IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 2, Issue 3 (July-Aug. 2012), PP 06-11
www.iosrjournals.org

Analysis and Implementation of Cluster Computing Using Linux Operating System

Zinnia Sultana
(Lecturer, Department of Computer Science & Engineering, International Islamic University Chittagong, Bangladesh)

Abstract : *Cluster computing is one of the most interesting innovations in the field of parallel computing in the recent past. Due to record low prices on hardware and the availability of free, open source software, massive amounts of computing power are available to the general population. This availability allows for the implementation of cluster computing that can serve as the function of one PC based computer. This paper focuses on the implementation and the use of small cluster computing of three computers. It is quite impossible for us to effort a costly supercomputer and knows its performance. To minimize cost we have to consider some issues, these are software choice, installation, configuration and networking. In addition to the implementation of the cluster computing here uses LINUX operating System and OSCAR (Open Source Cluster Application Resources) & MPICH (Implementation of MPI, the Standard for the message passing libraries) that is fully free.*
Keywords - *Cluster Computing, MPI, OSCAR, Parallel System, Super Computer*

I. INTRODUCTION

The idea of cluster computing may be odd to someone new to the computing world, but clustering computers is not a new idea. Clustering theory dates back to as early as 1970. IBM was the first company to implement the clustering theory into its design.

In the mid-1980s, Digital Equipment Corporation released the VAX-cluster, a clustering of minicomputer was supposed to solve the fault tolerance dilemma of systems. The VAX-cluster was an attempt to duplicate every component that could fail within a system and run those components simultaneously. This clustering produced a computing environment in which two computers could provide simultaneous service to users.

In 1995, Microsoft was delighted to announce the development of "wolfpack" the code name for the Mocrosoft Cluster Server (MSCS) software package developed for the windows NT 4.0 Enterprise server platform. This software was developed, in collaboration between Digital Equipment Corporation and Tandem Computers, to allow two Windows NT 4.0 Enterprise servers to share a hard disk, providing automatic fail-over in the event of a failure within one of the servers. This would have been a triumph for Microsoft, for the cluster server functionality was beginning to be in high demand.

In 2000, Microsoft released Windows 2000, along with a version of their server product called windows 2000 advanced server. This server offered the reliability and stability of an enterprise class operating system, as well as the tools and applications needed to build high availability cluster servers.

Cluster computing is a system, which consists of a server node and multiple client nodes connected via Ethernet or some other network topology , where each node is an individual PC or Workstation and where the nodes work together to solve a computational problem concurrently or in parallel.

With a few exceptions, network hardware is not designed for the parallel processing. Typically latency is very high and bandwidth relatively low compared to SMP and attached processors.

We choose Linux as an operating system because Linux is well known for being an easy-to-use commodity, reliability and fully free for all. No other Operating System allows for this level of versatility.

II. CLUSTER COMPUTING

A cluster is a group of independent computer system, interconnected through a network, that work together to solve a common problem. This definition is too generic, covering a broad range of system, from two computers connected through a LAN , to form massively basic PC based supercomputer technology.

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to objects in the other cluster. Cluster computing is a type of parallel processing system, which consists of a connection of interconnected stand- alone/complete computers cooperatively together as a single, integrated computing resource.

1.1 General Architecture of cluster computing

Three basic units to build cluster computing:

The first unit of the cluster computing is a computer. Clustering use complete PC instead of expensive especially designed computers made for parallel computing. Due to this, clusters can typically be built for far less money, as no special architecture needs to be used. The second item required for a cluster computing is the network. The network is used for all intra-clustering communication such as message passing. Cluster computing need the fastest network possible since the sending of data messages between computers in the cluster is the main reason for slowdown in a parallel processing. The final required item for a cluster is a message passing library. Message-passing libraries allow for the creation of parallel programs on a cluster by providing functions used to send messages and data between cluster components. The main message-passing library is MPI. MPI is an open standard and has a few different implementations by various groups and is the library most used today.

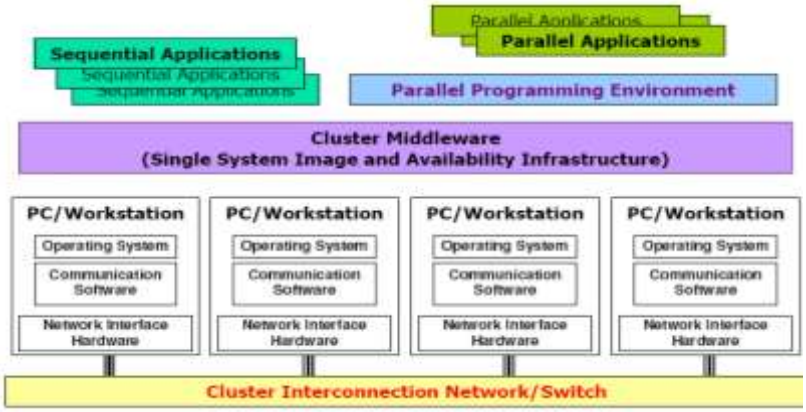

Figure 1: General Architecture of Cluster[1]

1.2 Why Linux for Cluster Computing

Cluster computing is currently both the most popular and the most varied approach, ranging from a conventional network of workstations (NOW) to essentially custom parallel machines that just happen to use Linux PCs as processor nodes. Linux run on just about any hardware platform imaginable. Linux is well known for being an easy-to-use commodity and fully free for all. Although the availability for Linux drivers might not be as prevalent as other operating systems, there is still plenty of hardware that works without a hitch ,Linux also supports a great deal of legacy hardware ,enabling older computers to be brought back into service .The creators of Linux even envision it as the premiere OS of embedded device because the kernel can be modified in any shape or form (Although Linux Torvalds invented Linux and holds the copyright , he didn't write the entire thing himself)No other OS allows for this level of versatility. It's this approach to modular computing that makes Linux for cluster Computing.

1.3 Topology used in Cluster Computing

Setting up interconnection networking for small cluster computing requires some effort to successfully put together the system. Normally to build a cluster computing system requires at least one network interface card (NIC) per PC and one network switch or hub to exchange information among the PCs. The underlying data exchanging devices such as Ethernet will have to avoid any contention during data transferring operation that happens quite often on cluster system. Using this kind of device helps promote the scalability of the system.

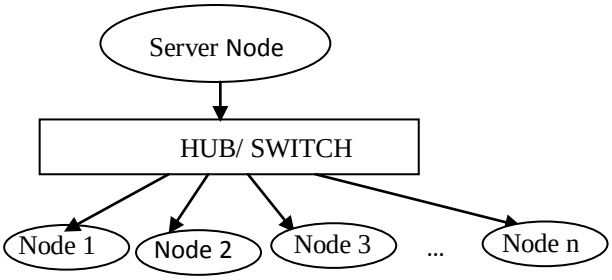

Figure 2: Using network topology to connect multiple PCs to form a cluster computing environment

III. CLUSTER COMPUTING IMPLEMENTATION USING OSCAR

The OSCAR (Open Source cluster Application Resource) software package intended to simplify the complex tasks required to install a cluster computing. While the usual intended use of OSCAR clustering is for high performance computing (HPC), OSCAR clustering can be used for any cluster-enabled kinds of

applications. Since OSCAR is aimed towards HPC, several HPC related packages are installed by default, such as popular MPI implementations, PVM, PBS etc.

1.1 Why we chose OSCAR?

OSCAR is a fully integrated easy to install bundle of software designed to make it easy to build and use a cluster computing. Everything we need to build , maintain and a modest size LINUX cluster is included in OSCAR that is fully free for all.

1.2 OSCAR software components

C3 is a command line interface that may also be called within programs. The cluster command and control (C3) tool suite was developed for use in operating the LINUX cluster.

MPI:

Message passing is a model for interactions between processors within a parallel system. In general, a message is constructed by software on one processor and is sent through an interconnection network to another processor , which then must accept and act upon the message contents. Although the overhead in handling each message (latency) may be high, there are typically few restrictions on how much information each message contain.

LAM:

LAM(Local Area Multicomputer) is an MPI programming environment and development system for heterogeneous computers on a network. With LAM , a dedicated cluster or an existing network computing infrastructure can act as one parallel computing solving one problem.

LUI:

The LINUX utility for cluster installation(LUI) is an open source utility for installing LINUX workstations remotely, over an Ethernet network. LUI provides tools to manage installation resources on the server, that can be allocated and applied to installing clients, allowing users to select just which resources are right of each client.

System installation suite:

The system installation suite(SIS)is a cluster installation tool developed by the collaboration of the IBM Linux technology center. SIS was chosen to be the installation mechanism for OSCAR for multiple reasons:

SIS is a high quality , third party open source product that works well in the production environments.
SIS does not require the client nodes to already have Linux installed.
SIS maintains a database containing installation and configuration information about each node in the cluster computing.
SIS supports heterogeneous hardware and software installation.

1.3 Minimum System Requirements:

The following is list of minimum system requirements for the OSCAR server node:
CPU of i586 or above
A network interface card that supports a TCP/IP stack
At least 4GB total free space
An installed version of Linux, preferably a fully supported distribution
The following is list of minimum system requirements for the OSCAR server node:
CPU of i586 or above
A disk on each client node
A network interface card that supports a TCP/IP stack
Same Linux distribution and version as the server node
All clients must have the same architecture
Monitors and keyboards may be helpful but not required

1.4 Cluster computing implementation using MPI:

Message passing is a model for interactions between processors within a parallel system. In general, a message is constructed by software on one processor and is sent through an interconnection network to another processor, which then must accept and act upon the message contents.

Although the overhead in handling each message may be high, there are typically few restrictions on how much information message may contain. Thus, message passing can make it a very effective way to transmit a large block of data from one processor to another.

Setup Process:

For implementation of MPI we used MPICH software, which is free for all. Operating system is obviously LINUX because it is also free for all. We used RedHat LINUX 7.3 version, which is fully supported.

Following steps should be followed for implementation:
-Configure rhs,rcp,rlogin and telnet to all nodes(including master), for configure we need to edit etc/xinet.d and make enable corresponding services.
-For remote operation edit the $HOME/.rhosts file such that each node can access without password. Must use SSH for remote operation.
-Configure NFS to make common file system in all nodes. To make such things we need to edit etc/exports in master mode and etc/fstab in client nodes.
-Install MPI in master node
-Install MPICH in master node.
-unzip and untar.tar.gz
-run. /configure
-run#make
-run#make PREFIX=$HOME/mpinstall
-Configure lamboot in master mode
-create $HOME/host_file that contains all node such as
Node0
Node1
……..
Noden
-run the lamboot command
$lamboot -v host_file
This wants password of the user
-Now we are in the world of Cluster Computing

IV. ANALYSIS AND TESTING

After implementing cluster computing on three computers, using OSCAR & MPICH we needs to check out its performance. We know that cluster computing is needed to solve computational problem in parallel. In this paper we want to find out can we benefited by cluster computing or not.
For this reason we compile & run bubble sort program in MPI Environment

```
#include<stdlib.h>
#include<stdio.h>
#include<time.h>
#define NUM_ITEMS 100000
void bubblesort(int numbers[],int array_size);
int numbers[NUM_ITEMS];
int main(int argc, char*argv[])
{
//declare initializing variable
int nprocess, rank;
char processor_name[MPI_MAX_PROCESSOR_NAME];
int length;
int i,j,d=NUM_ITEMS;
double Start,End,Time;
    //freopen("a.out","a",stdout);
    //initialize MPI
MPI_Init(&argc,&argv);
MPI_COMM_SIZE(MPI_COMM_WORLD,&nprocess);
MPI_Get_processor_name(processor_name,&length);

if(rank=0)
        {
        if(nprocess=1)
                {
                Start=MPI_Wtime();
                for(i=0;i<d;i++)
                numbers[i]=rand()%32500
                bubbleSort(numbers,d);
                End=MPI_Wtime();
                Time=End-Start;
                printf("%.6f mili seconds need for sort %d data with Bubble Sort.\n",Time,d);
```

```
	}
	else
	{
	Start=MPI_Wtime();
	for(j=0;j<nprocess;j++)
		{
			//fill array with random integers
			for(i=0;i<d;i++)
			numbers[i]=rand()%32500
			bubbleSort(numbers,d);
		}
		End=MPI_Wtime();
		Time=End-Start;
				printf("%.6f mili seconds need for sort %d data with Bubble
		Sort.\n",Time,d);
		}
	}
	else
		{
			if(nprocess=1)
	{
		Start=MPI_Wtime();
		for(i=0;i<d;i++)
		numbers[i]=rand()%32500
		bubbleSort(numbers,d);
		End=MPI_Wtime();
		TIME=END-START;
		printf("%.6f mili seconds need for  %d data with Bubble Sort.\n",Time,d);
	}
	else
		Start=MPI_Wtime();
		for(j=0;j<nprocess;j++)
		{
		//fill array with random integers
		for(i=0;i<d;i++)
		numbers[i]=rand()%32500
		bubbleSort(numbers,d);
		}
		End=MPI_Wtime();
		Time=End-Start;
				printf("%.6f mili seconds need for sort %d data with Bubble
		Sort.\n",Time,d);
		}
	}
/*Terminate MPI*/
MPI_Finalize();
return 0;
}
void bubbleSort(int numbers[], int array_size)
	{
		int i,j,temp;
		for(i=(array_size-1);i>0;i--)
		{
		for(j=1;j<i;j++)
		{
		if(numbers[j-1]>numbers[j])
		{
							temp=numbers[j-1]
		numbers[j-1]=numbers[j]
		numbers[j]=temp;
```

```
                }
            }
        }
    }
```

We analyze and testing cluster computing on three PC. When we run program in single PC it takes 141 seconds for sorting 1000000 of data. It takes 84 seconds when the program runs in cluster computing environment on two PC. 47 seconds is needed for run the program in three PC in cluster computing environment. The following graph shows the result

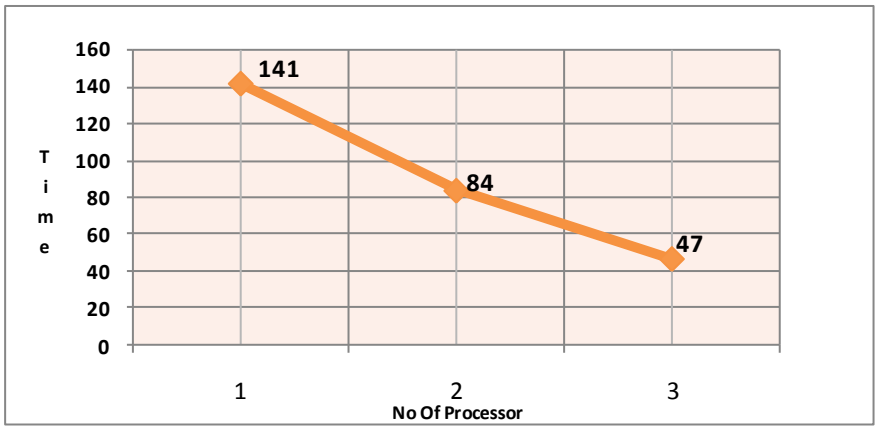

Figure 3: Performance testing graph.

So we find that processing time is reducing tremendously after cluster computing.

V. CONCLUSION

Cluster computing has opened the world of parallel computing to the average user. Parallel programming is depictive different from normal programming for single processor system. This is due in part to the fact that communication between processors must be kept to a minimum and network speed must be taken into account during programming. Also debugging in parallel programming is much harder due to the fact that the program may be working right for all but one processor. I felt that the visualization tools and libraries were extremely useful and helped me a lot to understand the programs much better.

REFERENCES

[1] Buyya, *High performance Cluster Computing* (NJ: Prentice-Hall,1999).

[2] Message Passing Interface Forum. MPI: A Message Passing Interface. In *Proc. Of Supercomputing '93*, pages 878–883. IEEE Computer Society Press, November 1993.

[3] Benoît des Ligneris, Stephen L. Scott, Thomas Naughton, and Neil Gorsuch. Open Source Cluster Application Resources (OSCAR) : design, implementation and interest for the [computer] scientific community. In *Proceeding of 17th Annual International Symposium on High Performance Computing Systems and Applications (HPCS 2003)*, pages 241–246, Sherbrooke, Canada, May 11-14, 2003.

[4] Benoît des Ligneris, Stephen L. Scott, Thomas Naughton, and Neil Gorsuch. Open Source Cluster Application Resources (OSCAR) : design, implementation and interest for the [computer] scientific community. In *Proceeding of 17th Annual International Symposium on High Performance Computing Systems and Applications (HPCS 2003)*, pages 241–246, Sherbrooke, Canada, May 11-14, 2003.

[5] System Installation Suite (SIS),http://www.sisuite.org/.

[6] NSF/TFCC Workshop on Teaching Cluster Computing Wednesday, July 11th - Friday July 13th, 2001 Department of Computer Science University of North Carolina at Charlotte

[7] Richard Ferri. The OSCAR revolution. *Linux Journal*, (98), June 2002. http://www.linuxjournal.com/article.ph p?sid=5559.