VOLUME I ISSUE V ICAE-2012 ISSN 2278-2540

Design and Implementation of Optimized Light-Weight Communication system in Grid

Nitin Pandya[1],Tarun Sharma[2]
Shankersinh Vaghela Bapu Institute of Technology,VASAN,GANDHINAGAR
Gujarat Tehcnological University
Ahmedabad, Gujarat, India nitin.pandya@bapugkv.ac.in[1],
tarun.sharma@bapugkv.ac.in[2]

Abstract—The advantage of high performance networks in conjunction with low-cost, powerful computational machines have made possible the development of a new set of technologies termed ”computational grids”. These technologies are making possible the creation of very large-scale distributed computing systems by interconnecting geographically distributed computational resources via very high-performance networks. This provides tremendous computational power that can be brought to bear on large-scale problems in several domains. In many problems remain before Grid computing can reach its full potential. One particularly difficult issue is that of utilizing fully the available bandwidth while being in some sense fair to competing traffic flows. It has been widely demonstrated that TCP, the communication protocol of choice for most distributed application, often performs quite poorly in the emerging high-bandwidth high-delay network environments. This has led to significant research on the development of user-level applications that can circumvent some of the performance problems inherent in TCP.

Index Terms—Light-Weight Communication,Grid Computing,Distributed Systems,Multi-casting

I. INTRODUCTION

Research efforts have focused on the development of Grid computing, a fundamentally new set of technologies that create large-scale distributed computing systems by interconnecting geographically distributed computational resources via very high-performance networks. The advanced applications being developed to execute in Grid environments include distributed collaboration across the Access Grid, remote visualization of terabyte (and larger) scientific data sets, large-scale scientific simulations, Internet telephony, and multimedia application.Challenges in designing such infrastructure arise due to the distributed nature of the resources to be used, the distributed communities, the size of the data to be shared and the limited network bandwidth.

II. EXISTING METHODOLOGIES

A grid application is defined simply as an application that is to be executed on a grid and that consists of a number of grid threads. Grid applications and grid threads are exposed to the grid application developer via the object-oriented Alchemi .NET API[4].

Madhuri Bhavsar and Dr.S.N.Pardhan are with department of Computer Science and Engineering , Institute of technology, Nirma University, Ahmedabad-382481

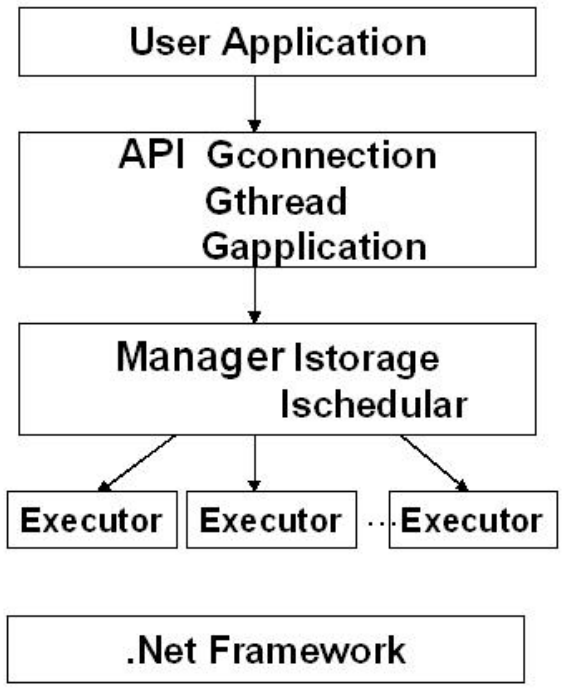

Fig. 1. Basic architecture of the grid

A. Basic architecture of communication system in the grid

The user application is send to the Grid head node(Manager), which has container in which all layer for the communication system is described as the Application model like,thread,task, etc.Then the Allocation Manager check the available resources and then the Message Handler/or Dispatcher, send or receive the message for the communication. There is security mechanism like, Authentication,Authorization is used for the communication according to the gird protocol.And finally the Communication layer can communicate to the other Grid Execution node(Executor).[9]

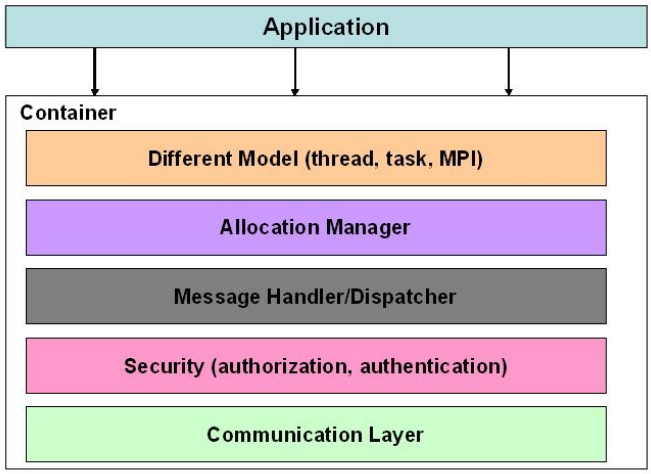

Fig. 2. Architecture of communication system

B. Normal flow of alchemi grid

The normal flow of the Alchemi grid is as described below. There is/are the user application or application client sends the application thread or the query message to the grid head node(Manager),which contains the services like message handler/dispather services, directory services, etc.Grid head node see the available resources, and send the query message or the application thread to other grid node.The grid executor or the execution node compute the application thread or the query message and give the output back to the grid head node(Manager), and that node collect the output of from all other grid execution node and combine it to gather as one and give it back to the application client or the user.

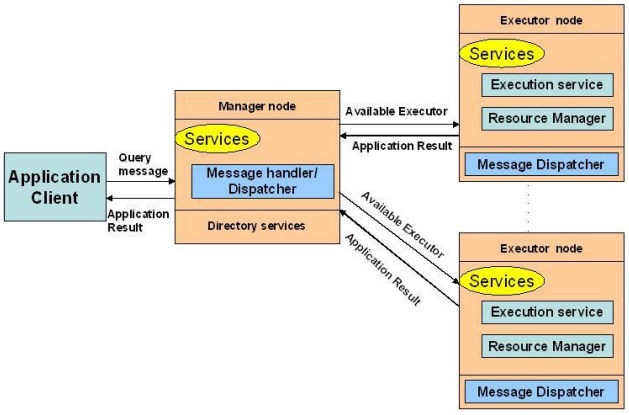

Fig. 3. Normal flow of alchemi grid

III. Multi-cast Protocol

In Grid environment there are several receiver's, for the communication the grid head node(Manager) is using the multi-cast protocol to send or receive the data or for the communication in the grid.In multi-cast protocol, the sender and receiver side socket program, in .Net (windows) is used to design the algorithm. The Proposed algorithm for the multi-cast protocol is describer for sender and receiver as below.

Sender:

- Declare and initialize all the required variable,
- Validate the input multi-cast IP address,
- Validate the port number,
- Create the instance of IP address and port number and then create the Socket,
- Set the Time to Live,
- Send the Data Packet,
- Close the socket.

Sender:

```
// Set the Time to Live
sock.SetSocketOption(SocketOptionLevel.IP, SocketOption-
Name.MulticastTimeToLive, ttl);
Console.WriteLine("Hello! Welcome " + "(return to send,
ESc to quit):");
while (!done) {
// Read and format input from the terminal
string str = Console.ReadLine();
System.Text.ASCIIEncoding encode = new Sys-
tem.Text.ASCIIEncoding();
byte[] msgToBeSent=encode.GetBytes(str);
// Send the data packet
sock.SendTo(msgToBeSent, 0, msgToBeSent.Length,
SocketFlags.None, ipep);
}
```

Receiver:

- Declare and initialize all the required variable,
- Validate the input multi-cast IP address,
- Validate the port number,
- Create the instance of IP address and port number and then create the Socket and bind the socket,
- Add membership in the multi-cast group,
- Receive the multi-cast packet,
- Format and output the received packet,
- Drop the membership and close the socket.

Receiver:

```
// Create the EndPoint class
receivePoint = new IPEndPoint(IPAddress.Any, 0);
EndPoint tempReceivePoint = (EndPoint)receivePoint;
while (!done) {
byte[] recData = new byte[MAX_LEN];
// Receive the multicast packets
int length = sock.ReceiveFrom(recData, 0, MAX_LEN,
SocketFlags.None, ref tempReceivePoint);
// Format and output the received data packet
System.Text.ASCIIEncoding encode = new Sys-
tem.Text.ASCIIEncoding();
Console.WriteLine("Received " + length + " bytes
from " + tempReceivePoint.ToString() + ": " +
encode.GetString(recData, 0, length)); }
// Drop membership
sock.SetSocketOption(SocketOptionLevel.IP, SocketOp-
tionName.DropMembership, new MulticastOption(mcIP,
IPAddress.Any));
```

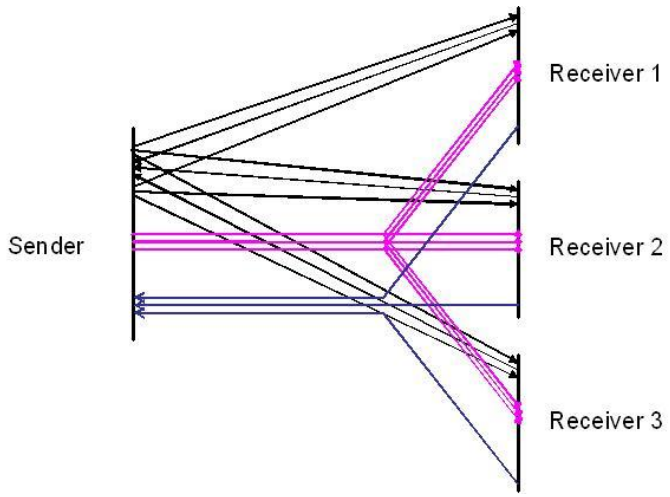

Fig. 4. Data Transmission in Multi-cast

A. API's used in Multi-cast protocol

Sender:

- New connection type.
- Connect to port on array of destination address.
- Single write sends data to all hosts.

Receiver:

- No API changes.
- IGMP is used to join or leave the group.
- Accept data on both uni-cast/multi-cast ports.
- tcp_input() accepts:
  - Packets addressed to existing uni-cast destination.
  - Those addressed to multi-cast group.

IV. RESULTS

A. Various Grid node and Execution Time

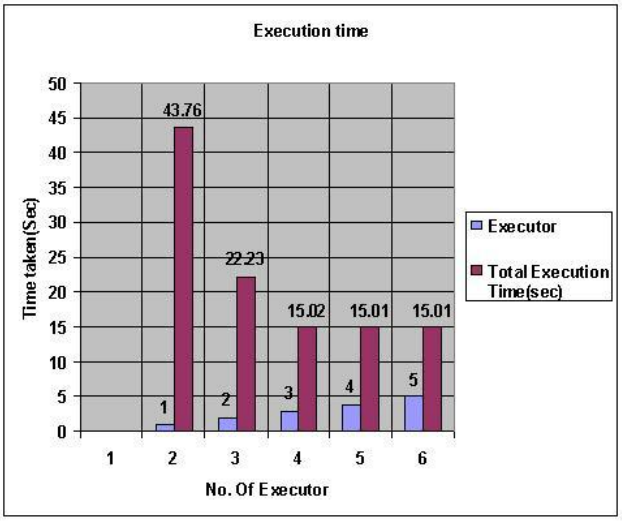

Fig. 5. Job Execution time

B. Avg bytes sent to the grid executor node

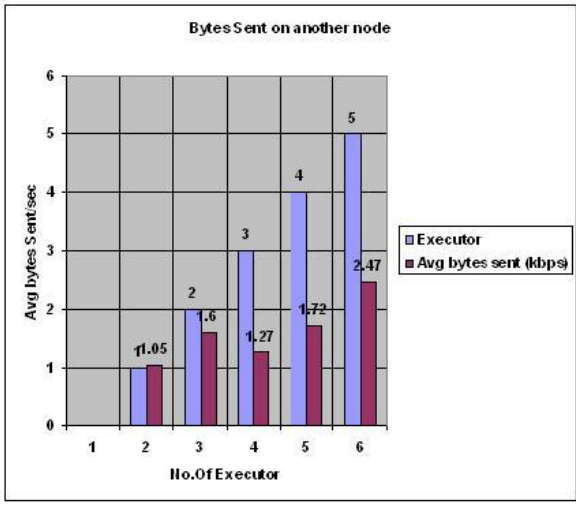

Fig. 6. Avg bytes sent to the grid executor node.

C. Avg bytes receive from the grid executor node

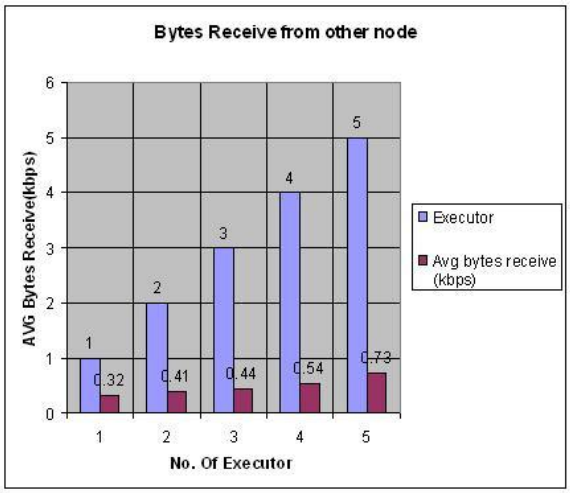

Fig. 7. Avg bytes receive from the grid executor node.

V. ANALYSIS OF RESULT

From the various kind of graphs i have shown the different parameter for measuring the performance of the grid. In first Graph, shows that the execution time of the application or the job is decrees according to the increasing the no of grid execution node.In second and third graph, the AVG bytes sent/receive on the grid is increase according to the grid execution node.

VI. Conclusion

Light-Weight Communication helps in improving the performance parameters like, execution time, network bandwidth,memory usage,data sent/receive, etc.Since existing TCP/IP protocol when used in grid, does not provide, priority based routing,rate and burst control, multi-casting, etc.An attempt is made to overcome this and build the Light-Weight Communication Protocol. From the various result, it can be concluded that using ”Light-Weight Communication System in Grid”. Application’s Execution time can be reduced up-to 10-20% of the normal Execution time. Other identified parameter is volume of data transfer and the network protocol used in Grid, specifically for the multi-casting.

References

[1] Deqing Zou, Laurence T. Yang, Weizhong Qiang, Xueguang Chen, Zongfen Han, ”An Authentication and Access Control Framework for Group Communication Systems in Grid Environment”,21st International Conference on Advanced Networking and Applications(AINA’07),0-7695-2846-5/07 2007 IEEE.

[2] I. Foster, C. Kesselman, L. Pearlman, et al.,”The Community Authorization Service: Status and Future”, In Proceedings of Computing in High Energy Physics 03 (CHEP ’03), 2003.

[3] Fu Yanfang, Liu Bailin,”Research of Communication Technology for Simulation Grid System”, 978-1-4244-2800-7/09 2009 IEEE.

[4] R. Buyya, M. Murshed, ”GridSim:A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing”, Concurrency and Computation: Practice and Experience, vol.14,pp.1175-1220, November 2002.

[5] Geoffrey Fox, ”MESSAGE PASSING: FROM PARALLEL COMPUTING TO THE GRID” 1521-9615/02/ 2002 IEEE.

[6] H. T. Ho, The MITRE Corporation, Bedford, MA, ”Windows NT Multithreaded Design in Multi-Link Translator and Display system”, 0-7803-5749-3/99 1999 IEEE.

[7] ”High Performance Cluster Computing, Architectures and Systems” Chapter 10, ”Lightweight Messaging Systems”,Rajkumar Buyya.

[8] Lazar Kirchev, Minko Blyantov, Vasil Georgiev, Kiril Boyanov, Maciej Malawski, Marian Bubak, Stavros Isaiadis, Vladimir Getov, ”USER PROFILING FOR LIGHTWEIGHT GRIDS”.

[9] George Clapp, Joel W. Gannett, Ronald Skoog,”Requirements and Design of a Dynamic Grid Networking Layer”, 0-7803-8430-W04F 2004 IEEE.

[10] Craig A. Lee, Eric Coe, B. Scott Michel, James Stepanek,Ignacio Solisy, J. Matt Clark, Brooks Davis,”Using Topology-Aware Commu- nication Services in Grid Environments” Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID.03),0–7695-1919-9/03 2003 IEEE.

[11] Jack Dongarra, Thomas Sterling, Horst Simon, and Erich Strohmaier,”HIGH-PERFORMANCE COMPUTING: CLUSTERS, CONSTELLATIONS, MPPS, AND FUTURE DIRECTIONS”, 1521-9615/05 2005 IEEE.

[12] Ryan X. Wu, Andrew A. Chien,Matti A. Hiltunen, Richard D. Schlichting, Subhabrata Sen, ”A High Performance Configurable Transport Protocol for Grid Computing”, 0-7803-9074-1/05 2005 IEEE.

[13] Marinho P. Barcellos,Maziar Nekovee, Michael Daw,”High-Performance Reliable Multicasting for Grid Applications”, Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID04)1550-5510/04 2004 IEEE.

[14] P. G. Viscarola, W. A. Mason,”Windows NT Device Driver Development”, OSR Open Systems Resources, Inc. 1999.

[15] ”A networking approach to grid computing”,Willey Edition,2004,Daniel Minoli.