Parallel Processing using multiple CPU

Dr. Mahesh Kumar, Ms. Sapna Jain, Ms. Snehalata

Abstract— **currently, clusters of Personal Computers (PCs) with sharing Fast Ethernet may offer a significant price/performance ratio in parallel processing. This paper presents a study of the promising possibilities of parallelism in this environment, using both a classical numerical method to solve a differential equation and an innovative algorithm to solve the statistical problem of density estimation. To solve the large problem using a single process and a single processing is difficult task, such as main frame computers works on a lot process execution at a single unit of time using multiple parallel processing. In this paper there is a solution to break a process into small and small chunk and thread and run them simultaneously using multiple CPU as parallel processing. This emulator is based on Advanced Java.**

Index Terms— **Main frame, Process, Parallel, Chunk or Thread, Simultaneously, Reliability, Execution, Algorithm.**

I. INTRODUCTION

Telecommunication applications and network technologies are growing faster and becoming more complex to handle different types of logical traffic. Today computing and execution applications needed in various domains like business, research, Industry, e-commerce and many others. To handle a lot of data and execute then with a less unit of time using real time system is the major task. A single unit of system may occur error or delaying of execution of the information and delayed information don't have priority value. But this parallel processing of multiple task can resolve the above problems in a less unit of time with a great intensity of reliability and accuracy. This parallel computing is the simultaneously use of multiple compute resources to solve a computational problem. This thesis is about to broken the problem into discrete parts that can be solved concurrently. Each part is further broken down to a series of instructions. Instructions from each part execute simultaneously on different processor. An overall control/ coordination mechanism is employed and all the helping machines are called client machine that help to listen the process from the broadcast machine and execute the instructions and replied acknowledgment as an output.This thesis is based on Advanced Java because the java is platform independent and well suited for network and hardware. It is also a platform independent language.

Dr Mahesh Kumar, CSE,M.R.K. Institute of Engg and Technology, Rewari,,India, 9813431025.

Ms Sapna Jain, CSE, M.R.K. Institute of Engg and Technology,Rewari, India, 9215546800.

Ms Snehlata, CSE, M.R.K. Institute of Engg and Technology,Rewari,India, 8053431293.

II. GENETIC ALGORITHM FOR REPRESENTATION

Parallel computing is a form of computation in which many calculation are carried out. Simultaneously operating on the principle that large problems can often be divided into smaller one, which are then solved concurrently. There are several different forms of parallel computing. Bit-Level instruction level, Data and Task Parallelism. The Thesis emulator is prepare in advanced java. The algorithm behind the parallel processing is Java Socket programming. A socket uses a port number and the IP Addresses for client machine to broadcast messages over all client machines. An all client machine have a java library function named, LISN function then uses to listen the upcoming protocol and store temporally until the emulator executed it. After execution the client machine uses same method to return value to the server or master machine.

The difficult task is to divide the process. The process divided through the loop into the small chunk and thread. The loop control the process division. The loop initialize from the starting value and goes up to n+1. Where n= No. of client machine and another +1 shows a server machine. A server machine also execute the process but have less amount of date rather than other machine. This process shows the *OPTIMIZATION* of data and recourses. For example: For (int i=1; i<=100; i++) uses to control process and process divided as total number of instruction that is 100 and 3 client machine so that process divided as : 100/4.

III. FASTER EXECUTION AND OPTIMIZATION OF RESOURCES.

Once an instructions given as input to the emulator for parallel execution, then instructions are divided into small thread and send to the parallel processor for further processing. Parallel computers can be roughly classified according to the level at which are hardware supports parallelism with multi core and multi-processor computers having multiple processing elements within a single machine, while cluster , MPPs and grids use multiple computers to work on the same task. Specialized parallel computer architectures are sometimes used alongside traditional processors, for accelerating specific tasks.

The maximum possible speed up of a single program as a result of parallelization is known as Amdahl's law. To make faster execution and optimization there are an algorithm is constructed and implemented as a serial steam of instructions. These instructions are executed on a central processing unit on one computer. Only one instruction may executed at a time. After that instruction is finished, the next is executed.

As a optimization of resources there are uses a algorithm based that called Frequency scaling was dominant reason for improvements in the computer performance. The runtime of a program is equal to the number of instructions multiplied by the average time per instructions. Maintaining everything else constant, increasing the clock frequency decreases the average time it takes to execute an instructions. An increase in frequency thus decrease runtime for all. The mechanical phenomena is behind of this performance is P=c*V2 * F where P is Power C is captaincies being switched per clock cycle, v is voltage and F is processor frequency.

A. Proposed Algorithm:

- Prepare a socket and initialize IP addresses and Port number for the client machine in order to broadcast message and start loop for breaking process.
- Create client socket using Listen function that receive the broadcast message and send acknowledgement as an output.

B. Working Algorithm:

- Start the emulator: Server emulator for server / master machine. Client emulator for client machine.
- Create the LAN connection between then and give an IP address to each machine to identify the different machines.
- Give the client IP addresses to the master/server machine and start processing.
- Give the master machine IP addresses to the every client machine for listen protocols.
- Now give the input into the server machine and start processing.
- When server machine start then click on client emulator GUI to start working.
- If client machine start all then processing being and generate a result and client machine send the result to the server machine with a variable.
- Now server machine collect all the information as different variable and combine the result into the single entity and display.
- Stop
- Exit

When input given to the server machine then loop start to break the problem into the thread and broadcast these thread to every machine. When client machine receive they work simultaneously on given instruction and send the result to the server machine and server machine reassemble then and show result.

C. Related Work on Crossover Operator.

Basically, we have used three and four PC as experiments, but Researchers have experiments on multi PC in order to make the phenomenon of mainframe computer and super computer.

D. Experimental Results:

Parallel processing techniques for solving complex computational problems. HPC technology focuses on developing parallel processing algorithms and systems by incorporating both administration and parallel computational techniques. High-performance computing is typically used for solving advanced problems and performing research activities through computer modeling, simulation and analysis. HPC systems have the ability to deliver sustained performance through the concurrent use of computing resources. Experiments play a significant role to test a configuration of an experiment. Experiments give a judgment to discover and prepare an experience for further corrections. Experiments help to generate needs, Test Model and Test cases. The Experiments faces during this thesis are:

- Emulator Design.
- Networking or connectivity between Server and Client Machine.
- Process Division.
- Measurement of Performance and flexibility.
- Correct Execution and Test Cases.

E. Emulator Design.

Emulator is an API provide a soul to a thesis and work as a control unit, Emulator help to configure architecture of thesis and measure points of checkpoints. The J2EE design revolves around the request/response paradigm. For a login request, a user typically provides a Client id and server id to the server and waits for a response to get access to the site. A J2EE container can serve multiple users at the same time (in parallel) by managing a pool of threads, but for various reasons opening independent threads within a single J2EE container is not recommended. Some containers' security managers will not allow user programs to open threads. Moreover, if some containers allowed opening threads, then they would not manage those threads and therefore no container-managed services would be available for the threads. That would leave you to implement these services manually, which is tedious and liable to add complexity to your code.

F. Networking or connectivity between Server and Client.

Here, Experiment apply to create connectivity between server machine and client machine using a LAN cable IPv4 and preparing of IP series. Experiment apply to create socket for each and generate a listen function. Experiment measure here:

- How to connect a client machines through network
- A network is working properly in a design pattern.
- What are the performance of a Network?
- A Network is flexible or not.
- Create and activate the ports of machine.

IV. SERVER-CLIENT COMMUNICATION

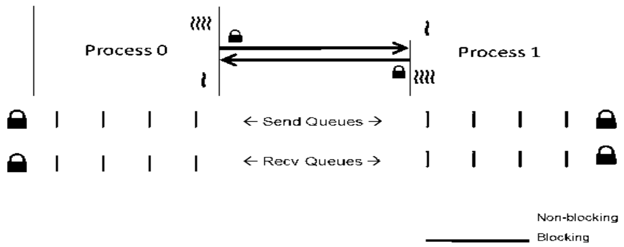

Figure 1. Server-Client Communication.

The language and rules of communication are defined in a communications protocol. All client-server protocols operate in the application layer. The application-layer protocol defines the basic patterns of the dialogue. To formalize the data exchange even further, the server may implement an API (such as a web service). [The API is an abstraction layer for such resources as databases and custom software. By restricting communication to a specific content format, it facilitates parsing. By abstracting access, it facilitates cross-platform data exchange.

A. Process Division

A parallel processing works on cluster computing. A system (Server machine) breaks an instructions into the thread and these thread broadcast to each machine and each machine listen upcoming threads and process further [4-3]. The Experiment phase occur the cases here:

- How to break a process.
- How to control a process.
- Which process division Method is used.
- Which process occur a Deadlock or not.
- .How to distribute a process to each machine.
- How to call socket and LISTEN process.
- To provide distribution if to each process.

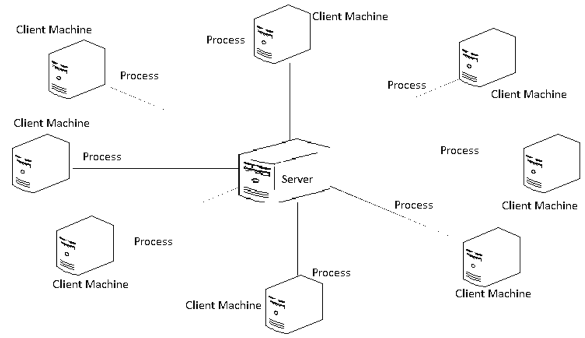

Figure 2. Measurement of Performance and flexibility.

Performance and flexibility extend the scope of work done. Experiment on Performance and flexibility measure the following cases.

- How fast a work done complete.
- Flow of data in correct and accurate format.
- What is Time Delay, Time around and Average Time?
- Complexity of Algorithm.

Parallel graph processing. Parallel algorithms have been a classical way to improve the performance of graph processing. On multi-core CPUs, parallel libraries like MTGL have been developed for parallel graph algorithms.

B. Data-driven computations.

Graph computations are often completely data-driven. The computations performed by a graph algorithm are dictated by the vertex and edge (node and link) structure of the graph on which it is operating rather than being directly expressed in code. As a result, parallelism based on partitioning of computation can be difficult to express because the structure of computations in the algorithm is not known a priori.

a) Unstructured problems.

The data in graph problems are typically unstructured and highly irregular. Similar to the difficulties encountered in parallelizing a graph problem based on its computational structure, the irregular structure of graph data makes it difficult to extract parallelism by partitioning the problem data. Scalability can be quite limited by unbalanced computational loads resulting from poorly partitioned data.

b) Poor locality.

Because graphs represent the relationships between entities and because these relationships may be irregular and unstructured, the computations and data access patterns tend not to have very much locality. This is particularly true for graphs that come from data analysis. Performance in contemporary processors is predicated upon exploiting locality. Thus, high performance can be hard to obtain for graph algorithms, even on serial machines.

c) High data access to computation ratio.

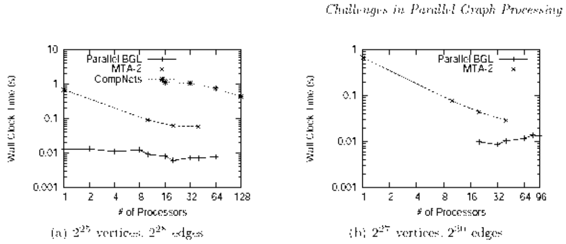

Figure 3 Challenges in Parallel Processing

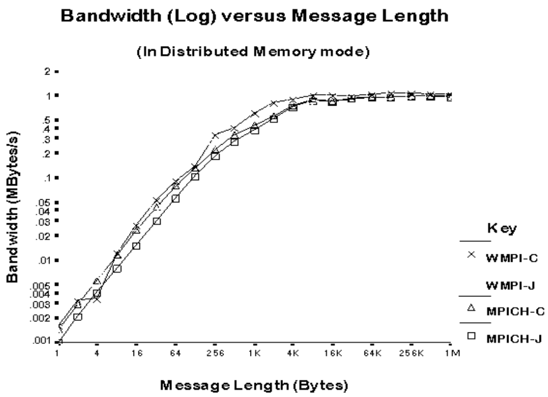

Figure 4. To Measure Performance

V. CONCLUSION

The implemented Parallel Processing system allows any research center to install and use a low cost parallel programming environment, which may be administered in an easy to use basis even by unfamiliar with clusters. Such clusters allow evaluating the efficiency of any parallel code to solve the computational problems faced by the scientific community. This type of parallel programming environments are expected to be subject to a great development efforts within the coming years, since an increasing number of universities and research centers around the world include Beowulf clusters in their hardware. The main disadvantage with this type of environment could be the latency of the interconnections between the machines. This HPCC can be used for research on object-oriented parallel languages, recursive matrix algorithms, network protocol optimization, graphical rendering etc. Also it can be used to create college's own cloud and deploy cloud applications on it, which can be accessed from anywhere outside world just with the help of web browser. Computer science and Information Technology students will receive extensive experience using such cluster, and t is expected that several students and faculty will use it for their project and research work.

The conclusion feature of Parallel Processing is:

- Parallel computing is fast.
- There are many different approaches and models of parallel computing.
- Parallel computing is the future of computing.

VI. FUTURE WORK

As computer networks become cheaper and faster, a new computing paradigm, called the Grid, has evolved. The Grid is a large system of computing resources that performs tasks and provides to users a single point of access, commonly based on the WWW interface, to these distributed resources. Users can submit thousands of jobs at a time without being concerned about where they run. The Grid may scale from single systems to supercomputer-class compute farms that utilize thousands of processors. By providing scalable, secure, high-performance mechanisms for discovering and negotiating access to remote resources, the Grid promises to make it possible for colleges and universities in collaboration to share resources on an unpriced dented scale, and for geographically distributed groups to work together in ways that were previously impossible. Additionally, the Parallel Processing can be used to create cloud applications. and give actual experience of this very booming technology to students. The advantages of cloud computing could work in the students advantage when it comes to getting hands-on experience in managing environments. Before virtualization, it would have been impossible for an individual student to practice managing their own multiple-server environment. Even just three servers would have cost thousands of dollars in years past. But now, with virtualization, it takes just a few minutes to spin up three new VMs. If a college were to leverage virtualization in its classroom, students could manage their own multi-server environment in the cloud with ease. The student could control everything from creation of the VMs to their retirement, giving them great experience in one of the hottest fields in IT.

ACKNOWLEDGMENTS

It is great pleasure for me to acknowledge the assistance and contribution of number of individuals who helped me in my project titled Design and Implementation of HPCC for Educational Purpose. First and foremost I would like to express deepest gratitude to my Guide Mrs. Sapna Jain (AP CSE MRKIET Rewari) who has encouraged, supported and guided me during every step of the Project Without her invaluable advice completion of this project would not be possible. I take this opportunity to thank our Head of Department, Prof. Dr. Mahesh Yadav and Mr. Ramesh Laur (AP & M.Tech coordinator in CSE MRKIET Rewari) for their able guidance and for providing all the necessary facilities, which were indispensable in the completion of this project. I am also thankful to the staff of Computer Engineering Department for their invaluable suggestions and advice. I thank the college for providing the required magazines, books and access to the Internet for collecting information related to the Project. Last, but not the least, I am also grateful to my friends for their valuable Comments and suggestions.

REFERENCES

[1] F. T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercube, Morgan Kaufmann, CA 1992.

[2] J. H. Rief, Synthesis of Parallel Algorithms, Morgan Kaufman, San Mateo, CA, 1993.

[3] An Introduction to Parallel Algorithms, Addison- Wesley Pub Co, Reading, MA, 1992.

[4] Robert Englander, Developing Java beans, O'Reilly, 1997

[5] Robert Eckstein and Marc Loy and Dave Wood, Java Swing, O'Reilly, 1998.

[6] David Flanagan, Java Foundation Classes in a Nutshell, O'Reilly, 1999.

[7] David Flanagan, Java in a Nutshell, second, O'Reilly, 1996.

[8] http://www.en.wikipedia.org

[9] http://www.google.com