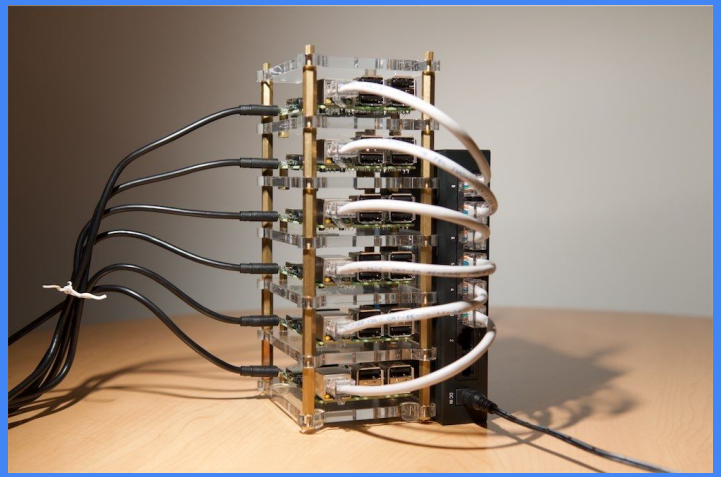
Setting Up Pi Clusters

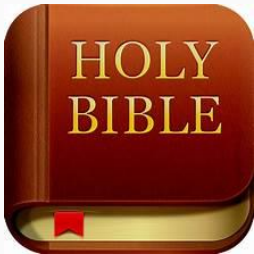
By Ben Peterson



Verse of the Day

Proverbs 3:5-7(NIV)

Trust in the Lord with all your heart and lean not on your own understanding; in all your ways submit to him and he will make your paths straight. Do not be wise in your own eyes; fear the Lord and shun evil.

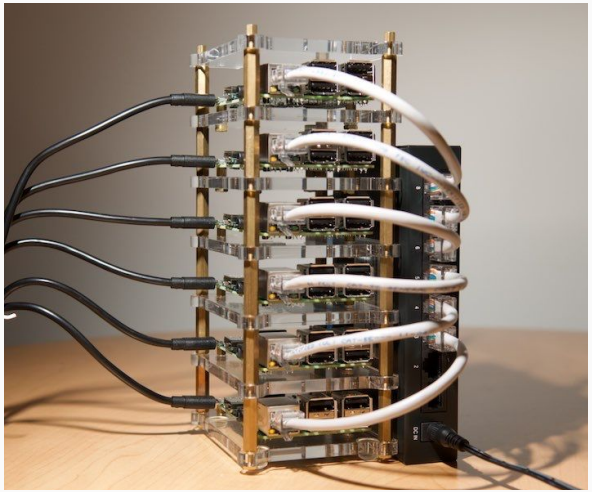


Overview

- Basics of Raspberry Pi
- Raspberry Pi installation
- Setting up Pi’s in a cluster
- Setup DHCP server to connect Pis

What is a Pi Cluster?

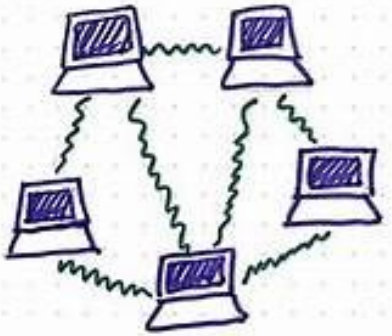
- Group of connected Raspberry Pi's acting as one system
- Used to explain basic concepts of HPC
- Fun to mess around with!



(Pi cluster)

Why Pi Cluster?

- Increased performance
- Fault tolerance- If one process fails on one node, other nodes can still function
- Load balancing- Makes sure one node isn't overwhelmed while the other is doing nothing
- Scalability/Affordability- Pis are much cheaper than servers
- Parallel/Distributed Computing- Tasks can be divided into smaller tasks



Resources Needed

- 2 or more Raspberry Pi
- 2 or more SD cards
- A cheap gigabit switch to plug all Pi’s together
- Power cables for Pi’s
- A network cable for each Pi

Raspbian Installation

1: Install Raspbian (or other linux) on one or more of the Pi's micro SD card using the Raspberry pi imager

2: Plug in micro SD, ethernet, and power to each pi

3: Power on the pi you want to be the master (more on that later)

How do I make them communicate?

- Physical Ethernet Connections- Allows devices to find each other over ethernet (can work wirelessly, not recommended)
- MPICH - Lightweight MPI, defines protocols for parallel and distributed computing
- SSH- Established secure connection between nodes, allows remote access of nodes + extra security.

Getting Started

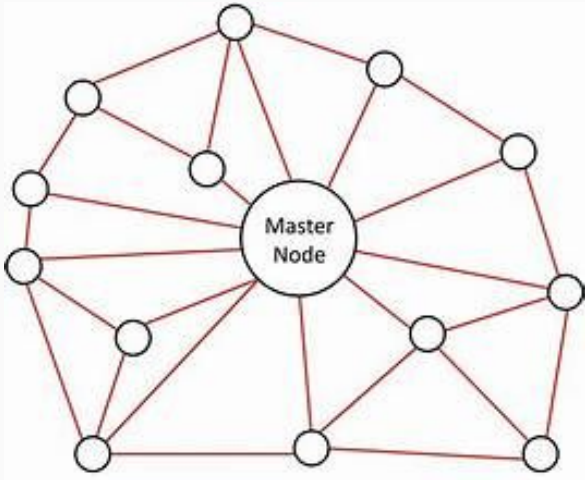
- Get the IP of your master Pi
- First, you need to update and install software, using these commands:
  - sudo apt update- updates the package source list and gets the latest list of available packages (identifies updates)
  - sudo apt upgrade- downloads package information from all available packages (downloads updates)
  - sudo apt install mpich- downloads latest version of mpich, free version of MPI which passes messages for parallel computing

More installs

- **sudo apt install python-pip python-dev libopenmpi-dev** - installs python and other resources needed to run scripts parallel.
- **sudo pip install mpi4py** - Allows you to work with MPI in python
- **sudo apt install dnsmasq** (Only used for master)- Installs lightweight version of DNS and DHCP, which allows you to access websites through URL and assign IP addresses to nodes

Setting Up a Master Node

- What is a master node?
  - Manages cluster and divides tasks between nodes. Think of it like the manager
- Use nmap command to get the IP address of the Pi's on the network
  - nmap -sP IP.* | grep master
- Set a static IP address for the master node
- Reboot (subo reboot): needed to apply new changes



DHCP Setup

- Reminder: DHCP gives out IP address to all the nodes on the network
- Use this command to set up DCHP (opens path to DNS config file with supervisor privileges)
    - sudo nano /etc/dnsmasq.conf

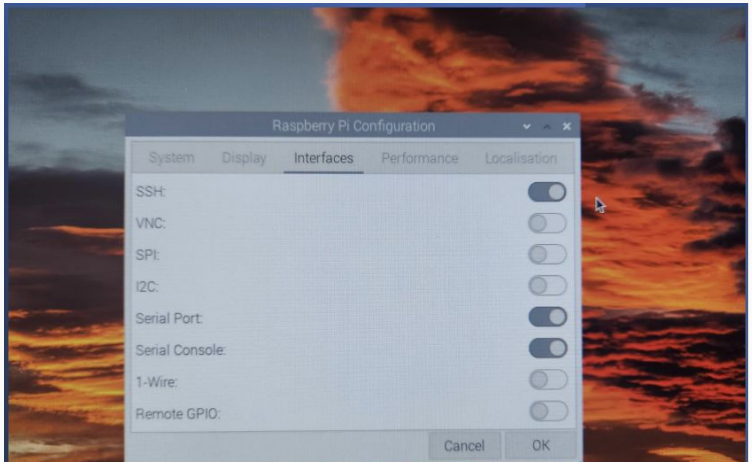
DHCP Setup continued

Within nano:

-interface=eth0 - specifies device on port 0 to listen on for DHCP and DNS requests
-bind-dynamic -
-domain-needed - makes sure DNS requests are only sent to valid IP's
- bogus-priv- Causes dnsmasq to reject private IP ranges that can't be used on the internet
-dhcprange=192.168.1.3,192.168.1.10,255.255.255.0,6h - Tells master to assign IP addresses within the covered range, separated by comma, hour time at end specifies lease time(how long before IP needs to be renewed or released

What now?

- Log onto second pi
- Ping the ip address of the master pi to test connection
- Enable ssh on each pi by going into the main OS, then go to configuration, then enable ssh
- Set up ssh



SSH setup

Start the SSH Service:

- After installation, start the SSH service. This allows the computer to accept SSH connections from other devices.

Configure SSH (Optional):

- While SSH usually works with default settings, you can configure it further based on your requirements. Configuration files are typically found in `/etc/ssh/`. You can make changes to these files to enhance security or accommodate specific needs.
- **Allow Firewall Access:**
- If you have a firewall enabled on your computer, ensure that it allows incoming connections on the SSH port (default is 22). This step ensures that external devices can connect to your computer using SSH.

Test SSH Connectivity:

- Verify that SSH is working by attempting to connect to your computer from another device. Use the `ssh` command followed by the username and IP address or hostname of your computer. You should be prompted for authentication (usually a password or key), and upon successful authentication, you'll gain access to your computer's command line.

Repeat for each node:

- If you want to enable SSH on multiple computers in a network or cluster, follow the above steps on each computer to ensure that SSH is enabled and configured consistently.

Questions?

Any questions?