PERFORMANCE EVALUATION OF PARALLEL APPLICATIONS USING MESSAGE PASSING INTERFACE IN NETWORK OF WORKSTATIONS OF DIFFERENT COMPUTING POWERS

RAJKUMAR SHARMA

Computer Centre, Vikram University
Ujjain, MP, India

PRIYESH KANUNGO

Patel College of Science & Technology
Indore, MP, India

MANOHAR CHANDWANI

Institute of Engg. & Tech., Devi Ahilya University
Indore, MP, India

Abstract

A network created from so called commodity hardware can effectively and affordably be used for high performance computing. Major attraction from such networks is cost-effectiveness due to efficient utilization of existing hardware. By combining processing powers of many scattered workstations, a large computational problem can be solved within desirable time constraint, which is comparable to that of a proprietary or dedicated parallel computer. Popularly such systems are known as Cluster or Network of Workstations (NOWs). Among the fastest computers in the world, many of them have been developed by using the same technology. In this paper, we evaluate performance of parallel applications on cluster of nodes having different computing powers in terms of hardware attributes/parameters. We also investigates impact on execution time after running parallel application on selected favorable nodes, in terms of computing power and compare it with performance of application executed on random nodes.

Keywords: Execution Time; Commodity Hardware; High Performance Computing; MPI.

1. Introduction

A distributed computing system is a collection of processors interconnected by a communication network in which each processor has its own local memory and other peripherals, and the communication between any two processors of the system takes place using **message passing** technique over the communication network. Together, a processor and its resources are usually referred to as ***node*** or ***site*** or ***machine*** or ***workstation*** of the distributed computing system.

In contrast to the parallel systems (tightly coupled), the processors of distributed computing systems (loosely coupled) can be geographically distributed to cover a wider area. In parallel systems, the number of processors is usually small and limited by the bandwidth of the shared memory. On the other hand, distributed computing systems can be easily expanded and can have an almost unlimited number of processors. If we consider Flynn's classification of parallel computing, as shown in Fig. 1, all architectures except MIMD have given better performance in their contemporary time [2,3]. But nowadays almost all popular distributed systems are based on MIMD architecture. Network of Workstations (NOWs) fall under distributed memory of MIMD architecture.

2. Related Work

Load distribution is key issue in exploiting resources in a cluster. Various techniques have been considered in the literature for load distribution among workstations of a cluster. Some applications evenly distributes the load among workstations with job migration that occurs only at the end of predefined intervals [4]. But even distribution of tasks among workstations always leads to much variation in completion time at different processors due to varying computing capabilities. Performance of Dual Core and Core 2 Duo Processor was studied by various computer scientist for object oriented programming [6]. Some changes in the programming language are suggested to exploit processing power, which require huge efforts. An I/O-aware load balancing scheme has been proposed to improve overall performance of a distributed system by [5]. In addition to provide a remedy for I/O load imbalance on nodes, the scheme also keeps track of CPU and memory load sharing. But this is achieved at a very high communication cost. [1] studies impact of running a parallel program on a network of workstations and associated delays to interactive users. Immediate job migration is suggested to provide relief to sequential programmer, which require preemptive job migration strategy and high communication cost.

3. Methodology

3.1. *Load distribution scheme*

In NOWs, a parallel application is run by distributing processing tasks among the processors. In almost all load distribution techniques found in literature, processing tasks are distributed among processors randomly, i.e., without considering the computing capability of individual workstations. The impact on execution time due to varying computing powers of workstations was previously studied by the authors[7]. It was observed that workstation's hardware parameters like clock speed of processor, cache memory, primary memory, hard drive capacity and RPM etc. contribute to variation in execution time of application. In the present paper, we prepare hardware profile of workstations, as shown in Table 1, and analyze impact on execution time by running a parallel application on different combination of workstations.

Table 1: Hardware profile of workstations

Node No.	Processor	Cache Memory	Front Side Bus	RAM	Hard Disk Drive	
					Capacity	RPM
1	Intel P-4 2.0 GHz	512 KB	400 MHz	128 MB	40 GB	5400
2	Intel P-4 2.4 GHz	1 MB	533 MHz	256 MB	40 GB	5400
3	Intel P-4 HT 2.8 GHz	1 MB	800 MHz	256 MB	80 GB	5400
4	Intel P-4 HT 2.8 GHz	1 MB	800 MHz	1 GB	160 GB	7200
5	Intel Dual Core 2.0 GHz	1 MB	800 MHz	512 MB	80 GB	5400
6	Intel Dual Core 2.0 GHz	1 MB	800 MHz	1 GB	160 GB	7200
7	Intel Core 2 Duo 2.2 GHz	2 MB	800 MHz	2 GB	320 GB	5400
8	Intel Core 2 Duo 2.93 GHz	2 MB	1066 MHz	2 GB	500 GB	7200

3.2. *Cluster environment*

We performed the experiments on cluster of eight nodes having different computing power, connected with a ethernet 10/100 IEEE 802.3 switch, as shown in Table 1. To run parallel application on the cluster, a Message Passing Interface (MPI) Library is needed to establish point-to-point and collective communication among the processors. We use MPICH2, a open MPI library, for running our applications. MPICH2 is a high performance implementation of the message passing interface standard : MPI-2. We solve problem of numerical integration by 'Trapezoidal Rule of Integration' by running parallel program on the cluster.

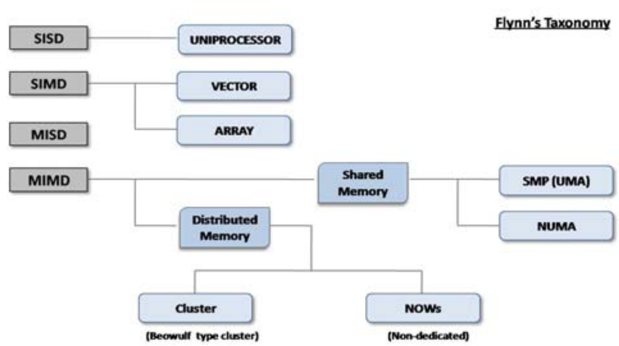

Fig. 1 : NOWs fall under MIMD architecture

3.3. *Trapezoidal rule*

The general problem of numerical integration is to find an approximate value of the integral

$$I = \int_a^b f(x)dx \tag{1}$$

by Trapezoidal rule, we divide the interval [a,b] into N subintervals, each of length $h = (b-a)/N$. We denote the subintervals as $(x_0, x_1), (x_1, x_2), \ldots\ldots (x_{N-1}, x_N)$, where $x_0 = a$, $x_N = b$ and $x_i = x_0 + ih, \quad i = 1 \quad to \quad N-1.$ We write

$$I = \int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \ldots\ldots + \int_{x_{N-1}}^{x_N} f(x)dx \tag{2}$$

Evaluating each of the integrals on the right hand side of (2) by the trapezoidal rule, we get

$$I = \frac{h}{2}[(f_0 + f_1) + (f_1 + f_2) + \ldots\ldots + (f_{N-1} + f_N)]$$

$$= \frac{h}{2}[\ f_0 + 2\ (\ f_1 + f_2\ \ldots\ldots + f_{N-1}) + f_N\ ] \tag{3}$$

4. Experimental Results

We form different groups of five nodes out of eight nodes, as shown in Table 2, to compare the performance of each group. Some groups consist of nodes having more computing power than all other nodes and some groups consist of nodes having less computing power than all other nodes. A combination of both types (more powerful and less powerful) of nodes are kept in remaining groups.

Table 2 : Combination of Nodes in Different Groups

Group Id.	**Nodes in Group**				
G 1	Node 1	Node 2	Node 3	Node 4	Node 5
G 2	Node 2	Node 3	Node 4	Node 5	Node 6
G 3	Node 2	Node 3	Node 4	Node 6	Node 7
G 4	Node 3	Node 4	Node 5	Node 6	Node 7
G 5	Node 3	Node 4	Node 5	Node 7	Node 8
G 6	Node 4	Node 5	Node 6	Node 7	Node 8

We run the parallel program for Trapezoidal Rule many times on each group separately and find the mean execution time as shown in Table 3.

Table 3 : Execution Time for Different Groups

Group Id.	**G 1**	**G 2**	**G 3**	**G4**	**G 5**	**G 6**
Execution Time (ms)	374	352	339	304	287	260

We observe that the execution time is decreasing for the groups with the nodes having higher computing powers, as shown in the Fig. 2.

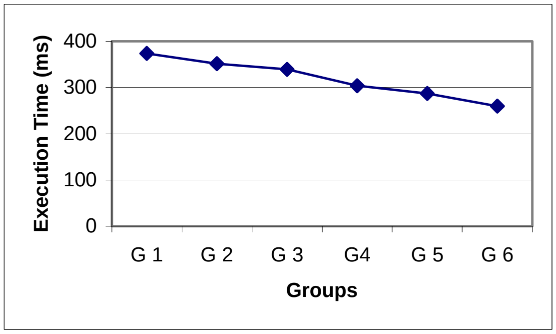

Fig. 2 : Comparative Execution Times for Different Groups.

5. Conclusion

The analysis of experimental results shows that the performance of a parallel application over a cluster improves, if different tasks are assigned to nodes having higher computing power rather than simply distributing tasks to random nodes. If nodes having higher computing powers are busy than tasks should be assigned to nodes in decreasing order of their computing power. In our future research, we shall consider development of algorithms to incorporate hardware parameters describing the computing power of a node and distribute the load towards the favorable nodes to minimize the overall response time of an application.

References

[1] Remzi Arpaci, Amin Vahdat, Thomas Anderson, David Patterson, "Combining Parallel and Sequential Workloads on a Network of Workstations," *Technical Report : CSD-94-838, University of California, Berkeley,* 1994.

[2] Armando E. De Giusti, Marcelo R. Naiouf and Laura C. De Giusti, "Dynamic Load Balancing in Parallel Processing on Non-Homogeneous Clusters," *Journal of Computer Science and Technology* (*JCS&T),* Vol.5, No.5, December 2005.

[3] Jawinder Pal Singh, "Dynamic Load Balancing for Cluster Computing," Retrieved from: *www5.in.tum.de/lehre/seminare /clust_comp/SS05*, 2008.

[4] Helen D. Karatza, "A Comparison of Load Sharing and Job Scheduling in an Network of Workstations," *International Journal of Simulation*, Vol. 4, No. 3-4, pp 84-87, 2003.

[5] Xiao Qin, Hong Jiang, Yifeng Zhu, David R. Swanson, "A Dynamic Load Balancing Scheme for I/O-Intensive Applications in Distributed Systems," *International Conference on Parallel Processing (ICPP),* October 2003.

[6] Vipin Saxena and Deepa Raj, "Impact of Dual Core on Object Oriented Programming Language through UML," *Internation Journal of Advanced Networking and Applications*, Vol. 01, Issue 02, pp. 125-130, 2009.

[7] Rajkumar Sharma, P. Kanungo and M. Chandwani, "A Comparison of Computation Intensive and Data Intensive Problems on Heterogeneous Nodes in Network of Workstations," *IEEE National Conference Proceedings on Education & Research in Information Tecnology*, March 6-7, Guna INDIA, pp. 291-298, 2010.