Using Panucci

March 9, 2017

1 What is Panucci

Panucci is a network-based system testing and imaging platform.

Panucci is a collection of multiple open-source and free software packages, custom built for our particular use case - multiple images going onto multiple computers at once.

2 Installing Panucci

The simplest way to set up Panucci is to use Ubuntu Server and DRBL (Diskless Remote Boot for Linux). A standard PXE environment is also viable, but is outside the scope of this document. If you will be installing Panucci on a standard PXE environment, skip to the appropriate section.

2.1 Overview of requirements

- PXE Server
  - Tested using Ubuntu Server and DRLB (Diskless Remote Boot for Linux)
  - Any PXE server will work
- NFS server (for images)
  Note: Using DRBL will place the NFS server on the same machine as the PXE server.
- Software required on the PXE image
  - Linux
  - Ruby
    - Sinatra gem
    - dotenv gem
  - A web browser
    - Chromium (tested)
    - Firefox (untested, but should work)
  - emtester
  - Clonezilla
  - NFS (for images)
  - dmidecode
  - sudo
  - smartmontools (specifically, `smartctl`)
  - seeker (a small utility used to test seek time on drive)
  - i3 Window Manager (this can be swapped out)
  - xterm (this can be swapped out)
  - rerun (this can be left out, follow notes for that case)

3 Hardware

The hardware requirements for Panucci are meager, but better hardware, especially networking and storage, will greatly improve the performance of the platform. While there are no defined minimums, standard server hardware serves as a solid starting point.

3.1 Storage

Many computer images require between 20 and 40GB of storage per image. While Panucci is able to handle multiple images for a single device line with little overhead over the original image, the storage requirements can quickly become cumbersome without proper planning.

3.1.1 Storage Space

To calculate the amount of storage needed to comfortably serve your entire line, calculate the maximum expected size of each image (in this example, 40GB) and multiply it by the number of product lines you expect to carry.

(40GB) * (40 lines) = **1.6TB** of total storage

As an added buffer, it is recommended to add an additional 25% to the pool to account for new product lines in the future.

3.1.2 Storage Speed

Storage speed requirements will vary based on the number of devices you will be imaging simultaneously. To determine speed requirement, multiply the number of machines you expect to image simultaneously by the expected "pull speed". If you will be using gigabit network connections (which is the limit of most current hardware), figure on approximately 115MB/s per unit.

(16 simultaneous units) * (115MB/s per unit) = **1840MB/s**

This number assumes that all units will be "pulling" at their network-limited speed of 115MB/s, which many devices simply will not be capable of, so there is already some degree of building for growth included.
Additionally, the calculated number is the theoretical threshold to have all

units imaging at full speed. If a lower speed to image is acceptable, lowering the access speed is a valid step to reduce implementation costs.

3.1.3 Recommendations

There are some standard methods to help improve your storage for Panucci.

- Consider using RAID (especially RAID 1+0). Properly configured, RAID can offer significant performance and resiliency advantages. If hardware RAID is not available, LVM or software RAID are viable alternatives, but you will see a performance hit from their use. Keep in mind that using RAID levels other than RAID 0 will necessarily reduce usable space, but are much more reliable and fault-tolerant. **Due to the high risk of data loss, using RAID 0 alone is not recommended under any circumstance.**
- Consider splitting out the drives by use. In a multi-array or multi-drive environment, place machine images onto one array or drive (usually as `/home/partimag`) and place the core operating system on another array or drive.
- Consider using Solid State Drives (SSDs) or SAS drives. Both provide a performance gain over traditional SATA drives. SSDs are more expensive per unit of storage, but are usually the fastest option available. SAS drives offer more storage, but are slower.

3.2 Network

Given that Panucci is a network-based imaging platform, it makes sense that it is heavily network-bound.

3.2.1 Network Ports

Ideally, each client machine will operate at the full speed of the network connection (although this is not always the case due to a number of factors). This can simplify the network calculations if both the server and client are running the same speed (e.g., both are using gigabit ethernet). In such instances, you will need as many network ports on the server as you wish to have simultaneous clients, plus an additional port to handle general network traffic.

Example: Gigabit to Gigabit
(16 simultaneous clients) + (1 general network) = **17 total ports**

As a general rule, expect gigabit ethernet on the client devices. Given this, the calculations for higher speed network connections are relatively simple. Keep in mind that to use a certain connection speed, all intermediate hardware must support it (e.g., plugging 8Gb fiber channels into a switch that only supports 4Gb will drop the speed to 4Gb).

Example: Servicing 16 Computers with 4Gb Fiber
16 Clients = **16Gbps**
16Gbps/4Gbps = **4 connections**
Result: You will need 4x 4gbps connections to simultaneously service 16 machines, plus the connection to the outside network.

3.2.2 Recommendations and notes

- When possible, offer every client the full extent of its network capabilities. For most modern machines, this means allowing each client 1Gbps.
- If providing each client its full network capacity is unfeasible, aim to minimize the reduction, as a reduction in capacity will inversely impact the imaging process (i.e., half speed means twice the time to clone an image).
- If you will be using a switch, rather than direct connections, be sure to set up VLANs or similar networking partitioning methods for each port on the server.

4 Infrastructure Installation Procedure

These steps pertain specifically to install the infrastructure supporting Panucci. These steps will get you a working base to install Panucci on. If you already have an NFS and PXE server, skip to "Panucci Installation Procecdure".

4.1 Installing Ubuntu Server

1. Download the image to create Ubuntu Server install media. Images can be found at `https://www.ubuntu.com/server`
2. Create the bootable media. If you are unsure how, follow these instructions using Rufus on Windows.
3. Boot from the media on the device you will be using as a server. The specifics will vary depending on the exact device being used, and are outside the scope of this document.
4. Follow the prompts in the Ubuntu Server installer. It is unnecessary to select any of the package groups other than the base package and, for your optional convenience, OpenSSH (to allow remote access.) For additional assistance installing Ubuntu Server, see Ubuntu's documentation.

4.2 Installing DRBL on Ubuntu 16.04

DRBL (Diskless Remote Boot for Linux) is a package designed to make PXE/diskless booting much easier. While DRBL is not required to use Panucci, it does greatly simplify setting up the infrastructure. Setting up PXE without DRBL is outside the scope of this document.

1. Log in to the server and open a command line. If you are logging in remotely using SSH, you should already have a command line open.
2. At the command prompt, enter the following commands to add the official Universe and Multiverse repositories:

   ```
   sudo add-apt-repository "deb http://us.archive.ubuntu.com/ubuntu/
   xenial universe multiverse"
   ```

   ```
   sudo add-apt-repository "deb http://us.archive.ubuntu.com/ubuntu/
   xenial-updates universe multiverse"
   ```

3. Add the DRBL repositories to the APT sources.list

```
sudo sh -c 'echo "deb http://drbl.sourceforge.net/drbl-core drbl
stable" >> /etc/apt/sources.list'
```

4. Add the DRBL repositories GPG keys by running the following two commands:

```
wget http://drbl.sourceforge.net/GPG-KEY-DRBL
```

```
sudo apt-key add GPG-KEY-DRBL
```

5. Update the repository listings and update the system

```
sudo apt-get update && sudo apt-get upgrade
```

6. Install DRBL

```
sudo apt-get install drbl
```

4.3 Network Configuration

While many network configurations will work, this is our recommended configuration, as it's simple to generate automatically. Feel free to customize the autogeneration script or the layout to fit your environment.
Note: Unless you are very familiar with your network environment, it is strongly recommended to make sure that all the interfaces are isolated from the rest of the network due to DHCP.

1. List all network interfaces in the server. Note the use of pipes in the command. Device `lo` can be ignored, as it is the loopback.

```
ip link show | grep mtu | sed 's/: <..*//g' | sed 's/.*: //'
```

2. Determine which network device is the "external" port (the one reserved for outside network access). Remove it from the list. The easiest way to determine this is to make sure that the only interface connected is the one you intend to use, then run `ifconfig` and look for the interface with an IP address.

3. For each network interface, define it in `/etc/network/interfaces` as follows:

```
auto eth2 //interface name
iface eth2 inet static //set interface to static
address 192.168.43.13 //set static IP address; increment the third
octet (c in a.b.c.d) for each interface.  We've set the fourth
octet (d) to 13, but any number should work.  If you set the interface
to x.x.x.1, be sure to account for that in the DRBL configuration.
netmask 255.255.255.0 // You can almost always leave this alone.
```

   In the above example, `eth2` is the interface name. `static` means to set a static IP address, rather than have the interface use DHCP to get an IP.

Note: There is a network interface generation script included in the Panucci repository. Feel free to use that if you are uncomfortable editing system configuration files by hand. Simply run the script as root (`sudo ruby interfaces.rb`).