

Handling of Congestion in Cluster Computing Environment Using Mobile Agent Approach

Sumit Mittal¹ and Dr. P.K. Suri²

¹ M.M. University, Mullana, Ambala, Haryana, India

Received: 9 April 2012 Accepted: 1 May 2012 Published: 15 May 2012

Abstract

Computer networks have experienced an explosive growth over the past few years and with that growth have come severe congestion problems. Congestion must be prevented in order to maintain good network performance. In this paper, we proposed a cluster based framework to control congestion over network using mobile agent. The cluster implementation involves the designing of a server which manages the configuring, resetting of cluster. Our framework handles - the generation of application mobile code, its distribution to appropriate client, efficient handling of results, so generated and communicated by a number of client nodes and recording of execution time of application. The client node receives and executes the mobile code that defines the distributed job submitted by server and replies the results back. We have also analyzed the performance of the developed system emphasizing the tradeoff between communication and computation overhead. The effectiveness of proposed framework is analyzed using JDK 1.5.

Index terms— Cluster Computing, Mobile Agent, Congestion.

Literature Survey : A number of research efforts in the area of improving the performance of distributed applications in a cluster computing environment have emerged [7] [8].

Martin, Vahdat, Culler and Anderson [9] revealed that the communication latency, overhead, and bandwidth can be independently varied to observe the effects on a wide range of applications. They showed that applications demonstrate strong sensitivity to overhead, slowing down, when overhead is increased. Lange, D.B [10] discussed the impact of cluster size and underlying network on the performance of distributed applications.

Our research effort (i) evaluates the performance of application as a function of problem size, network parameters and cluster size (ii) provides features for monitoring of cluster. The developed MCLUSTER model concentrates on reducing the communication overhead by using the mobility of code and bridging all system characteristics. There are many cluster configurations, but a simple architecture is shown in the Figure 1. Their abilities to cope with system heterogeneity and to deploy user-customized procedures at remote sites are well suited for cluster computing environments [12]. By interacting with a remote host after migrating to it, an agent can perform complex operations on remote data without transferring them, because the agent can carry the application logic to where it is needed. M status of each slave node. A slave node within a cluster provides the cluster a computing and data storage capability. These nodes are derived from fully operational, standalone computers that are typically marketed as desktop or server systems that are off-the-shelf commodity systems.

The framework consists personal computers, high speed communication network, sequential and parallel applications as shown in Figure 2.

PC's are connected to the network using Ethernet Network Interface Card. Cluster middleware is implemented in JAVA so that middleware can provide the single system image of the cluster to any computer with different OS platforms, once the JAVA virtual machine is installed. JVM makes it easier to implement, migrate and execute the mobile code at remote computer in the cluster. The user is guided through the creation and management of cluster via GUI. It frees the user from identifying the network topology of the framework of cluster. The

framework has been designed in such a way that incremental changes to it can easily enhance the generality and usability of cluster. AppManager coordinates all functions related to application description, distribution and execution. AppManager provides a GUI from which user can easily select a particular application, provides the input data and specify the parameters such as number of nodes to be used for the execution of application. Once the application is selected, then the AppManager will divides the data range into the number of nodes and distribute the mobile code to the selected client nodes as shown in Figure 4.

ClientManager listens to the requests from MCLUSTER related to cluster membership, application execution and service those requests. In lieu of invitation packet, ClientManager running on a node will communicate its IP address to the server. On receiving the mobile code from server it load and execute the mobile code and reply the results back. The performance of cluster, during the execution of distributed applications not having significant amount of data (such as between 1-20000) deteriorate as indicated by Overhead point in Figure ??.

At these data ranges the communication overhead between MCLUSTER Server and Client nodes overwhelm the advantage of distributed processing power obtained from client nodes. When data range is extensive then the time consumed by an application so executed in a distributed manner is several times smaller than the execution time of the same application processed on a single node. The MCLUSTER model is designed in such a way that even for a large problem size (such as between 1-5000000) and average cluster size (up to N=11) performance of distributed application will not deteriorate as shown in Figure ??.

Communication Overhead Analysis : When the scalability of cluster increases, the communication links near the server are congested due to large transmission of data, thereby degrading the performance of cluster. Communication overhead includes the overhead due to exchange of data between nodes and message delay caused by network congestion. Figure ?? is represented by the result so generated after execution of the application Series for a data range 1-100000 on a cluster of 11 client nodes.

On the basis of assumption that external network load is low and constant, the communication overhead for MCLUSTER model can be parameterized as a simple linear function of number of bytes transmitted and number of client nodes selected as represented in Figure ??.



Figure 1: Figure 1 .

2.

Figure 2: Figure 2 .

3

Figure 3: Fig. 3

¹Communication

²© 2012 Global Journals Inc. (US) Global Journal of Computer Science and Technology Volume XII Issue V Version I

³© 2012 Global Journals Inc. (US) 2012 March

4 INTRODUCTION

Figure 4: Figure 4

5

Figure 5: Figure 5 .

.1 Data Range

No. of Nodes Time (Milliseconds)

[Lemeire and Dirkx ()] *Causes of blocking overhead in message-passing programs*, J Lemeire , E Dirkx . 2002. (10 th)

[Martin et al. ()] ‘Effects of communication latency,overhead and bandwidth in a cluster architecture’. R Martin , A Vahdat , D Culler , T Anderson . *Proceedings 24th Annual International. Symposium Computer Architecture (ISCA)*, D Lange (ed.) (24th Annual International. Symposium Computer Architecture (ISCA)New York, USA) 1997. General Magic, Inc. California. p. . (Mobile objects and Mobile Agents: The Future of Distributed Computing?)

[Euro PVM/MPI 2002 Conference] *Euro PVM/MPI 2002 Conference*, (Venice, Italy)

[Walker and Steel ()] ‘Implementing a full single system image unixware cluster: Middleware vs. underware’. B Walker , D Steel . *Proceedings: International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, (International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)Las Vegas, USA) 1999.

[Jon and Grimshaw (1994)] ‘Network partitioning of data parallel computations’. B W Jon , A S Grimshaw . *Proc. Third International Symposium. High Performance Distributed Computing (HPDC '94)*, (Third International Symposium. High Performance Distributed Computing (HPDC '94)San Francisco, CA, USA) 1994. April 2-5, 1994. IEEE Computer Society.

[Satoh ()] Ichiro Satoh . *Configurable Network Processing for Mobile Agents on the Internet. Cluster computing, The Journal of Networks, Software Tools and Applications*), 2004. Kluwer. 7 p. .