Enabling Grid Computing over IPv6 within a Campus Network

Jun Chen∗, You Zou†, Zhongyu Liu‡, Qing Wu‡
∗High Performance Computing Center
Central South University, Changsha, PRC, 410083
Email: chenjun@csu.edu.cn
†College of Mathematics and Computer Science
Hunan Normal University, Changsha, PRC, 410081
Email: zouyou@csu.edu.cn
‡High Performance Computing Center
Central South University, Changsha, PRC, 410083
Email: zyliu@csu.edu.cn
Email: wuqing@csu.edu.cn

Abstract—The biggest problem to be solved in grid computing is data transmission and communication efficiency and stability. For computing grid composed of small local area network (LAN), this problem can be solved by properly building LAN with the use of IPv4 network or high-speed computing network such as Infiniband. However, when the computing resources are distributed in different physical regions, data transmission and communication will be affected by various factors such as time period and load. On the one hand, the stability of transmission cannot be guaranteed; On the other hand, the speed of cross-regional data transmission is not optimistic. As the next generation of Internet protocol, IPv6 is at a transitional stage, and the load is far less than the IPv4-based network. So, it has a better stability and higher bandwidth. With the help of IPv6 network, cross-regional computing grids can be constructed to connect the computing resources in different physical regions in order to deal with some complex computing problems. This paper studies and tests the feasibility of transplanting MPI application (which is more commonly used in distributed computing) into IPv6 network for grid computing, and presents its implementation process in detail.

Keywords—grid computing; IPv6; MPI

I. INTRODUCTION

At present, almost every high-performance computing centers are facing the same problem: no matter how rich the computing resources in computing center are, it would not satisfy all computing needs of the users. This phenomenon is particularly obvious, especially in universities which have a multi-disciplinary computing needs, software calculating for different disciplines or even the same software in different applications have different resources requirements from each other, while the structure of computing cluster is single usually, all these kinds of demand causes resources fragments easily, thereby the cluster’s overall efficiency is reduced. The popular approach of current domestic universities is to set up discipline project alone, purchase computing devices and use computing resources to calculate exclusively. In this case, many laboratories have a small cluster which can only satisfy their own needs. However, a project is not carried out entirely based on calculations, and with the development of interdisciplinary, there will come out variety of computing needs. Therefore, all laboratories are hoping to “borrow” parts of the suitable computing resources from other laboratories to promote their work, the same is true in different colleges and universities both in themselves and between. This idea of “borrowing” is coincide with grid computing, which is aimed at connecting computing resources in different physical regions for effective integration, thus completing more efficient calculation.

If using IPv4 network to accomplish these resources sharing process, there will be two difficulties: One is the limitation of address space, since the current IPv4 address is extremely rare, it is impossible to allocate an independent IPv4 address for all nodes in the grid, while VLAN and other mechanisms will have to face other difficulties brought about by cross-domain; The second is the instability of the transmission speed of IPv4, for most current network applications are based on IPv4 protocol, which makes the load of IPv4 network almost under full-load conditions in most cases, thereby, it is very adverse for the grid computing.

IPv6, as the next generation IP protocol, which is designed by the IETF, is currently in a transitional period of development, almost every university in China Education and Research Grid (ChinaGrid) has joined CNGI to be one of the CERNET2 nodes. Theoretically the address capacity of IPv6 can reach up to $2^{128}$, this is enough available for all computing nodes to be allocated an independent IPv6 address in the grid. In addition, the load of IPv6 network is much lower than IPv4 network, thus the data transmission and exchange are also more efficient.

This paper is organized as follows: at first, it analyzes the necessity of using the IPv6 network to achieve the connection of computing resources in grid, and in the second part, it introduces a test to prove the feasibility of this idea; the third part presents its implementation procedure of an
IPv6 based MPI computing experiment; The fourth part provides a brief summary of other IPv6 based applications in grid computing. Finally, it summarizes the content and proposes research work that can be furthered.

II. Speed test of IPv6 network transmission

To prove the efficiency of IPv6 network transmission speed, this paper gives a data transmission test among clusters of Central South University and other universities or campuses such as Hunan Normal University and Hunan University. Three files in different sizes are used to test the transmission speed.

Table I: IPv4 and IPv6 Data Transmission Results(KB/s)

<table>
<thead>
<tr>
<th>Location</th>
<th>50MB IPv4</th>
<th>50MB IPv6</th>
<th>150MB IPv4</th>
<th>150MB IPv6</th>
<th>200MB IPv4</th>
<th>200MB IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>484.8</td>
<td>686.6</td>
<td>763.3</td>
<td>684.4</td>
<td>482.6</td>
<td>667.6</td>
</tr>
<tr>
<td>1</td>
<td>1336.2</td>
<td>1877.4</td>
<td>568.3</td>
<td>906.3</td>
<td>700.8</td>
<td>1365.8</td>
</tr>
<tr>
<td>2</td>
<td>693.6</td>
<td>1797.4</td>
<td>666.9</td>
<td>1927.6</td>
<td>673.6</td>
<td>1950.9</td>
</tr>
<tr>
<td>3</td>
<td>1985.9</td>
<td>4205.1</td>
<td>2230.6</td>
<td>4076.3</td>
<td>2158.7</td>
<td>4117.4</td>
</tr>
<tr>
<td>4</td>
<td>2068.5</td>
<td>4082.3</td>
<td>1238.9</td>
<td>4228.5</td>
<td>2063.6</td>
<td>4214.8</td>
</tr>
<tr>
<td>5</td>
<td>145.2</td>
<td>4610.6</td>
<td>140.9</td>
<td>4690.0</td>
<td>140.7</td>
<td>4700.2</td>
</tr>
<tr>
<td>6</td>
<td>2453.0</td>
<td>4518.2</td>
<td>2427.7</td>
<td>4734.8</td>
<td>2497.5</td>
<td>4794.0</td>
</tr>
</tbody>
</table>

1Railway Campus of Central South University
2Xiangya Campus of Central South University
3Network Center of Central South University
4Mittal Building of Central South University
5Office of Campus Information Construction and Management of Hunan University
6High Performance Computing and Random Information Processing Lab of Hunan Normal University
7Network Center of Hunan Normal University

Through the above, the figure description of average transmission speed of IPv4 and IPv6 is as follows:

Figure 1: IPv4 vs. IPv6 Transmission Speed

It can be seen from figure 1: IPv6 network speeds is faster than IPv4, and the maximum difference is up to 32.8 times (HPC&RIP Lab, HUNNU). Thus, using IPv6 network to connect computing resources in different regions of grid is feasible. This conclusion is in agreement with [1].

III. MPI implementation based on IPv6

MPI (Message Passing Interface) is the most popular communication interface for computing cluster, it can be applied to a variety of computing platforms, and distributed computing of many softwares are also based on MPI. [2] introduces a kind of implementation strategy of grid computing MPI, and a completed solution for grid enabled MPI is proposed in [3], but these works are based on IPv4 network, the first work to extend IPv6 to MPI is contributed by [4], in which the author proposes a kind of implementation of expansion, and solves many problems of MPI over IPv6 mentioned in [5]. Our work does not focus on how to build a middleware or create expansion strategy for IPv6-based MPI, the present paper only introduces how to run an IPv6-based MPI computing between two clusters that distributed in different physical regions, and analyzes the diversities between them.

A. Preparation of Computing Environment

As computing resources are in different regions, it needs to meet the following three conditions to realize the IPv6 based MPI computing:

One is the operating system requirements. It requires all clusters installed in the same operating system. Almost all of the high-performance computing clusters are based on Linux, the kind of Linux operating system for high-performance computing includes SUSE, Redhat and CentOS, etc. This is relatively easy to implement. The operating system used in both regions of present experiment is SUSE 10.2:

Secondly, the regional IPv6 network must be unobstructed. This not only means that it requires each cluster can feed back the ICMP packets, but also requires each cluster support SSH or RSH protocol communications. Because MPI is generally based on SSH or RSH protocol, in this experiment, RSH protocol does not support IPv6, so we use SSH protocol to carry out the experiment. Namely, each cluster can login to the node of the other through “ssh username@ipv6_address”.

Thirdly, users’ information in each region must be consistent, which enables users in one region to use those resources to compute from other regions. Generally, in ordinary cluster system, NIS service is used to ensure the consistency of users’ information. But for the computing resources in cross regions, it is not easily to create an unified NIS service to complete the process. In this paper, we adopt a simple method by using the “scp” command to synchronize files of users’ information (/etc/passwd, /etc/group, /etc/shadow). When adding an user in one region, use the “scp” command to distribute the files to the nodes of other regions in grid.

B. Configuration of Computing Environment

We carry out the experiment with OpenMPI version1.4, which supports IPv6 Network communication [6]. What should be aware of some key points during installation and configuration, the OpenMPI should be installed in the same
The basic installation steps of OpenMPI are as follows:

1. Unzip the OpenMPI 1.4 installation kits, and configure the installation path:
   
   ```
   ./Configure --prefix=/software/openmpi-1.4
   ```

2. Run "make" command to compile the program, you can use the "-j" parameter to specify a multi-threading compilation, multi-threading compilation will be faster than the single-threading, for instance, if using 8 threads for compiling can be carried out by command "make –j 8".

3. Run "make install" command to install the compiled program to the specified directory.

After the installation of OpenMPI, we have to configure public/private key pair authentication mode for both clusters. Taking an user who named testv6 as an example, when configure this user to access cluster B from cluster A by the way of public/private key pair authentication mode, the steps for configuration process are as follows:

1. Using testv6 to login cluster A (assuming that testv6 is existed in both cluster A and B), run command "ssh – keygen – t rsa" to generate a public key file and a private key file, the default location for these two files is $HOME/.ssh/, therein the id_rsa.pub file is the public key, while the id_rsa file is the private key.

2. Login to B cluster by username testv6, and access the directory $HOME/.ssh (if the directory does not exist, then create it), create two files named authorized_keys and config separately, copy the contents of id_rsa.pub file in cluster A to the authorized_keys file, and edit the file config with an input contents as "StrictHostKeyChecking no".

After the above two steps, user testv6 can access cluster B from cluster A without inputing password. In the next, configuration in the same way to make cluster B can access cluster A without inputing password.

Generally, the executable files and input files (if present) must be the same in each computing cluster. There are two ways to achieve this goal: the first way is using the NFS service or SSHFS to mount on work directory, this approach is very easy to realize, but it needs to consider the impact of network transmission speed; The second way is to create the same directory on each cluster and copy the executable file and input files to them, the program read data quickly by this approach, but it needs to consider synchronization of data files. In specific use, we should choose different schemes according to the reality of the situation.

C. Test Cases

Computing resources used in this experiment are from High Performance Computing Center of Central South University and School of Information Science and Engineering of Central South University, which belong to different campus (As shown in Figure 2.). These two clusters meet the three requirements of IPv6 based MPI computing and each has installed OpenMPI 1.4.

The program is used for all-to-all data communication, which is a common way to test the complete exchange operations of data. Four different sizes of double precision matrix are used to test the all-to-all data exchange in both IPv4 and IPv6 network between two nodes of each cluster (See Figure 3 for an illustration.).

From Table II, we find that the result of IPv6 test is better, but not obvious, because both IPv4 and IPv6 networks are made up with the same physical connection, but IPv6 still has it advantage, for it can avoid NAT, and it has smaller data header in transmission than IPv4, the security of data
transmission can be accomplished by using the IP Security Protocol (IPsec) [7].

IV. OTHER APPLICATIONS

IPv6-based MPI model treats resources in different regions as a unit, which is very useful for the parallel computing which with little data exchange, but for those applications which are not based on MPI, this approach seems to be useless. However, we can still make use of the high transfer speed of IPv6 to complete some other kinds of cross regional calculations.

Take the LS-Dyna calculation as an example, generally, the input file of this application is less than 10MB, while the output file is usually up to tens or hundreds GB, when one of the regions in grid is lacking of computing resources, we can use IPv6 network to transfer the input file to another region, and then send the result back to local region and do some further processing. Actually, this process is using IPv6 to achieve an efficient data transmission and resources sharing, the user only needs to get an account in other regions to complete the process.

V. CONCLUSIONS AND FURTHER WORKS

We are experiencing the transition period of IPv6 network these years, we can use its high-speed bandwidth to accomplish some specific applications, this paper just takes advantage of its characters to realize the applications in grid computing, it introduces the advantages of IPv6 by a series of tests on transmission speed, then proposes an IPv6-based MPI model, and gives an example for other applications that can be implemented through IPv6 network.

Generally, it needs a job scheduling software for resources regulation and management in a big cluster system, but the model in this paper does not consider a job scheduling, in the further work, we can establish a cross regional scheduling mechanism based on IPv6 to accomplish the job scheduling and job management, which automatically chooses the operating location and mode through the scheduling system and users’ request, in this way, it is hoped that the resources of grid computing will be used more reasonably and more efficiently.

ACKNOWLEDGMENT

This work was supported in part by the High Performance Computing Center of Central South University, Network Center of Central South University, High Performance and Random Information Processing Laboratory of Hunan Normal University, Office of Campus Information Construction and Management of Hunan University, Network Center of Hunan Normal University, School of Information Science and Engineering of Central South University.

REFERENCES