System for Parallelism Process

Dr. P. R. Deshmukh[1], P. A. Tijare[2]
Govt. College of Engineering, Nagpur[1], Sipna College of Engineering, Amravati[2]
Email: pr_deshmukh@yahoo.com[1] ,pritishtijare@rediffmail.com[2]

Abstract-
To accomplish parallel programming in well manner, the way of balancing the load plays very crucial role. By achieving load balancing we can reduce time to execute and enhance the performance of program. This paper proposes a system for parallelism process in which systems in the network will communicate to complete large task through slicing.

Index Terms-: Parallelism, load balancing, slicing

1. INTRODUCTION

Parallelism process consists of a set of parallel processing elements which co-operate and communicate to solve large problems fast, but parallelism process is quite difficult as the parallelism process have bigger issues like:

- Transmission of data to remote nodes
- Type of inter-connection provided
- To execute data over the network
- Performance

Program slicing was firstly introduced by Weiser as a first program analysis technique [1]. In this they have shown depending upon the slicing criteria, program slicing slices the program into number of slices without disturbing the behavior of the program.

There are various types of slicing have been proposed in the literature. Some of them are static slicing, dynamic slicing [2], quasi static slicing [3], and simultaneous dynamic slicing [4], object oriented slicing and conditioned slicing [5].

Program slicing concept is a vast concept in soft computing field. The different authors have derived different slicing methods to slice the program using different flow graph and dependence graph.

2. PROCESS FOR PARALLELISM

2.1. *Connecting all Network Nodes*

The Research work runs in the network where different nodes are connected to each other. The tool designed for parallel computing works in the network. The distributed parallel computing tool developed in the research work scans the network and find nodes connected to host system.

2.2. *Program Slicing*

The given program is sliced by the “Slices” Algorithm. The Slicing Algorithm design in the research work does not use any slicing criteria but use the Slicing Point. The Program Slicing algorithm is designed for only iterative program.

2.3. *Executing Slices*

To run the slice on the remote node, new technology is designed for our work. It takes necessary input and gives required output of the program. Technology is designed to run the slice on other node from the host system.

2.4. *Integrating Result*

Proposed Technology helps the slices for execution on the remote node and creates the Result File. The Result file contains the result of each individual slice which is to be integrated on host system. To integrate the slice result on the host system, result file is read on the host system and integrated.

3. SYSTEM ARCHITECTURE

In our work, client server architecture is used. Server system is considered as a host system which is central source for communication in our system.

The system architecture for the work and different requirements for running the model is briefly described in this section. This section also describe prerequisite for the work and the model.

The work represented is designed for running with minimum resource utilization aiming to utilize the resources of client node in the network without influencing the client node.

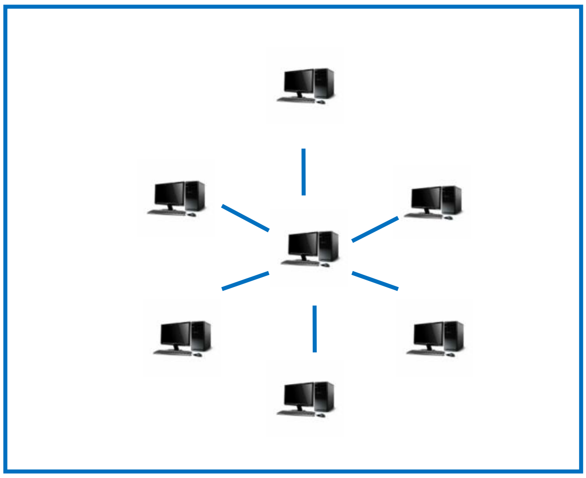

Fig 1. System architecture in proposed work

Figure 1 shows representation of architecture used in this work in which multiple nodes are connected to host system.

4. PARALLEL PROGRAMMING MODEL

To implement the parallel programming concept, it is necessary to choose right programming model. Scientific application software is built by considering the various parameters like system memory, execution time and accuracy of the result. Many developers used parallel programming to achieve this. Parallel algorithms are developed and used by many computational scientists in various fields.

To achieve the goal of Parallel Programming we need a model which is simple and acceptable to all parties involved: hardware designers, software developers, and end users. Parallel Programming model uses message passing system in which master slave configuration is used for communication between master and slave. Master system communicates with the client node by passing the message.

The message passing system has to include the following information:

- Sending Host name
- File name on Host Server
- Client Node Address
- Destination on client Node

Parallel programming model uses master-slave programming schemes in this work. As MPI programming mechanism needs to write functions in the program itself so MPI mechanism is not used. Also the concurrency of the programs used in the work does not interact with each other for execution so MPI was not a good choice. Instead of that we have used the SSH protocol to send and received the program to client node. The Program file created by the Slicing Tool does not need to interact with each other for execution.

In this work, we are creating an environment for parallel computing using the simple methods; we are not passing the messages in the program as other parallel programming methods do. The parallel environment developed in the work do not required to pass the message to the program, instead of that we are sending the file over the secured shell by using the SSH Protocol.

The SSH protocol helps in the remote login and execute the remote command which makes the environment to be controlled by the server system. Windows is one of the few modern desktop/server Operating System that does not include SSH by default. To make the SSH protocol enable in Windows Operating System we are using Bitvise SSH server which performs all the functionality of the SSH Protocol. The Bitvise SSH server supports file transfer from server to client and vice versa.

5. FILE MANAGEMENT

File management in our work indicates the naming, storing, and handling files created. Slices are created by the parallel computing tool. The task of naming the slices is done by programming. The slices are then converted to executable files by the C-Compiler, which are moved on the remote node using the PSCP command line application of the PuTTY.

The executable files which are to be transfer from the host system to remote system include the source and destination address. The executable file creates a result file of slices on the respective remote node. The host system collects the entire result file from the remote node.

6. PREREQUISITE FOR SETTING UP MODEL

To run the tool error free, some prerequisites are needed to set up the model. Slicing tool is used for Windows operating system. To run tool on Windows platform, SSH protocol is necessary to login onto remote system and execution of commands. So to achieve this, SSH server has to be implemented on all nodes in the network.

Following prerequisites are to be present.

➢ PuTTY: PuTTY is an SSH and telnet client.

- Bitvise SSH Server: Bitvise SSH Server is an SSH, SCP server, and SFTP for Windows.
- Turbo C Compiler: For Compilation of the C-Program and Program Slice (only on host system).

7. PROPOSED METHODOLOGY

The proposed methodologies have been divided into three phases where in first phase Program Slicing Algorithm for iterative program is designed and implemented. The iterative program is divided into number of slices using the program slicer developed in the research work. The slices are then redefined by adding necessary code to the slices so that able to run individually.

In second phase, the interconnection between systems connected in the network have been established using the SSH protocol in Windows System. In Windows operating system SSH Protocol is setup using the Bitvise and PuTTY. This makes the Windows operating system to connect remotely and to operate remotely. The method for shifting the slices to other nodes by using the necessary credentials of the operating system is also executed in this phase.

In the third phase, technology is developed which run the program slice on remote node. This phase also explains the merging of results on host system.

STEPS FOR PROPOSED METHODOLOGY

- Nodes in the network are connected by RJ45 cable with same or different system configuration as host system
- Scan all the nodes present in the network by network analysis function in the tool that we developed. Also collect all information of the network such as number of systems connected, IP address of each node, name of node and workload already present on the node.
- Slicing of iterative program by using program slicing algorithm (tool slices the iterative program into number of slices given by the user)
- Tool decides to which node slice has to transfer
- Execution of program slices on client nodes
- Collection of result on host system from client nodes and merge to give final result of the program

8. SYSTEM INTERCONNECTION

- For using the resource of the client node, we must know the System Name and IP Address of the system so that we must communicate with the system.
- The operating system provides a secured structure which is quite difficult to break and enter the operating system.
- To remotely use the remote node secure shell is used in windows platform.
- SSH is a cryptographic network protocol which allow remote login in the operating system.
- Application Software PuTTy is used to transfer the file on network.
- Bitvise SSH Server and SSH client are used for remote login in windows platform.

Following figure shows the interconnection between host system and other nodes in the network using PuTTY and SSH server.

CONCLUSION

This paper describes basic of parallelism process. It also describes the parallel programming model used in proposed work. The proposed methodology section describes the flow of working model step by step.

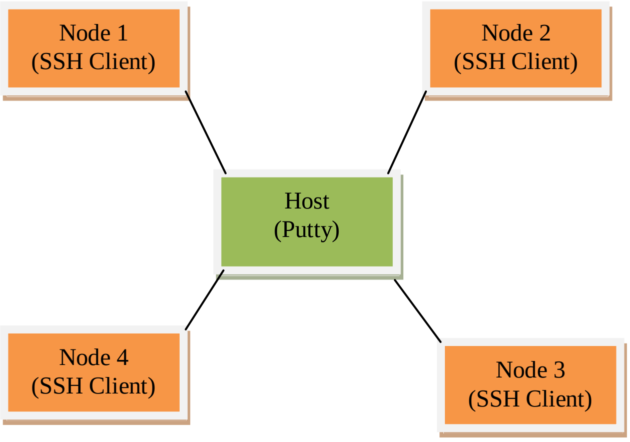

Fig 2. System interconnection using SSH and PuTTY

The main issues of the proposed work is how to interconnect the different nodes so that, they can support the computing model in the proposed work, this paper describe the basic system interconnection techniques we used in the proposed work.

REFERENCES

[1] Mark Weiser, “Program slicing”, In Proceedings of the International Conference on Software Engineering (ICSE’81), pages 439–449, 1981.

[2] B. Korel and J. Laski, (1990), “Dynamic slicing of computer programs”, The Journal of Systems and Software, vol. 13, no. 3, pp. 187-195.

[3] R.J. Hall,(1995), “Automatic extraction of executable program subsets by simultaneous program slicing”, Journal of Automated Software Engineering, vol. 2, no. 1, pp. 33-53.

[4] G.A. Venkatesh, “The semantic approach to program slicing”, ACM SIGPLAN Notices, vol. 26, no. 6, 1991, pp. 107-119.

[5] G. Canfora, A. Cimitile, and A. De Lucia,(1998), “Conditioned program slicing”, Information and Software Technology, vol. 40, no. 11/12, pp. 595-607.

□