PERFORMANCE STUDY OF CLUSTERED COMPUTERS*

Ece Yaprak
Division of Engineering Technology
Wayne State University
Detroit, MI 48201

Swamy Punyamurtula and Sumit Roy
Department of Electrical and Computer Engineering
Wayne State University
Detroit, MI 48202

Greg Follen
NASA Lewis Research Center
21000 Brookpark Rd.
Cleveland, OH 44135

KEY WORDS

Performance Analysis, Model Evaluation, Communications.

ABSTRACT

In this paper we present results of a performance study on Lewis Advanced Cluster Environment (LACE) model. Clustered computer environments play a crucial role in solving computationally intensive problems. With the help of suitable Parallel and Distributed Computing paradigms, the execution times of many such problems can be reduced significantly. In the distributed execution, the network environment of the clustered computers plays an important role. In this work, we have analyzed the LACE cluster performance in executing a simulated Computational Fluid Dynamics program from the NAS parallel benchmarks. The load on the network is monitored in real time using *NetMetrix* software. The network data can be captured with the help of the *Protocol Analyzer* feature of the NetMetrix. This data is then used to simulate the execution of the benchmark program on the LACE cluster, modeled using the *OPNET* modeler.

1 INTRODUCTION

Computer Networks have become significant tools in communications and parallel computing. Significant advances in parallel algorithms, for example, have demonstrated the potential gains from clustered environments, which seemed difficult to achieve only a few years ago. Moreover, with the increasing availability of software packages like PVM, HeNCE and Concert/C for distributed environments, it is now possible to execute many real time computationally intensive problems on these cluster computers. These advances have generated critical questions about the performance effectiveness of clustered systems. The widely varying performance levels observed when clustered systems operate under critical resource constraints, have generated interest in improving the performance range of networks. Simultaneously, interest has also emerged in the quantification of performance gains through experimentation, mathematical analysis, theory development and modeling. Better understanding of performance analysis can advance knowledge about system deficiencies and performance predication.

NASA Lewis Research Center has a cluster of 32 IBM RS/6000-560 work stations as shown in Figure 1. Each station is connected to two 10 Mbps Ethernet subnets, *public* and *private*. The *private* Ethernet is used for communication among these nodes and the *public* net connects these nodes to the Internet. The communication is established via TCP/IP protocol. Each node in the cluster has two hostnames and therefore it can access both the subnets through separate interfaces (Yaprak *et al.* 1993). The cluster is used for both parallel and batch processing. Eight of the nodes in the cluster are reserved for batch/interactive processing (node1-node8). The other twenty four nodes (node9-node32) are used for massively parallel processing (MPP) . The load on the network is monitored in real time using NetMetrix software in terms of its source, destination protocol, packet size, time, bandwidth used etc. In our work, we have studied the performance of this LACE cluster by exe-

*This work was supported by NASA through grant NAG3-1569

cuting Benchmark programs from the NAS Benchmarks suite. The Benchmark programs are modified for distributed execution with appropriate PVM constructs*. We have used the NetMetrix package to capture data that is transferred between the MPP nodes during the distributed execution of these programs.

The LACE cluster is also modeled using OPNET modeler of MIL 3., Inc. The advantage of such modeling is that we can simulate the execution of these programs and study the network behavior closely for varying loads of Ethernet traffic. The rest of the paper is organized as follows. In section 2, we introduce the NetMetrix package and describe the data capturing procedure. In section 3, we give details of OPNET modeling of the LACE cluster Ethernet. In section 4, we present the performance analysis figures for the LACE cluster Ethernet.

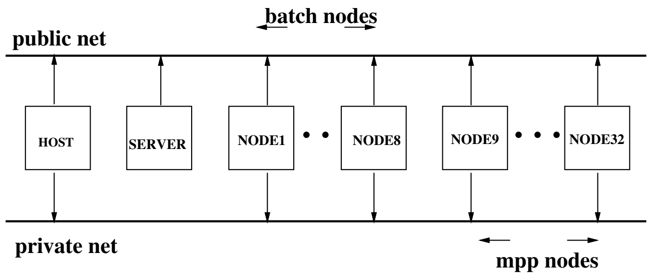

Figure 1: *The LACE cluster at NASA Lewis Research center*

2 DATA CAPTURING

The NetMetrix package from the Metrix Network Systems Inc., is very useful in monitoring the load on a network. It has many features like *load monitor*, *NFS monitor* and *protocol analyzer*. The load monitor option allows monitoring the load on both the *public* and *private* subnets in terms of time, source, destination, protocol and packet size. It also gives the number of packets seen between any two communicating stations, the number of network errors and the percentage bandwidth used for each interface. The NFS monitor option can be used to monitor the NFS load on a network to which a file server is connected. It gives the NFS response time, the number of retransmits and the number of errors. It also characterizes the load in terms of time, file server, workstations used and the type of NFS procedure.

*The authors are thankful to Dr. Chris Cox of Clemson Univ.,/NASA Lewis research center for providing the PVM version of these programs.

The protocol analyzer feature of the NetMetrix is useful in capturing and examining individual packets. With the filtering option, a group of nodes can be specified and the traffic to or from these nodes can be observed. It displays information about the packet length, time of transmission, source and destination nodes and the protocol used. The protocol analyzer automatically disassembles packets belonging to a number of protocols suites including TCP/IP, SUN RPC etc. From these packets we can obtain the protocol header information. This is useful in determining the actual size of the data transmitted and the delays involved in the transmission. We can select the protocol layer and obtain the information particular to that layer.

2.1 Program Execution

We have selected a *Computational Fluid Dynamics (CFD)* application program from the suite of NAS parallel Benchmark programs (Bailey *et al.* 1991). In this benchmark program, multiple independent systems of non-diagonally dominant Block Tridiagonal (BT) equations with a 5x5 block size are solved. We have used a modified version of the original code with proper PVM constructs for distributed execution. The PVM allows the user to configure the virtual machine by specifying a set of hosts through a *hostfile*. We have selected the 24 MPP nodes (node9 to node32) in the LACE cluster for distributed execution. The BT program can then be executed on this virtual machine. The individual hosts communicate through messages and the special PVM constructs in the code take care of the message passing constructs. Using the NetMetrix package we can capture and analyze these messages. The protocol analyzer can be started at the same time as program execution. The messages exchanged between the hosts as part of the computation gives us an idea about the communication overhead involved in the distributed execution. This data can be captured and stored in a file for further analysis.

In the following section we describe how the modeling of LACE cluster using the OPNET package is described.

3 OPNET MODELING of LACE NETWORK

3.1 Network Model

The base LACE network consists of 33 workstations, of which one acts as the server. The next 8 stations are used for batch-processing and the remaining 24 are used for parallel processing. A *private* Ethernet network connects these 33 workstations with each other. Another *public* Ethernet network connects the 33 stations to other machines.

In the simulation, each station consists of two independent Ethernet interfaces. These stations generate messages, create Ethernet frames and transmit them on the network if there is no collision. Else, the messages observe the back-off and retry protocol (*IEEE 802.3 Standard*). Two nodes, *sink* and *source*, represent the rest of the public network and communicate with the nodes of the LACE network. These nodes can communicate only via the public network.

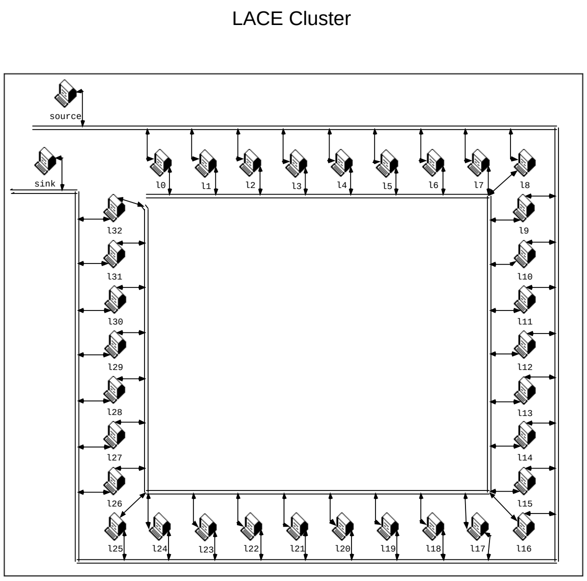

Figure 2: *OPNET model of the LACE cluster*

The network is modeled using the OPNET Modeler. Through its hierarchical structure, OPNET allows a step by step design of any network. It provides editing environments for the design of different functional/logical levels within the network. The structure of a network typically consists of a Node level and a Process level. At node level, OPNET supports the design of specific communicating entities (nodes). A node in general is the computer plus its network interface. This node will have both software and hardware components that support data management functions like creation, transmission, reception, storage etc. The hardware and software components which give the node the above data handling capabilities, form the process level. With the help of OPNET *process models*, the logic of many real-world processes like communications protocols and algorithms can be described using state transition diagrams (STD). The process model operations can be coded in the C language. At each level, appropriate model attributes can be specified. At the network level, the nodes are interconnected via links. All these hierarchical models can be compiled together to form a executable simulation code in C language, which will be helpful in analyzing the behavior and performance of the network.

OPNET also has other useful features like simulation probe editors and performance analyzers. Its probe editor allows us to specify the parameters to analyze the performance and the analysis tool shows the performance curves.

3.2 Assumptions

Certain assumptions were made regarding the traffic generated by the various nodes for our simulation purposes.

- The batch–processing nodes do not generate any traffic on the *private* network. They generate only a low volume of messages to all other nodes. These messages are small in size and are generated at a very low rate, which would represent activities like *e-mail* messages, occasional *ftp sessions* or *rlogins.* Thus their contribution to the overall *public* traffic is minimal

- The parallel processing nodes communicate with each other exclusively over the *private* network. The traffic rate as well as the message length could be different, depending on the program being run. However a program generating large messages would do so only at a low rate, and similarly high traffic would indicate that the messages are quite small.

- The pattern of communication on the *private* network is one of a complete exchange i.e., all parallel processing nodes communicate with each other.

- The parallel processing nodes generate traffic consisting of very large packets for the server node.

This represents an initial data distribution to the parallel processing nodes, or periodic collection and updating of data that was being processed in parallel. This traffic uses the *public* network and the rate at which it is generated is low, compared to that on the *private* network.

4 PERFORMANCE ANALYSIS

In this section we present the results of our simulation. We have simulated the execution of the BT program on the LACE cluster. As explained in the previous section, the load on the *private* Ethernet of the LACE cluster is captured using the Protocol Analyzer feature of the NetMetrix package. This load is nothing but the data communicated between the MPP nodes while executing the BT program. From this data, we can calculate the arrival rate of packets to or from a node and the mean packet size. This information can then be used to simulate the execution of the BT program on the OPNET model of the LACE cluster.

Every node generates packets of random size with specified mean packet size and at the given rate. This packet is then transmitted to a destination node which is selected randomly following a uniform distribution. It is of interest to study the performance of the network under this scenario. In the following subsection we describe the parameters that are studied also present the performance curves.

4.1 Simulation Parameters

The main parameter measured in this study is the end–to–end delay experienced by the Ethernet packets. This indicates essentially the latency due to the Ethernet protocol. The software used also allows measurement of the *Transmitter thruput*, *Average transmitter delay*, *Average MAC delay* etc. The packet destinations are generated with a uniform distribution over a *low* and *high* address. The packet size is specified in terms of its *mean packet length*. The traffic rate can also be controlled by the *arrival rate* parameter.

4.2 Simulation results

In the simulated LACE cluster Ethernet model, the node level consists different sub process modules such as source, sink, MAC, bus transmitter, bus receiver and difference units. The media access protocol (MAC) of the Ethernet provides efficient and fair sharing of the communication channel which is the bus connecting the stations of the LAN. This MAC accepts data packets from a higher layer protocol like IP and attempts to transmit them at appropriate times to other stations on the bus. When several stations attempt to transmit simultaneously, "collision" occurs and it results in loss of data packets. Ethernet MAC has special mechanisms like "carrier sensing" and "collision detection" to deal with such situations. So, it is interesting to study the performance of any Ethernet model in real time data communication scenario.

In our work, we have generated a stream of data packets to simulate the execution of the BT program on the OPNET modeled LACE cluster Ethernet. The performance figures of the Ethernet for this execution are given in Figures 4 and 3.

In Figure 3 we have shown the average transmitter delay and the MAC delay. These figures show the average delay involved in data communication due to the *deferring* activity. Every time the MAC finds the bus to be busy it backs off and after a random duration it attempts to transmit again. This deferring causes delay at the transmitter level as well as at the MAC level. In both the figures we can see that even though these delays are initially small, they grow large when the size of the data packets becomes large.

In Figure 4 the average end-to-end delay is shown. It can be observed that during the initialization of the PVM processes the end-to-end delay is less because of the small data packet size. Once the communicated data size grows the latency also becomes high.

5 CONCLUSION

In this paper, we have presented results of a performance study on Lewis Advanced Cluster Environment (LACE) model. A Computational Fluid Dynamics problem execution was simulated on the OPNET modeled LACE cluster Ethernet and its performance figures are given.

REFERENCES

Bailey D., Barton J. et.al., (Eds), "The NAS Parallel Benchmarks", Tech. Rep. RNR-91-002 revision 2, NASA Ames Research Center, August 1991.

Yaprak E., Sanz J., Follen G. and Holbert R., "LEWIS Advanced Cluster Environment Performance Study" in *Proceedings of International Simulation Technology Conference*, pp. 23 to 28, San Francisco, Nov., 1993.

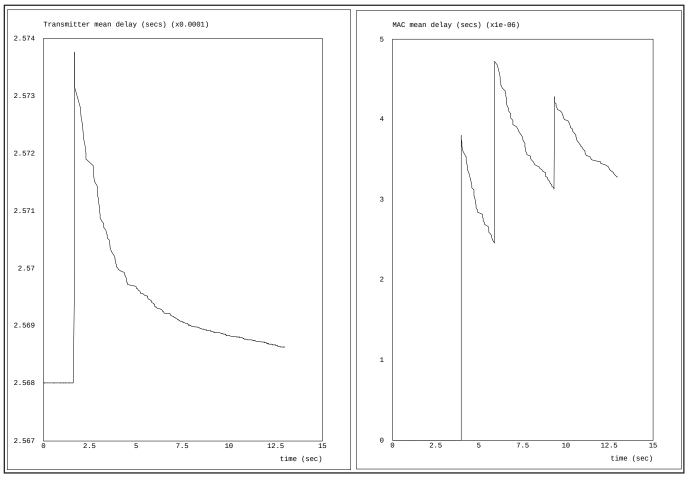

Figure 3: *Transmitter mean delay and MAC mean delay*

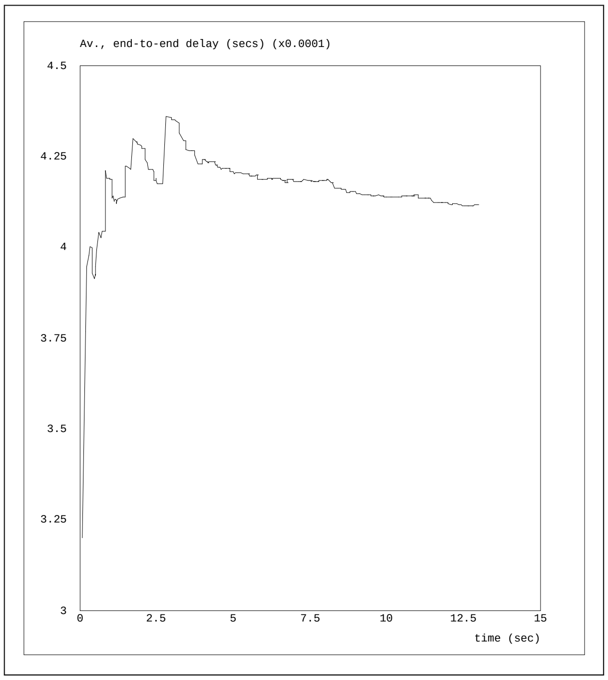

Figure 4: *Average end–to–end delay*