BST is derived by using eight cores (four computer). If we used just two or three computers the running time becomes larger. It means that the more computer we used the better performance we have. In sharp contrast to its benefit, parallel programming can notbe used if we used it to process small data. it is happened because the time speed of BST is more affected by communication process among the computer not by its computational process.

VI. FUTURE WORKS

The next research, we plan to process DNA sequence or protein sequence by using parallel programming. It is important because DNA sequence have as many as 64 million different strings that need to be taken into account.

VII ACKNOWLEDGMENT

We are very grateful to Indonesian Ministry of Research and Technology under SINas 2014. We are also gratefuly thanks to Miss Teti and Miss Bulkis for their valuable assistance.The department of electrical engineering, which has provided the opportunity to conduct this research in Lab. Kendali Digital. Also, we would like to say thanksto LilisNugrahani, Helmibaskara and toharsagara for their support, especially in producing of numerical data.

REFERENCES

[1] Oracle, Parallel Programming with Oracle Developer Tools., 2010.
[2] Harry F. Jordan ICASE, Shared versus Distributed Memory Multiprocessor. January, 1991.
[3] Message Passing Interface Forum. *MPI : A Message-Passing Interface Standard*. 2009.
[4] Qing Guan, Jienhe Guan"*The Realization of Parallel Computing in LAN*, IEEE Journal 2012.
[5] Jian Feng, Daniel Q Naiman and Bret Cooper *"A Parallelized Binary Search Tree"*.K Inform Tech Engg 2011.
[6] Rajesh Ramachandran *"Kakkot Tree- A Binary Search Tree with Caching"*IEEE Journal 2012.

.