Design And Implementation Of High Performance Computing Cluster For Educational Purpose

Anjali Rathaur[1], Dr.Rajendra Kumar[2]
[1]Dept of CSE/IT,
[2]Professor, Dept of CSE/IT
[1, 2] NITM Datia, India

Abstract- *This project work confronts the issue of bringing high performance computing (HPC) education to those who do not have access to a dedicated clustering environment in an easy, fully-functional, inexpensive manner through the use of normal old PCs, fast Ethernet and free and open source softwares like Linux, MPICH, Torque, Maui etc. The projects primary goal is to provide an instantaneous, distributed computing environment. A consequence of providing such an environment is the ability to promote the education of high performance computing issues at the undergraduate level through the ability to turnan ordinary the shelf networked computers into a non-invasive, fully-functional cluster.*

Keywords- High Performance Computing, Clustering, C-DAC, GARUDA.

I. INTRODUCTION

HPC is a collection or cluster of connected, independent computers that work in unison to solve a problem. In general, the machines are tightly coupled at one site, connected by Infiniband or some other high-speed interconnect technology. The High Performance Computing (HPC) allows scientists and engineers to deal with very complex problems using fast computer hardware and specialized soft- ware. Since often these problems require hundreds or even thousands of processor hours to complete, an approach, based on the use of supercomputers, has been traditionally adopted.

Recent tremendous increase in a speed of PC-type computers opens relatively cheap and scalable solution for HPC using cluster technologies.

Types of HPC architectures-

1)Symmetric multiprocessors (SMP),
2) Vector processors,
3) Clusters.

II. LITERATURE SURVEY

PVM- PVM (Parallel Virtual Machine) is a freely-available, portable, message-passing library generally implemented on top of sockets. PVMs daemon based implemen- tation makes it easy to start large jobs on multiple machines. PVM was the first standard for parallel computing to become widely accepted.

JavaSpaces- JavaSpaces is Java's parallel programming framework which operates by writing entries into a shared space. Programs can access the space, and either add an entry, read an entry without removing it, or take an entry.

Speedup and Efficiency-

Speedup-

The speedup of a parallel code is how much faster it runs in parallel. If the time it takes to run a code on 1 processors is T1 and the time it takes to run the same code on N processors is TN, then the speedup is given by

$$S = T1/TN$$

This can depend on many things, but primarily depends on the ratio of the amount of time the code spends communicating to the amount of time it spends computing.

Efficiency-Efficiency is a measure of how much of available processing power is being used. The simplest way to think of it is as the speedup per processor. This is equivalent to defining efficiency as the time to run N models on N processors to the time to

III. DESIGN AND IMPLEMENTATION

Beowulf Clusters: A Low cost alternative

Beowulf is not a particular product. It is a concept for clustering varying numbers of small, relatively inexpensive computers running the Linux operating system. The goal of Beowulf clustering is to create a parallel-processing supercomputer environment at a price well below that of conventional supercomputers.

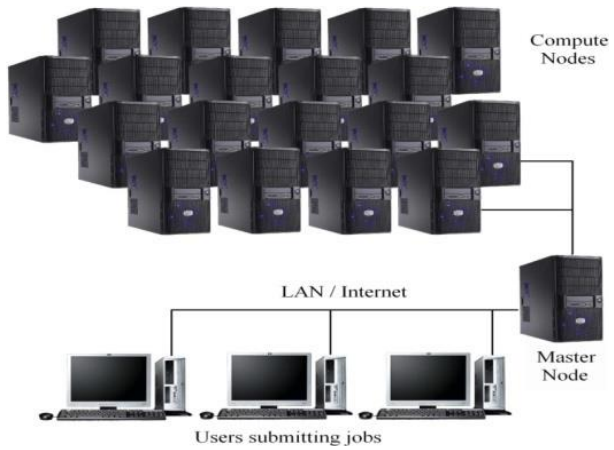

Figure 3.1: The Schematic structure of proposed cluster

A Beowulf Cluster is a PC cluster that normally runs under Linux OS. Each PC (node) is dedicated to the work of the cluster and connected through a net- work with other nodes. Figure 3.1 schematically shows the structure of a proposed cluster. In this cluster, a master node controls other worker nodes by communicat- ing through the network using the Message Passing Interface (MPI).

IV. EXPERIMEMTS

To evaluate the usage and acceptability of the cluster and its performance few parallel programs are implemented. The first one is a finding the prime numbers in given range. The second is to calculate the value of π. Then one embarrassingly parallel program to solve circuit satisfiability problem is tested. Implemented 1D Time Dependent Heat Equation and Radix-2 FFT algorithms as a real life programs. The work of a global problem can be divided into a number of independent tasks, which rarely need to synchronize. Monte Carlo simulations or numerical integration are examples of this. So here in below examples the code that can be parallelized is found and then it is executed simultaneously on different cluster node with different data. If the parallelizable code is not depend on the other output of other nodes we get a better performance. The essence is to divide the entire computation evenly among collaborative processors. Divide and conquer.

Finding Prime Numbers- This C program counts the number of primes between 1 and N, using MPI to carry out the calculation in parallel. The algorithm is completely naive. For each integer I, it simply checks whether any smaller J evenly divides it.

The total amount of work for a given N is thus roughly proportional to $1/2 * N^2$. Figure 4.1 shows the performance of cluster for finding various primes as compared to single machine.

This program is mainly a starting point for investigations into parallelization.

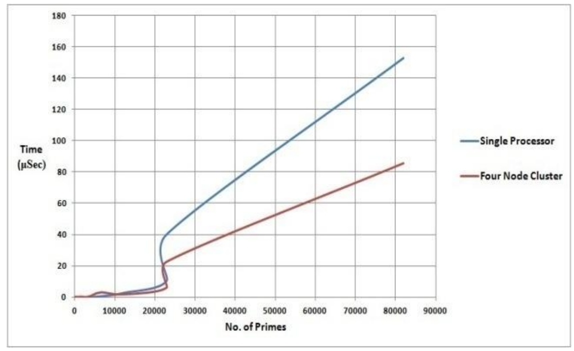

Figure 4.1: Graph showing performance for Finding Primes

Here the total range of numbers for which we want to find the primes are divided into equal parts and then distributed amongst the computing nodes. Every node has to carry out its task and send back the results to master node. At last its the job of master node to combine the results of all the nodes and give the final result.

PI Calculation -

The number π is a mathematical constant that is the ratio of a circle's circumfer- ence to its diameter. The constant, sometimes written pi, is approximately equal to 3.14159. It calculate the value of π using:

$$\int_0^1 \frac{4}{1+X^2}\, dx = \pi$$

Then compare the calculated π value with the original one and find out the ac- curacy of the output. Also the time taken by program to calculate it is also displayed. Figure 4.2 shows the time taken by different no. of PCs to calculate π.

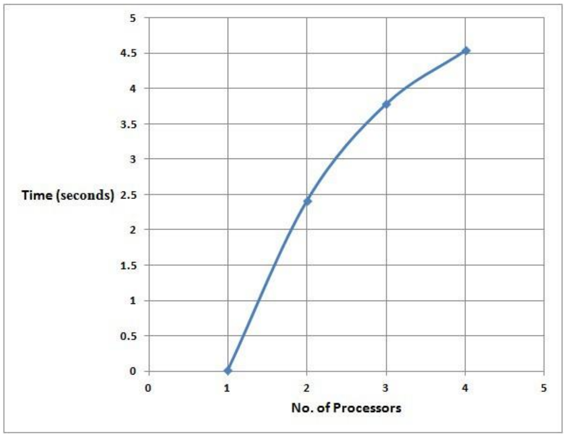

Figure 4.2: Graph showing performance for Calculating π

Circuit Satisfiability Problem- CSAT is a C program which demonstrates, for a particular circuit, an exhaustive search for solutions of the circuit satisfy problem. This version of the program uses MPI to carry out the solution in parallel. This problem assumes that a logical circuit of AND, OR and NOT gates is given, with N binary inputs and a single output. Determine all inputs which produce a 1 as the output.

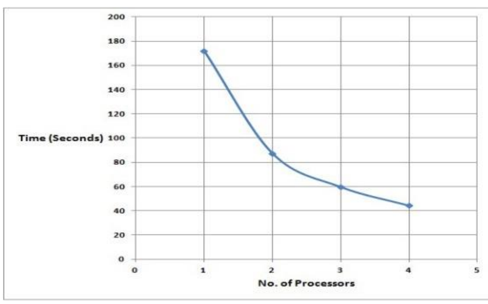

Figure 4.3: Graph showing performance for solving C-SAT Problem.

1D Time Dependent Heat Equation- The heat equation is an important partial differential equation which describes the distribution of heat (or variation in temperature) in a given region over time. This program solves

$$\frac{\partial u}{\partial t} - k * \frac{\partial^2}{\partial x^2} = f(x, t)$$

over the interval [A,B] with boundary conditions

$$u(A, t) = uA(t),$$
$$u(B, t) = uB(t),$$

over the time interval [$t0$, $t1$] with initial conditions

$$u(x, t0) = u0(x)$$

Fast Fourier Transform- To make the DFT operation more practical, several FFT algorithms were pro- posed. The fundamental approach for all of them is to make use of the properties of the DFT operation itself. All of them reduce the computational cost of per- forming the DFT on the given input sequence.

$$n = e{-}j2\pi kn/N$$

This value of $W\,n$ is referred to as the twiddle factor or phase factor. This value of twiddle factor being a trigonometric function over discrete points around the 4 quadrants of the two dimensional plane has some symmetry and periodicity prop- erties.

Symmetry Property: $W\,k{+}N/2 = -W\,k$
Periodicty Property: $W\,k{+}N = W\,k$

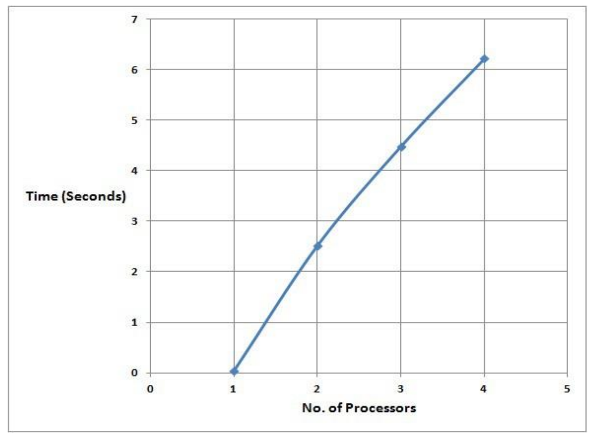

Figure 4.6: Graph showing performance Radix-2 FFT algorithm

Benchmarking- It is generally a good idea to verify that the newly built cluster actually can do work. This can be accomplished by running a few industry accepted benchmarks. The purpose of benchmarking is not to get the best results, but to get consistent repeatable accurate results that are also the best results.

HPL- HPL (High Performance Linpack) is a software package that solves a (random) dense linear system of equations in double precision (64 bits) arithmetic on distributed- memory computers. The performance measured using this program on several computers forms the basis for the Top 500 super computer list**.**
N

V. RESULTS AND APPLICATIONS

Clusters effectively reduce the overall computational time, demonstrating excellent performance improvement in terms of Flops.Finally, performance on clusters may be limited by interconnect speed. Finally, performance on clusters may be limited by interconnect speed. The choice of which interconnect to use depends more on whether inter-server communications will be a bottleneck in the mix of jobs to be run.

Application-

1.Cost-effective: Built from relatively inexpensive commodity components that are widely available.

2.Keeps pace with technologies: Use mass-market components. Easy to employ the latest technologies to maintain the cluster.

3.Flexible configuration: Users can tailor a configuration that is feasible to them and allocate the budget wisely to meet the performance requirements of their applications.

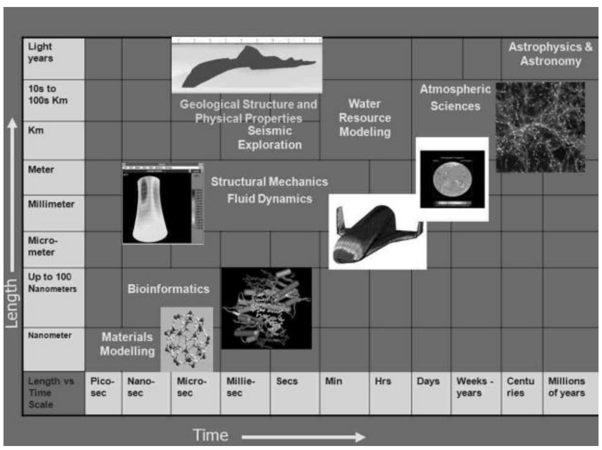

Figure 5.1: Application Perspective of Grand Challenges

4.Scalability: Can be easily scaled up by adding more compute nodes.

5.Usability: The system can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.

Almost everyone needs fast processing power. With the increasing availability of cheaper and faster computers, more people are interested in reaping the techno- logical benefits.

There is no upper boundary to the needs of computer processing power; even with the rapid increase in power, the demand is considerably more than what’s available.

1.**Scheduling: Manufacturing:** Transportation (Dairy delivery to military de- ployment); University classes; Airline scheduling.
2.**Network Simulations:** Power Utilities, Telecommunications providers simu- lations.

3.ComputationalElectroMagnetics: Antenna design; Stealth vehicles; Noise in high frequency circuits; Mobile phones.

4.Environmental Modelling-Earth/Ocean/Atmospheric Simulation: Weather forecasting, climate simulation, oil reservoir simulation, waste repository simulation

5.Simulation on Demand: Education, tourism, city planning, defense mission planning, generalized flight simulator.

VI. CONCLUSION AND FUTURE WORK

Conclusion-

The implemented HPCC system allows any research center to install and use a low- cost parallel programming environment, which may be administered in an easy- to-use basis even by staff unfamiliar with clusters. Such clusters allow evaluating the efficiency of any parallel code to solve the computational problems faced by the scientific community. This type of parallel programming environments are expected to be subject to a great development efforts within the coming years, since an increasing number of universities and research centers around the world include Beowulf clusters in their hardware. The main disadvantage with this type of environment could be the latency of the interconnections between the machines. This HPCC can be used for research on object-oriented parallel languages, recursive matrix algorithms, network protocol optimization, graphical rendering etc. Also it can be used to create college’s own cloud and deploy cloud applications on it, which can be accessed from anywhere outside world just with the help of web browser. Computer science and Information Technology students will receive extensive experience using such cluster, and t is expected that several students and faculty will use it for their project and research work.

Future Work-

As computer networks become cheaper and faster, a new computing paradigm, called the Grid, has evolved. The Grid is a large system of computing resources that performs tasks and provides to users a single point of access, commonly based on the World Wide Web interface, to these distributed resources.

Additionally, the HPCC can be used to create cloud applications and give actual experience of this very booming technology to students. The advantages of cloud computing could work in the students advantage when it comes to getting hands-on experience in managing environments. Before virtualization, it would have been impossible for an individual student to practice managing their own multiple-server environment. Even just three servers would have cost thousands of dollars in years past. But now, with virtualization, it takes just a few minutes to spin up three new VMs. If a college were to leverage virtualization in its classroom, students could manage their own multi-server environment in the cloud with ease. The student could control everything from creation of the VMs to their retirement, giving them great experience in one of the hottest fields in IT.

Appendix A PuTTy

PuTTY is a free and open source terminal emulator application which can act as a client for the SSH, Telnet, rlogin, and raw TCP computing protocols and as a serial console client. The name ”PuTTY” has no definitive meaning, though ”tty” is the name for a terminal in the Unix tradition, usually held to be short for Teletype.

PuTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems. Official ports are available for some Unix- like platforms, with work-in-progress ports to Classic Mac OS and Mac OS X, and unofficial ports have been contributed to platforms such as Symbian and Windows Mobile.

PuTTY to connect to a remote computer-

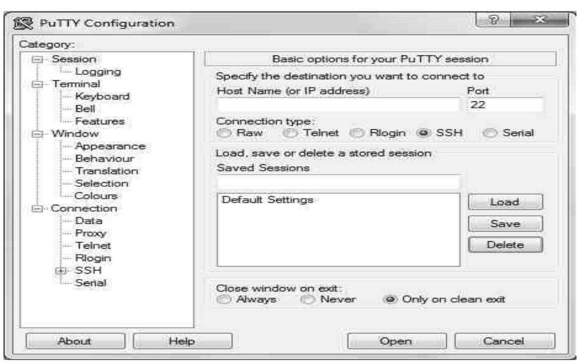

Figure A.1: Putty GUI

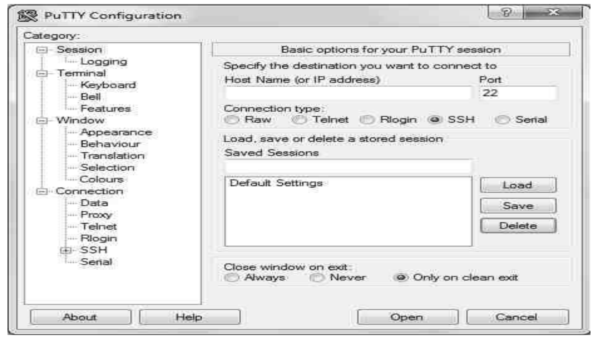

Figure A.2: Putty Security Alert

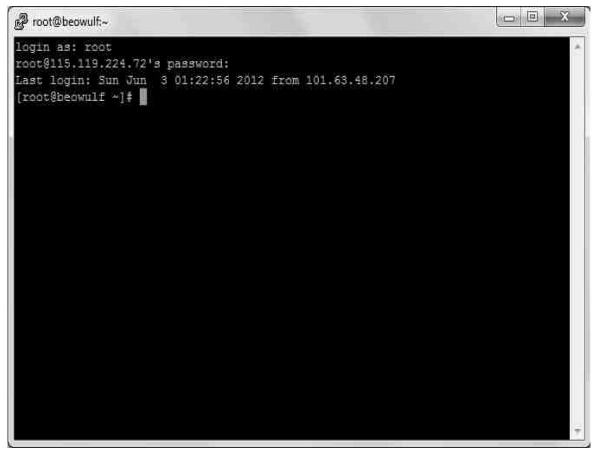

Figure A.3: Putty Remote Login Screen

ACKNOWLEDGMENT

I sincerely acknowledge the efforts of Dr. Rajendra Kumar, in guiding me to the successful completion o this thesis, without him this thesis would not have seen the light of the day. Prof. Dr. Rajendra Kumar made available all possible resources required throughout the progess of my thesis. It was his vision to entitled “**Design and Implementation of High Performance Computing Cluster for Educational Purpose**” . The numerous days I have spent at work had been very stimulating and enriching, thanks to wonderful colleagues I have been privileged to interact with. I have enjoyed many a conversation with the inhabitants of NITM and would especially like to thanks, Prof. Dr. Rajendra Kumar. I am also thankful to Nagaji Institute of Technology & Management of providing me to the top notch resources required for completing my project.

Finally, I am eternally indebted to my parents for their love, blessings, encouragement sacrifice and support. They are my first teachers after I came to this world and had set great examples for me about how to live, study and work.

REFERENCES

[1] Christian Vecchiola, Suraj Pandey, and Rajkumar Buyya : High-Performance Cloud Computing: A View of Scientific Applications at Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN 2009, IEEE CS Press, USA), Kaohsiung, Taiwan, December 14-16, 2009

[2] Luiz Carlos Pinto, Luiz H. B. Tomazella, M. A. R. Dantas : An Experimental Study on How to Build Efficient Multi-Core Clusters for High Performance Computing at *2008 11th IEEE International Conference* on Computational Science and Engineering.

[3] IkerCastaos, IzaskunGarrido, AitorGarrido, GorettiSevillano: Design and Im- plementation of an easy-to-use Automated System to build Beowulf Parallel-Computing Clusters at University of the Basque, *IEEE International Confer- ence* 2009

[4] Azzedine Boukerche Raed Al-Shaikh and Mirela Sechi Moretti Notare :To- wards Building a Highly-Available Cluster Based Model for High Performance Computing at Proceedings *20th IEEE International Parallel and Distributed Processing Symposium* 2006

[5] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. : Above the Clouds: A Berkeley View of Cloud computing. Technical Report No. UCB/EECS-2009- 28, University of California at Berkley, USA, Feb. 10, 2009.