Parallel & Distributed Computing (PDC) Using Low Cost, Compact and Portable Raspberry Pi Mini Cluster: Pradyut

Deepak Aggarwal
Computer Division
Institute for Plasma Research
Gandhinagar, Gujarat, India
deepakagg@ipr.res.in

Prashant Kumar
Computer Division
Institute for Plasma Research
Gandhinagar, Gujarat, India
prashantk@ipr.res.in

Harish Charan
Chemical & Biological Physics
Weizmann Institute of Science
Rehovot, Israel
charan.harish@gmail.com

Hemant Joshi
Computer Division
Institute for Plasma Research
Gandhinagar, Gujarat, India
hjoshi@ipr.res.in

***Abstract*—Scientific discoveries via High-Performance Parallel Computing have come of age and resulted in unprecedented growth in all areas of science. It is highly desired for the academia to utilize the power of parallel computing, specifically using the cluster-based High-Performance Computing (HPC) facilities. Therefore, there exists a great need for skilled and educated students who are able to design, implement and run such HPC facilities. However, the lack of infrastructure, experience and the high cost of setting up an HPC facility make it practically very difficult to teach Parallel & Distributed Computing (PDC) to research students in the academic institutes, especially in India.**

The main objective of the work presented here is to explore an alternative approach to the big HPC facilities which are affordable, easy to set-up, compact, portable and require no maintenance. Small single board computer (SBC) represents a good basis for cluster formation and support for parallel computing. The work presented here demonstrates the implementation of a 4-node working cluster named as Pradyut, using Raspberry Pi boards. The test problems highlight the parallel capabilities of the cluster and demonstrate that the cluster does display the well-known behaviour of PDC systems.

Keywords-PDC; Raspberry Pi; HPC; Cluster;

I. INTRODUCTION

The importance of High-Performance Computing (HPC) has increased over the years in almost all domains of science. Many countries have set up big HPC facilities to fulfil their scientific computational needs. As a consequence, there is a huge demand for skilled professionals and students having the knowledge of Parallel & Distributed Computing (PDC) to utilize these HPC facilities. At present, the access to big HPC clusters is very limited. Considering the cost of setting up HPC facilities, most of the universities are unable to provide their students hands-on experimenting with the cluster design, set up and the administration of such systems. This forces the universities to exclude the lab sessions from their HPC academic curriculum.

With 4 cores and 1 GB RAM as well as the ability to connect to the Ethernet network, small SBC namely Raspberry Pi is a good candidate for the formation of the appropriate computer cluster. The exceptionally affordable Raspberry Pi is easy to set-up, compact and provides a great opportunity to teach PDC to the students on a fully operative small-scale HPC cluster. Performance wise it cannot compete with much larger systems but it explores the principles of HPC and covers most aspects of an HPC curriculum.

This paper presents the implementation of a 4-node working cluster named as Pradyut using Raspberry Pi boards. High Performance Linpack (HPL) benchmark has been used to measure the computing power of Pradyut. A case study has been carried out to analyze the speedup and efficiency of Pradyut which will help the students to understand the concepts of PDC.

II. IMPLEMENTATION STEPS OF 4-NODE WORKING CLUSTER

A. *Pradyut Overview and Components Details*

Fig. 1 shows the assembled components of the Pradyut cluster. It consists of one master node (pi-master) and three compute nodes (pi01, pi02 and pi03) connected through a 5-port switch. The 4 nodes and the switch are powered by a 5-port USB 2A power hub. A Frankenstein Cable has been made from unused cables for connecting the switch to the power hub. The cluster also equips an active cooling fan to cool the nodes. Pi-master acts as the gateway for accessing the cluster. The hardware components required to assemble cluster are given in TABLE I. Most of the components can be easily bought from online stores (Amazon/Flipkart).

B. *Pradyut Cluster Setup*

Here we describe the required steps followed to set the Pradyut cluster up and running. The setting up of the cluster took around 1 week.

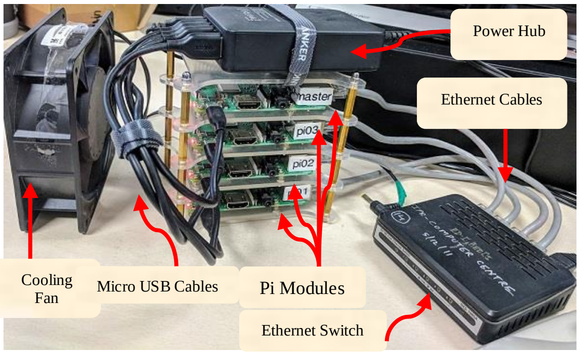

Figure 1. Assembled 4-node Raspberry Pi cluster, Pradyut.

TABLE I. COMPONENTS DETAILS OF PRADYUT CLUSTER

Name	**Table Column Head**		
	Description	***Quantity (Nos.)***	***Total Cost (INR)***
Raspberry Pi Module 3 B +	1.4GHz 64-bit quad-core processor, 1GB RAM, dual-band wireless LAN, faster ethernet.	04	14600
Micro SD Card	16GB (3) and 32GB (1)	04	2300
USB Power Hub	Anker 60W 12A 6-port USB charger	01	4999
USB to Micro USB Cable	Anker 4-pack PowerLine micro USB cables (0.5 ft)	04	1594
Raspberry Pi Case	Transparent 4-layer acrylic cluster case	01	2044
HDMI to VGA Converter	HDMI Male to VGA Female converter adapter for connecting to monitor	01	399
5-port Ethernet Switch	D-link 5-port 10/100 fast ethernet switch	01	Nil*
Ethernet Cables	Cat-6 ethernet cables	04	Nil*

*These components are taken from the spare materials lying with the Network Team of Computer Center, IPR.

1) Master Node and Compute Nodes Setup: This includes installing all the software typically found in a cluster and configuring network for the communication. The micro SD cards are used to house the operating system (OS) on each node of the cluster. The idea used here is to first configure the master node and then clone its SD card to configure remaining three compute nodes. To set up our master node, we started by first installing latest version of Raspbian (https://www.raspberrypi.org/downloads/) on its SD card. Then it was boot up and connected with a monitor to get the desktop view of OS. The default login username is pi with password raspberry. The hostname raspberrypi is changed to pi-master. For setting up the compute nodes, the image of master node SD card is cloned to other three SD cards. The default hostnames of compute nodes are changed to pi01, pi02 and pi03 to avoid conflicts in network. All four nodes are booted up and connected to the network switch. Pi-master has been assigned one public IP for providing access to the cluster from outside world and one private IP for internal communication with pi01, pi02 and pi03, each of which has unique private IPs. For providing passwordless internal communication, SSH keys for each node were generated and then put each key to each node of the cluster for authorization.

2) Network File System (NFS) Setup & Software Environment for PDC: NFS allows cluster nodes to share files over a network. We have configured pi-master to be NFS server and pi01, pi02 and pi03 as NFS clients. A 16GB pen drive connected to pi-master is used as a shared directory across the nodes. To understand PDC, two standard parallel programming modes used in HPC viz. MPI and OpenMP need to be understood. OpenMP is available by default in the OS with the GNU compiler (gcc) and we have opted to use MPICH-3.1 for MPI implementation.

III. PERFORMANCE TEST RESULTS

A. *HPL Benchmarking*

The HPL benchmark is used to benchmark the cluster performance in various numbers of nodes. HPL-2.1 was installed in the shared directory. The cluster performance results for several problem sizes (N) are given in Fig. 2.

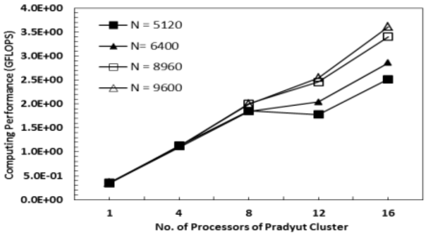

Figure 2. Computing performance versus the no. of processors of Pradyut.

B. *Case Study: Speedup and Efficiency of Pradyut for Computing π*

Here we have analyzed the speedup and efficiency of a program (in serial and MPI modes) that computes π using integration approach. The results are shown in Fig. 3.

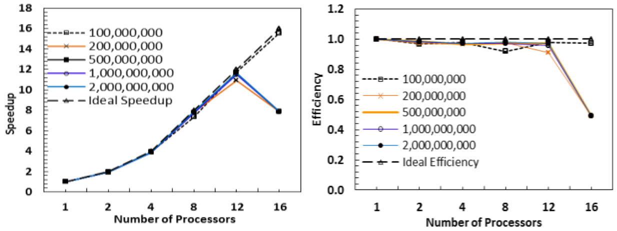

Figure 3. Speedup and Efficiency of Pradyut cluster for π calculation.

IV. CONCLUSION

In this paper, we have presented implementation of the 4-node working cluster using Raspberry Pi boards with performance results. We believe this affordable & portable cluster can be used to teach PDC. The authors also ensure that the work presented here is reproducible.

ACKNOWLEDGMENT

The authors warmly thank Dr. R. Ganesh for believing in the project and acknowledge the Computer Center, Institute for Plasma Research for providing the necessary materials for the project.