Wireless clustering using Raspberry pi

Ranjitha U.N, ranjitha.un@revainstitution.org
Asst Prof Dept. of CSE
Reva University
Bangalore

Meghana, Narendra,Sagar, Shridhar,
Dept. of CSE,
Reva Institute of Technology and Management
Bangalore

***Abstract*—A wireless cluster of Raspberry Pi computers has been created to demonstrate the effectiveness of small computers in parallel processing. The pi's use MPICH, a message passingprotocol to share a large task and then coordinate their results atthe end of the processing among a group ofRaspberry Pi. APN (Access point Network) has been created to connect and communicate each Raspberry Pi devices.**

Keywords: Distributed Computing, Raspberry Pi, supercomputer, technology demonstration.

I. INTRODUCTION

The cluster computing is a technique where more nodes connect together to accomplishes the same task. It can be viewed as the single system. In cluster all nodes will be connected to a main device using wired/wireless communication. Using software like MPI the clusters doe's job of parallel processing which reduces the time taken to complete the tasks. Cluster computing is widely used in the areas like networking servers for high performance, colleges and institutions to demonstrate the technology. The aim of this is to provide a small clustering technique using the wireless technology with help of Raspberry Pis.

Raspberry Pi is a device which in small size that runs the Linux OS (operating System). The device has the ARM Processor which runs the OS. . In our case we used the Raspberry Pi B model which has the processor speed 700MHz, 512MB RAM[4]. Raspberry Pi runs on the Raspbian OS which is not the fastest Operating system but it's close to the Ubuntu. It is less expensive compared to other devices so that we can make high performance computer cluster.

The MPICH is software version of MPI (Message Passing Interface) which supports message passing protocols for communication between the computers[4]. MPICH supports compilers for FORTRAN, C, and C++. In our case we chose FORTRAN as our language. MPICH starts all of the programs on as many processors as specified and runs all to completion[5]. Every processor has the information's about their task and the array index values in the processor arrays.Some distribute the tasks by sending data to each processor and waiting for the "master" processor to receive data from all the slaves.

II. BACKGROUND RESEARCH

The literature shows that researches define the computer clustering technology [1]. They implemented the fully functional computing cluster using Rocks (OS) with Condor batch job system. Joshua Kiepert creates the Beowulf cluster using the Raspberry Pi [2]. The project was to implement the cluster technique on Raspberry Pi to provide high performance in the low cost [3].Implemented a cluster using the support for the serial peripheral interface and exchanging of data according to TDMS and random access MAC protocols.

III. METHEDOLOGY

A. Installing Raspbian os

- Download the latest Raspbian OS and put it in SD Card using Win32diskImager utility tool. Put the SD Card in Raspberry Pi device and turn on the device[8]
- Upon first boot click on the menu icon in the upper left-hand corner and go to Preferences>Raspberry Pi configuration.
- In Raspberry Pi configuration expand the file system, change the hostname, setup the keyboard layout and finish the configuration and reboot the device[7].

B. Installing MPICH

Open the terminal and follow the commands.

- mkdir mpich2
- cd ~/mpich2
- wget http://www.mpich.org/static/downloads/3.1/mpich-3.1.tar.gz
- tar xfz mpich-3.1.tar.gz[7]
- sudomkdir /home/rpimpi/
- sudomkdir /home/rpimpi/mpi-install
- mkdir /home/pi/mpi-build
- cd /home/pi/mpi-build
- sudo apt-get install gfortran
- sudo/home/pi/mpich2/mpich-3.1/configure-prefix=/home/rpimpi/mpi-install
- sudo make

- sudo make install
- nano .bashrc
- PATH=$PATH:/home/rpimpi/mpi-install/bin
- sudo reboot

C. Installing Python to MPI Interpretor

Open the terminal and follow the commands

- sudo aptitude install python-dev
- wgethttps://bitbucket.org/mpi4py/mpi4py/downloads/ mpi4py-2.0.0.tar.gz[10]
- tar -zxf mpi4py-2.0.0
- cd mpi4py-2.0.0
- python setup.py build
- python setup.py install
- export PYTHONPATH=/home/pi/mpi4py-1.3.1
- mpiexec -n 5 python demo/helloworld.py

D. Configuring the remaining Raspberry Pi's

- Connect all Raspberry Pi devices using AP(Access Point) of Master Pi
- Find the current IP address for the master Pi (ifconfig)
- Then find other devices IP address using nmap software
  - sudonmap -sn 192.168.*.*
- On the master Pi,create a new textfile called "machinefile"
  - nanomachinefile
- At this point, pi's could run a test MPICH for all the devices connected within the network.
  - mpiexec -machinefilemachinefile -n 2 hostname

Verifying Host Keys

- On the Master Pi, in the default home folder, create a new key.
  - ssh-keygen
- Navigate to the ssh folder and copy the key file to a new file called "masterpi"
  - cd .sshcp id_rsa.pub masterpi
- Next connect via SSH into Pi1 and repeat those same steps to create a Pi1 key file
  - sshpi@IP address of Pi1
  - ssh-keygen
  - cd .ssh
  - cp id_rsa.pub pi1
- Before exiting out of Pi1, copy the masterpikey file over to it and authorize it.
  - scp 192.168.43.223:/home/pi/.ssh/masterpi .cat masterpi>> authorized keys exit
- After generating keys for each of the Pi's, exit back to Master Pi and copy over all of the keys generated on each of the Pi's.
  - cp 192.168.43.223:/home/pi/.ssh/pi1
  - cat pi1 >>authorized keys
- Running a Program on the Supercomputer
  - mpiexec -machinefilemachinefile -n 2 python python_test/md5_attack.py[9]

IV. Software and Hardware Requirements

A. Software Requirements

- Raspbian Jessie OS
- Python Compiler
- MPICH
- Win32DiskImager
- Putty open-source terminal emulator

B. Hardware Requirements

- 3 X Raspberry pi 2, pi 3 Model,
- SD Card for each Pi (8 GB Minimum)
- Wi-Fi Dongle/Router
- Powered USB Hub (optional)

Conclusion

This whole idea makes an cost effective way to implement a supercomputer which is around 70 time cost effective than the supercomputer which we get in the market. As it also allows roaming it is very easy to configure in a new environment. Live monitoring the pi's allows the user to be keep the data safe at all conditions.

References

[1] Jennifer Helsby,Rafael David Pena Rocks Cluster Computing,Undergraduate Research, Fall 2006.

[2] Joshua Kiepert,Creating a Raspberry Pi-Based Beowulf Cluster, Updated: May 22nd, 2013

[3] Rajkumar Buyyaa,Hai Jin b,Toni Cortes,Cluster computing,Monash University, Melbourne

[4] CREATING A LOW-COST SUPERCOMPUTER WITH RASPBERRY PI,Shaun Franks,Johnathan Yerby,Middle Georgia State College

[5] https://en.wikipedia.org/wiki/MPICH

[6] Luca De Nardis, Maria-Gabriella Di Benedetto,Wireless Access Networks for Raspberry Pi ,DIET Department, Sapienza University of Rome, Italy

[7] https://www.instructables.com/id/How-to-Make-a-Raspberry-Pi- SuperComputer

[8] https://www.raspberrypi.org/downloads/

[9] http://stackoverflow.com/questions/13896502/mpiexec-was-unable-to-launch-the-specified-application-raspberry-pi

[10] https://pypi.python.org/pypi/mpi4py

[11] https://learn.adafruit.com/adafruits-raspberry-pi-lesson-3-network-setup/setting-up-wifi-with-occidentalis

[12] http://makezine.com/projects/build-a-compact-4-node-raspberry-pi-cluster/