Modeling HPC using the commodity PCs

Swee Liang Aw, Arslan Ayari and Sarah Aw

Foundation Program, Math and Computer Dept, Qatar University, Doha, Qatar

Abstract. Qatar University is one of the Universities in Middle East that's seriously takes part in the enormous task of creating a productive research environment through the utilization of university research, knowledge, and resources for the benefit of present and future generations. Undergraduate students have been encouraged to involve in research project which will be funded by Qatar University Internal Student Grant. This paper will discuss about the project that's has been done by the undergraduate students from Qatar University in Prototyping the Supercomputer using the Commodity PCs. This paper will discuss about the whole approaches and process the implementation of the project.

Keywords: High Performance Computing, Clustering, MPI.

1. Introduction

Foundation Program's Math and Computer department is a department that's responsibility to deliver the math and computer courses for the pre undergraduate students before they can starts on their undergraduate program. For the computer courses, the syllabus were designed by the experienced teaching professional to ensure that's course contents are suitable and useful for the pre undergraduate students. One of the chapters in Computer 1 course syllabus that's offered is about the types of Computer. In this chapter, students are exposed about the types of Computer that's being using by the individual and organization in nowadays world. In this topic, students were introduced about the Supercomputer, Mainframe, Minicomputer and Microcomputer. All the instructor explained about the types of computers and most type of computer are exposed to the students except for the Supercomputer. In this project, two third years' undergraduate students from the College of Engineering, majoring in Computer Science were selected to join in to do the research in this project. These students were selected according to their criteria and show their interest in doing the research project. This research project explained about the concept of the Supercomputer in performing the compilation and calculation of the complicated tasks which needs the very high processing power. The major objective of this project is the build a cluster that's have very high processing power to ensure that students will understand how the Supercomputer works. In this project, we have implemented the method in building the High Performance Computer Clusters where using the Linux and MPI tools. High Performance Computer Cluster that's build has proved its ability to have much better performance than a single computer by measuring the performance using the benchmark tools.

2. Cluster and MPI

2.1. Cluster

Cluster is group of connected computers, operated together in each other which formed a single computer [2]. The components of a cluster are commonly connected to each node through local area networks (nowadays). Clusters are deployed to improve performance and availability over single computers of comparable speed or availability with reduced cost expenses. Computer Clusters can be divided into 3 major categories, which are *High-availability (HA) Clusters*, *Load-balancing Clusters* and *High Performance Clusters.*

High-availability Cluster or known as Failover Cluster are implemented to improving the services availability by including several redundant nodes to make sure that the provided services are always available and stable. HA Cluster implementation will make sure that's redundant nodes is always standing by as the backup for the on access nodes to prevent from the out of service. The implementation normally involved 2 nodes/computers where the redundancy will prevent from single point failure happened to the provided services.

Load-balancing cluster is needed to balance the computational work among different computers. By using the load balancing, the workload among the computer will be shared as a single virtual machine from several connected machines.

High Performance Cluster is used primarily for computational purposes, such as simulations and complicated scientific experiments computation and are of special interest to the scientific community. They commonly run simulations and other CPU-intensive programs that would take an inordinate amount of time to run on regular machine. This cluster design is usually referred to as Beowulf Cluster and traditionally been called as “Supercomputing”. The popular middleware such as MPI (Message Passing Interface) or PVM (Parallel Virtual Machine) permits compute clustering programs to be portable to a wide variety of clusters.

2.2. MPI

MPI (Message Passing Interface) is a kind of programming model database news can transmit, and becomes the representative of programming model [1]. This is a kind of standard representative but doesn’t specify certain realization. MPI that has not only various advantages, such as good transplantation ability, powerful function, high efficiency and so on but also different free, efficient, practical realization edition, applies to the news-transmitting model based on the parallel computer system which has distribution memory, moreover, nearly all of the manufacturers of parallel computer offer the support of technology, which other parallel environment does not compare. MPI which produced in 1994 and became MPI-2 today defines the following standards

- MPI database can be used by Fortran and C, which on grammars complies with all application rules of function or process database, has no difference with general function or process and ensure transplant ability that MPI program complied special standard scan run on any platform.
- The MPI supported by high-speed networks (10 Gigabit Ethernet, InfiniBand, Myrinet, Quadrics) and proprietary high-end computing systems (Blue Gene, Cray, SiCortex)
- Support expands I/O, dynamic process, unilateral communication, unblocked crowd communication pattern.

In this project, we have implemented the MPICH2 [3] which using the MPI implementation that efficiently supports different computation and communication platforms including commodity clusters (desktop systems, shared-memory systems, multi core architectures), and enable cutting-edge research in MPI through an easy-to-extend modular framework for other derived implementations.

3. Implementation and Configuration

In this project, the machines that’s selected has the same hardware specification, using the same version of the Operating System and all the machines has been configured using the same tools components such as NFS, GCC and SSH. For the hardware parts, we used the Lenovo M51 desktop for all the nodes which are interconnected through an Ethernet Network, using the *Beowolf* cluster as the Cluster Architecture and Ubuntu version 10.04 or Lucid Lynx operating systems completely installs in all the nodes as the Operating System. We have named the nodes as node1 (Master Node), node2, node3. Below sections will explain about the process of setting up the clusters.

3.1. Defining hostnames in all the nodes

Defining hostnames in all the nodes is the most essentials part to setup the cluster. All nodes must be defined a specific IP address and hostname so that’s the communication between the nodes are working perfectly and the nodes are not going to get any disturbed by the DHCP services. The hostnames has been defined in the etc/hosts for all the nodes.

```
 127.0.0.1     localhost
192.168.1.1    node1
192.168.1.2    node2
192.168.1.3    node3
```

After the hostname has been defined in all the nodes, ping command has been used to make sure the hostname are correctly defined and all the nodes are interconnected. Example:

```
mpiuser@node1:~$ ping node1
mpiuser@node1:~$ ping 192.168.1.1
```

To make sure that's the nodes are interconnected; ping command has been used from node1 to check the interconnection between the nodes.

```
mpiuser@node1:~$ ping node2
mpiuser@node1:~$ ping node3
```

The successful message will shows on the terminal screen showing the results of the packets received, percentage of the packet loss and response time.

3.2. Installing and configuring Tools Component

Installing and configuring the Network File System protocol is very important in building the cluster in Linux environment because NFS enable the nodes to access the shared folder and to make sure that's all the nodes are maintained in using the same setting and configuration. The installing and the configuration of NFS have been done as follow:

```
mpiuser@node1:~$ sudo apt-get install nfs-kernel-server
```

After the NFS has been successfully installed, one shared master folder will be created to store all the data and programs.

```
mpiuser@node1:~$ sudo mkdir /mirror
```

To ensure the contents in Shared Master Folder can be shared with other nodes, we have edited the /etc/exports file in node1to add this line /mirror *(rw,sync) by using this command:

```
mpiuser@node1:~$ sudo echo /mirror *(rw,sync ) >> /etc/exports
```

Below commands has been executed to make sure that other nodes are enabled to access the Shared Master Folder

```
mpiuser@node2:~$ sudo mount node1:/mirror /mirror
mpiuser@node3:~$ sudo mount node1:/mirror /mirror
```

and the file /etc/fstab has been edited to add this line node1:/mirror/mirror to make sure all the nodes mount the Master Shared Folder each time the machine boot up.

The next component to be installed and configured is SSH Server. Before the installation of SSH Server begins, we have created a new user specifically to run the programs and assigned this user to have full privilege on the Master Shared Folder.

```
mpiuser@node1:~$ sudo chown mpiuser /mirror
```

Installation and configuration of SSH Server has been done by not setting pass phase for the communication between nodes.

```
mpiuser@node1:~$ sudo apt-get install openssh-server
```

Generate DSA key for mpiuser, leave pass phase empty and authorized key has been added.

```
mpiuser@node1:~$ ssh-keygen –t dsa
mpiuser@node1:~$ cd .ssh
mpiuser@node1:~/ .ssh$  cat id_pub.dsa  >> authorized_key2
```

Last step in Installing and configuring Tools Component is to install GCC and other compilers, such Inter Fortran, SGI compiler

```
mpiuser@node1:~$ sudo apt-get install build-essential
```

4. Installing and configuring MPICH2 and MPD

After the tools and components has been successfully installed and configured, the next step is to install the MPI program. For this project, we have chosen the MPICH2. The process of installation of MPICH2 will be shown as follows:

```
mpiuser@node1:~$ sudo apt-get install mpich2
```

After successfully compiling and installing mpich2, we edited the /mirror/mpiuser/.bashrc by adding these lines.

```
export PATH=/mirror/mpich2/bin:$PATH
export PATH
```

```
LD_LIBRARY_PATH=”/mirror/mpich2/lib:$LD_LIBRARY_PATH”
export LD_LIBRARY_PATH
```

Next, we run the command in order to define MPICH installation path to SSH and for testing purposes, we have executed which command.

```
mpiuser@node1:~$ sudo echo /mirror/mpich2/bin >> /etc/environment
```

5. Testing the MPICH2 installation

```
mpiuser@node1:~$ which mpd
mpiuser@node1:~$ which mpiexec
mpiuser@node1:~$ which mpirun
```

Setup the MPD is the final steps in this project. We have created the mpd.hosts in mpiuser’s home directory with the nodes name, after that we executed the command of

```
mpiuser@node1:~$ echo secretword=something >> ~/.mpd.conf
mpiuser@node1:~$ chmod 600 .mpd.conf
```

The final work is to run the MPD daemon by executed the mpdboot and mpdtrace commands

```
mpiuser@node1:~$ mpdboot –n 3
mpiuser@node1:~$ mpdtrace
```

Finally we got the output from the mpdboot and mpdtrace command and the result shows the list of all the nodes involved. If the result shoes error message, we had to run the mpdcheck command on all the nodes to find the possible errors in conf files.

After we obtained output from the last command shows that we the MPI programs has been successfully installed and configured and it is ready to work as the HPC cluster that’s will past the processing to all the nodes. To test the cluster, we have run the command to compile the files that’s located in MPICH2 example folder. The command that’s being executed as follows:

```
mpiuser@node1:~$ mpiexec –n 3 cpi
```

The result from the executed command will indicates that’s the cluster is ready for the compilation.

6. Testing Tools

To confirm that’s the steps and method to build the high performance computer cluster is completed. A benchmark tool has been used to measure the system floating point of the computer clusters.

7. Conclusion

This project has exposed the undergraduate students to basic concepts of the “Supercomputing” by giving them opportunity to build the High Performance Computing cluster using Desktop and Linux Operating System. The basic concepts of the supercomputing or high performance computing is very essential for the undergraduates students that’s studying in Computer Science so that’s they can experiences with the realization of the method in compiling the high complication scientific experiments or calculation.

8. Acknowledgements

The fully appreciation are given to Qatar University, where have approved the grant to supported on this project and Thanks for Dr Maha Nabhan, as the head of department in Foundation Program, Math and Computer department for her support on this project.

9. References

[1] M. Snir, S. Otto, S. Huss-Lederman, D. Walker and J. Dongarra, “*MPI: The Complete Reference*” Cambridge, Massachusetts, London, 1996.

[2] G. Hager and G. Wellein, *Introduction of HPC for Scientists and Engineer*, CRC Press: Taylor & Francis Group, 2011.

[3] Argonne National Laboratory, Mathematics and Computer Science.” The Message Passing Interface (MPI) standard”. http://www.mcs.anl.gov/research/projects/mpich2/