Performance Evaluation of Network based Distributed Supercomputing Environment

OP Gupta[1], Karanjeet Singh Kahlon[2]
opgupta@gmail.com karankahlon@yahoo.com

[1]Faculty of Computer Science, Punjab Agricultural University,Ludhiana, 141004 India
[2]Department of Computer Science, Guru Nanak Dev University, Amritsar, India

Abstract: - In the past decade, supercomputing has witnessed a paradigm shift from massively parallel supercomputers to network computers. Though dedicated high end supercomputers still have their place in the market yet combined unused CPU cycles of desktop PCs available in the campus network can form comparable virtual supercomputers. Consequently, Parallel Processing in a network of PCs are attracted a boost of attention and becoming one of the most promising areas of large scale scientific computing. In this paper, we are presenting Grid-enabled PC Cluster (GPCC), exhibiting low latency and bandwidth scalable sub-communication system. The design of the GPCC is such that it keeps in view the socket buffer size of local and non-local nodes in the network environment. The design is relatively easy to use, inexpensive to apply and extremely accurate. The highly accurate results provided by TCP/IP ping-pong were coupled with parallel matrix multiplication benchmark. Parallel Matrix Multiplication (PMM) performance benchmark is used to test the GPCC for node-to-node network performance and parallel floating point performance of all involved processor in a local and non-local cluster environment. PMM benchmark is developed on the basis of master-slave model using dynamic distribution scheme.

Key words: Distributed Computing, Parallel Computing, Grid Computing, Local Area Network

(Received September 23, 2006 / Accepted January 03, 2007)

1. Introduction

The demand of high performance computers is always a hunt for scientist and researcher in bio-technology and in allied areas. Parallel computing is the area, which is becoming prominent to achieve the supercomputing for complex problems. A number of research groups in universities and industry are building efficient communication hardware and software for parallel computing on the network of PCs. The reason for such an interest is because of the price /performance advantage of the network of PCs in comparison to super-computers. The other reason, which pushed this technology, is that switched based LANs improved a lot in late nineties. It is worth while to harness the idle cycles of computers available on the network. In LAN, latency involved in sending the data from one node to other node depends on the software overhead of the message passing library, called startup cost. By reducing the latency, the parallel computation time can be reduced drastically. The tuning of TCP/IP [1] for sending and receiving the message for local and non-local nodes is kept under BDP (Bandwidth –Delay Product) so that there should be no congestion in the network. Ping-pong benchmark is used for standardizing the message size to be used for communication over the network and clubbed with BDP measurement for effective communication for fine grain to coarse grain applications.

2. Design and Communication of GPCC

The environment is implemented using DeskGrid API and DLL files [2]. DeskGrid is a full implementation of communication over TCP sockets Microsoft Win32 Platforms. This feature opens up the possibility of utilizing resources commonly excluded from network parallel computing systems such as Macintosh and Windows based PCs. For the experiment, DeskGrid communication library is loaded on the server and the contributor is loaded on the each local and non-local PCs in the local area network (LAN) in the background with high priority. It forms a grid of computers. The submitters are located any where in the LAN and submit

the jobs to server. Tasks are managed by background daemon which is resident on each node of the grid enabled cluster. The daemon communicates with each other using TCP protocol. The message is kept to the packet size [4] of 4k keeping in view the BDP (Bandwidth Delay Product) value calculated based on the Windows TCP Buffer size of 48K. The buffer size is set dynamically in the benchmarking software so that the congestion in the network is avoided for coarse grain applications.

For implementing the parallel applications in LAN, a set of four best suitable computers , whose performance are identical, from the grid is taken as a PC cluster for executing the sub-tasks (jobs) as shown in the figure 1.

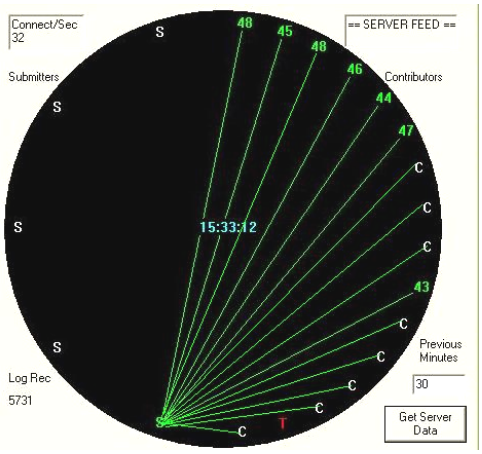

Figure 1. Grid enabled PC Cluster

Moreover, we assume that at any given instant only one parallel program is in execution on the cluster, and that the main memory of each desktop system is large enough to accommodate the working set of the parallel process it executes. Finally, we assume that the communication network carries only traffic generated by the desktop PC in the cluster (both by the parallel program and from jobs executed by other desktop PC). The communication protocol is shown in the figure 2.

2.1 Communication Performance

The communication performance [3] of GPCC is tested with Ping-pong benchmark and message size to the tune of 4k is standardize for the calculation of optimum size of TCP window socket so that network congestion can be avoided for the local and local nodes in the grid enabled PC cluster.

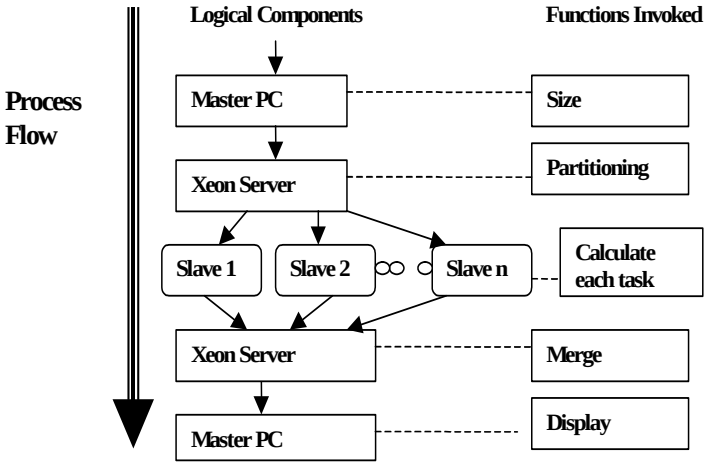

Figure 2. Communication Protocol

	TCP Window Size			
No. of Hops	16kB (Default)	32kB	48kB	Congestion if BDP >=TCPWinSize
	BDP (kB)	BDP (kB)	BDP (kB)	*means Congestion
0	8.4	8.2	7.8	No
1	16.3*	15.28	16.30	No – Size to be =32kB
2	20.8*	24.0	22.34	No –Size to be =32kB
3	24.7*	32.2*	38.8	No – Size to be = 48kB

Table 1. BDP Measurement

From the table 1, It can be concluded that we can make use non local nodes of network of PCs up to 3rd level for parallel computing without having any hindrance of the bandwidth choking. This can set statically by the network administrator or can be set dynamically in the parallel application as we did in the benchmarking software. The design of GPCC communication subsystem is low-latency and scalable in nature. Our performance evaluation shows that it effectively delivers low latency for small messages and high bandwidth for large messages.

3. Benchmarking the GPCC

The developed network design is tested with Parallel Matrix Multiplication algorithm for node to node communication when the cluster is formed from local nodes and non local nodes in the LAN. The statiscal analysis is done to study the effect keeping the buffer size comparable to BDP measurement.

The important aspect of using MM as performance evaluation tool is

- The workload of the MM can easily be changed from fine grain to coarse grain granularity
- It is not only simple to understand but also based on Linear Algebra Kernel.
- A no. of MM algorithms is available for network of workstations.
- It is scalable in nature.

The parallel algorithm for installed network of PCs has two main characteristics.

- It is based on Master-Slave paradigm
- Equal workload distribution

Algorithm

Step 1.Master (Head Node) reads data from user
Step 2.Master decompose the matrix A into multiple rows
Step 3.Master broadcasts dynamic allocation of matrix B columns to slaves
Step 4.Master sends respective parts of first matrix to all other processes.
Step 5.Every process performs its local multiplication.
Step6. All slave processes send back their result.
Step 7.Master (process 0) reads data and merges them.

4. Easy Deployment and Observations

Parallel Matrix Multiplication (PMM) is developed using visual C++ and batch of jobs (ASCII file) is created and submitted via calls to DeskGrid dynamic link library for communication over the network. The dynamic link library (.DLL) provides the following API that PMM used in the networked of PCs.

```
extern "C"
{extern DESKGRIDDLL_API int
DeskGrid_submitJob(char * sessionID, char *fileName,
char *response);
 extern DESKGRIDDLL_API int
DeskGrid_submitJobFromTo(char * sessionID, char
*fileName, int from, int to, int maxSeg, char *response);
 extern DESKGRIDDLL_API int
DeskGrid_getJobStatus(char * sessionID, char
*response, int *suggestedWaitMilliSecs);
extern DESKGRIDDLL_API int
DeskGrid_cancelJob(char * sessionID);
extern DESKGRIDDLL_API int
DeskGrid_fetchFile(char *fileName, int jobNum);
}
```

Any application can be distributed using the job template file by defining the input and output files, without doing the programming.

DG scope as shown in figure 1 is used to observe the activities of the GET and PUT operations of the master PC with the contributors of the cluster. It provides the replay action of submitter/job/contributor so that improvement in the GET and PUT operations is made. It maintains a server log and data can filtered based on the job ids before display. The state of the contributor is color coded: yellow (downloading files), green (executing), and cyan (uploading results). The executing contributors display the job segment they are working on.

5. Computational Performance

The network latency needs to be measured when multiple local and non local machines are used in the experiment. We run the PMM on four number of networked PCs and the size of the problem was kept fixed i.e. 3000 x 3000. The performance of GPCC is checked while local and non-local nodes are added to the cluster from the grid of computers. The local computers constitute from the level I network and non-local computers constitute a set of computers from level II or Level III network. The data collected for these types of cluster from the networked PC is shown in the table1, table2 and table 3.

No of Nodes	Computation time (sec.)	Speedup
1	6.78	1
2	4.59	1.477124
3	3.45	1.965217
4	3.2	2.11875

Table 1. Speedup with Buffer 16KB

No of Nodes	Computation time (sec.)	Speedup
1	6.98	1
2	5.9	1.183051
3	5.78	1.207612
4	4.8	1.454167

Table 2. Speedup with Buffer 16KB & Non-local nodes

No of Nodes	Computation time (sec.)	Speedup
1	7.1	1
2	5.84	1.215753
3	3.9	1.820513
4	3.4	2.088235

Table 3. Speedup with buffer 48KB & Non-local nodes

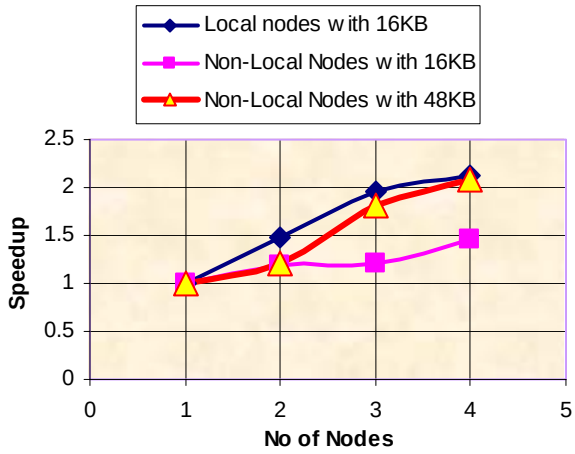

Figure 3. Comparison of Local and Non-Local Nodes

6. Conclusion

It is concluded from the graphical presentation that parallel computing in a grid of PC cluster is greatly influenced by the communication parameters of TCP/IP. To achieve the performance of non-local nodes as that of local nodes in a cluster as shown in the figure 3, it is necessary to keep the socket buffer size under BDP measurement so that network congestion has no impact on the transfer of data. From the analysis, we can conclude that the performances of the two clusters are almost identical with the present setup of network design.

Further research and study is open for characterizing granularity (no. of sub-tasks) of the parallel application and modeling for these types of clusters can be studied.

References

[1]Dave MacDonald and Warren Barkley,(2000) "Microsoft Windows 2000 TCP/IP Implementation Details", White Paper, Microsoft Corp., 2000. Retrieve June 27, 2006, from http://www.microsoft.com/technet/itsolutions/network/deploy/depovg/tcpip2k.mspx

[2] John F. Doyle (2005) “DeskGrid – A framework for distributed processing computing”, 2005. Retrieve April, 2005, from http://www.deskgrid.com

[3] R. Zamani and A. Afsahi (2005), “Communication Characteristics of Message-Passing Scientific and Engineering Applications”, **P**roceeding Parallel and Distributed Computing and Systems (PP 466), 2005.

[4] Z. Nedev, T. Gong, and B. Hill, (2004), “Optimization Problems in the Implementation of Distributed MergeSort on Networked Computers”, Proceeding Parallel and Distributed Computing and Networks (pp-420), 2004.