

How I migrated my computing environment from a Mac Mini to a Raspberry Pi

This is part 1 of a 3-part series detailing how I moved my computing environment from a Mac Mini PPC to a Raspberry Pi Model 2. In this article I discuss the background and reasons for doing so. It's quite technical in places, but if you're considering a move from an existing system to something decidedly lean and mean there are things that you'll need to know before you make that leap. There are lots of links to click as well, which will lead you to the software and apps that I've mentioned. Enjoy!

Saying goodbye to the Mac

I have to admit, I've never really been an Apple Fanboi, especially following a short and ultimately-unsatisfactory fling with a plastic polycarbonate MacBook back in 2006. Although it was beautifully designed, and the software *Just Worked*, I was understandably upset when it decided to expire shortly after the warranty period expired – design faults apparently. Ah well. I swore never to “invest” in an Apple machine again – until of course I discovered a used Mac Mini PPC on eBay which could be had for around \$100, in 2012. It was new back in 2005, but had apparently been refurbished.

“*What have I got to lose, especially at that price?*”, I asked myself. Nobody answered, so I placed a last-minute bid, won it and invested about the same sum of money again in bumping the memory up to 1Gb and buying the OS on DVD. The OS X version was 10.4.7; Tiger, and the architecture was Power PC (PPC). It was sedate but reliable, and I was happy. It didn't take a lot of power either; some 60 watts at full load, so that was a bonus. I spent many happy hours tinkering with it and trying to find software that was supported on a device that old. Predictably though, as



my computing requirements grew and the Mac got older, it started to get noticeably slower, and I was aware that even simple tasks such as asking it to run a web browser *and* display an HTTPS page were causing it problems. When I finally managed to find anti-virus software for it, I became aware just how noisy the Mini's cooling fan was as the CPU struggled with the extra load. A quick check of the performance monitors revealed thousands of memory paging faults, and I realised that my old friend was soon destined for the knackers yard. Of course, that meant searching for a replacement, and that's when the fun started.

So I had this idea for a(nother) small computer

My main problem was that I didn't have a big budget. I looked at eBay again and found a number of Mac Minis for sale, all around the \$500 mark, and many of those were early basic-spec Intel units that,



like my old Mac, people had simply grown out of. Essentially I wanted something like the old Mini, ideally with similar power consumption. A new one was out of the question, obviously.

Let me state now that my computer requirements are pretty undemanding, and for photo/graphics work I have another computer that consumes power like there's no tomorrow and gives off enough heat to keep me warm in winter.

And then I got to think about the [Raspberry Pi Model 2](#). Now before you laugh, I currently have around 6 of the things running various servers, and they do just fine. Suffice to say that I have one that runs a small web server, another runs a mail server, and so on. Each costs around \$30 and most use a

cheap microSDHC card, so if one fails I can easily swap it out for another; *and* I can usually buy a suitable card at a local supermarket – try doing that when your laptop drive fails!. I also have a Netgear ReadyNAS 102 with a couple of 2Tb hard drives to act as my bulk storage. Suddenly, my plan looked as though it might be viable after all!

Spec'ing it out

The specification was a bit of a no-brainer really: the Model 2 Pi comes with 1Gb of memory as standard, the ethernet runs at 100Mbps maximum, the clock speed is 900Mhz, there are 4 USB ports, and *that's yer lot mate*. You can overclock it, but I've never wanted to try this, for various reasons. I had a Pi in my spares drawer so no problem there. I ordered a posh aluminium case made by [Flirc](#) that was on offer for \$20, and duly slotted in the Pi. The power supply *had* to be a genuine 2amp device, and again, I had a spare lying around. If you're taking your Pi ownership seriously, I'd recommend the [Anker 40W](#) 5-port desktop charger: this has intelligent power management, and I'm running 5 Pi's from one unit. Incidentally, if you inadvertently use a PSU that can't deliver the required current, you'll keep seeing a square, multi-coloured icon appearing in the top right-hand corner of your screen, so be warned. The microSDHC "disk" was more of an issue though. I always use Sandisk, and this time I wanted something fast, especially as this was to be a "desktop" machine. In the end I went for a [Sandisk 8Gb Extreme Pro UHS-1](#) card which promised up to 90/95 Mbps write/read performance. "*8Gb? That's not a lot of space*" I hear you Windows users cry... and because this is Linux, there doesn't need to be.

The way I envisioned it being configured, I'd set up the Pi as normal, and use it primarily as a boot disk. I'd host all my documents and media files on the NAS box, and all would be well. The NAS shares would be accessed via NFS and I'd just mount these as directories on the PI. I elected quite early on to simply move my entire home directory onto the NAS and this has worked well, with some quirks. The problem I faced was a Pi quirk, and although I was sure there was a fix, I wanted to get it up and running before the Mac finally crapped out. When the Pi boots, it seems to enable the networking part quite late in the sequence, and I found that I couldn't do my NFS mounts as the networking interface hadn't come up yet. Rather than hack around with tricky scripts I decided to simply mount the NFS shares by hand once I'd logged in after a successful boot. This seemed to work and it's the solution I'm using now. Now that I had a basic strategy, it was time to implement it on the "live" machine. Oh, and that's the beauty of working with the Raspberry Pi: you can quickly hack together a test bed and have a system up and running in under 30 minutes. As regards video and sound, I bought an HDMI – DVI cable to use with my Dell Monitor, and when in GUI desktop mode this comes up as 1280x1024 – plenty good enough for my use. If you have a monster flat-screen TV you can always use that instead...

Software, software everywhere

As regards software/environment I chose the following:

- **OS:** I ultimately decided on [Arch Linux for ARM 7H](#) as the operating system. I'm actually a [Raspbian](#) veteran, but I simply didn't need the educational software that comes with it (I have other Pi's for that). Arch provides a minimal environment but is full featured, well-supported and powerful: it also has bucket loads of software available for it. Following initial installation, I'd used just over 1.2Gb of space, and even now, with all my necessary software on the microSDHC I'm only using 2.8Gb of my 8Gb card. Please note that the Pi 2 is officially Arch Linux Arm 7, not 6.
- **Desktop:** I wanted a graphical desktop environment (even though I'm a command-line sorta guy), but this needed to be in keeping with the lean/mean ethos. I'd used [LXDE](#) before and was happy with it, so that was installed: Gnome and KDE were just too big.

- **Web Browser:** The web browser was a bit of a problem, but after trying the default Midori, Epiphany and a couple of others I decided on [Firefox](#). It's a bit flabby but it follows standards well, and if you're going to digitally sign LibreOffice ODT documents you'll need it anyway.. One problem on a machine of this power is the tremendous toll that web-based ads place on the overall memory usage. In fact, a badly-ad'ed page can make the browser stop completely, so I had to make those ads disappear. One way would have been to install an ad-blocker plugin, but that's another hit on available memory, so a simpler method was called for.

As this is a Linux box, I simply downloaded an [ad-blocking hosts file](#). This is an amazing piece of community work that consists of over 15,000 hostnames for basically any server that spits out ads. *All* the entries point to an IP address of 0.0.0.0 so there's no time wasted and your bandwidth's your own again. It's a free download and can be added to the end of an existing hosts file. Of course the major value as far as I'm concerned is that page load times are *much* quicker.

The screen capture below shows an ad-free Firefox overlaid with the same page in [eLinks](#).



No ads in either, but if you don't need all the eye-candy rendered by Firefox, eLinks will provide a super-clean experience. (Normally, all that white space in the Firefox image is filled with ads!) The eLinks browser is an interesting hybrid browser that is primarily text-based and is similar to the classic pure-text Lynx browser.

- **Messaging:** It would be overkill, and undesirable from a security point of view to have MS Skype on the Pi, so I decided on a Jabber/XMPP client, [Psi](#). Psi has the advantage of not having a multitude of dependencies and also works really well. It's easy to take part in multi-user chats, and I have another Pi hosting a Jabber server to test it on. There's no character-mode version unfortunately, and most of the text-based clients I tried had problems, so it's a GUI-only situation at the moment. No matter; it works well and doesn't use a lot of resource.

- **Email:** I also tried a number of email applications: this was easily the most important application, and I eventually chose [Claws-mail](#). Sadly it doesn't do HTML mail, but it's rock-solid reliable. I have to say that I can't get the GPG plugin working properly yet due to some unresolved version issues, but I can always encrypt messages in a terminal if need be.
- **Audio:** Music is important to me, and I chose [SMPlayer](#) as my media player. This supports many options including playlists, for local/networked files and Internet Radio streaming. Does the job well.
- **Video:** I'll not go into the video player aspect in any great detail. Bearing in mind the hardware specs of the Pi, reliably playing back a video stream, even on the same network was problematic, and I decided that if I wanted to watch videos I had other devices more suited to it. I did try and experiment with the `gpu_mem` setting in the [PI /boot/config.txt](#), switching it from the default 64Mb to 96Mb: I was prepared to borrow a bit of application memory to loan to the video player but even that didn't seem to make it work well. In the end I kept that setting so that the desktop environment would run more smoothly and so far I haven't had problems. The irony of this is that I have another Pi (!) that has a [DLNA](#) server installed, and this can stream video exceedingly well: not just to one client, but several. In its defence though, it doesn't have a desktop environment to contend with. So for now I don't bother attempting to play video.
- **Image Processing:** I needed simple, lightweight photo/image editing, and I knew from prior experience that GIMP and similar packages would bring the Pi to its knees, so I eventually found an app called [Pinta](#), which resembles an enhanced MS Paint, but with more *cojones*.. As someone with a large image collection I also needed a slideshow application. After much evaluation of software I finally decided on "[feh](#)". Normally run from a terminal within the GUI desktop, it has an incredible array of options that can help you produce an image slide show, and again, it has low memory requirements.
- **Office Suite:** And then there was an Office Suite. On the old Mac Mini I was happily (and legally) running a copy of Microsoft Mac Office 2004, and I was truly sorry to lose that. I only really needed copies of an MS-Word and MS-Excel equivalent, but I had to bear in mind the limitations of the Pi. Sure, there were standalone versions of word processor and spreadsheet applications but there was nothing that really gave me confidence to edit a potentially full-featured document. I already knew of [LibreOffice](#), but I had my doubts about using it as there was a Java Runtime Environment (JRE) requirement, or so I thought. The JRE was thankfully an option, and as long as I didn't want to use (database) connection pooling or macros there was no need to enable it. I also used as many built-in options as possible, rejecting skins and themes: this brought overall memory footprint down to a reasonable level, and hey, I'm writing this on LibreOffice Writer now! I adopted the attitude that if it has a built-in theme, then use it!

Anyway, here's the current [memory overview](#) (in Mb) from within the GUI desktop:

	total	used	free	shared	buff/cache	available
Mem:	891M	165M	99M	7.5M	626M	689M

- **Miscellaneous:** Other desktop software I've installed – not much as I wanted to keep this a minimal installation:
 - [Filezilla](#) – SFTP/FTP client
 - [PuTTY](#) – SSH/telnet terminal front-end
 - [Mousepad](#) – a versatile plain-text editor, similar to Wordpad or Notepad, but much more powerful.

Overall, the entire setup works as intended. I've found that it performs well, if a little slowly sometimes, but this is to be expected as it's running on a Raspberry Pi with, as stated, a 900Mhz clock speed and 1Gb of memory. As long you're aware of and prepared to accept the limitations, you can have a cheap, very functional system that doesn't take up all your desk space.

Lacking in characters

Life with a Pi desktop is not all about the GUI – it's a *very* competent command-line environment too, should you need one. As a Linux developer and geek I am very comfortable in a character-mode environment, and this is where the Pi really comes into its own. The performance you can expect in the command-line environment, at least in my configuration is dependent on a number of factors. I'm limited to a certain extent by the network interface speed of the Pi, and the overall performance of the NAS, which is another slightly-underpowered, consumer-grade ARM box: the Netgear ReadyNAS 102. The one thing that did please me though was the noticeable increase in speed over the Mac Mini!

So running in a native terminal environment, this is the typical memory usage (in Mb) you might expect.

	total	used	free	shared	buff/cache	available
Mem:	912M	31M	260M	0M	621M	855M
Swap:	0	0	0			

One thing to note here is the lack of a swap partition. It's generally accepted that any type of swap system on a Raspberry Pi is a *Very Bad Thing™* and will wear out your SD card in no time. Again, I had considered setting up a swap partition on the NAS box, but this was ruled out early on as it would very negatively impact the network as a whole, and as with the NFS mount issue, the swap partition would need to be mounted before the network came up. So no go. Having lived with Raspberry Pi's for some time now, let's just say that one has to learn to set things up carefully to avoid the need in the first place, and ultimately it can teach you manage computers better.

As part of my efforts to make the Pi as useful as possible, I had to envisage the scenario in which whatever I was working on was either so resource-hungry that I couldn't run a GUI desktop, or the GUI was just not required. That meant reproducing as many of the desktop-only apps in a character-mode environment. In fact, this was actually easier than finding the equivalent desktop apps and so here is my current lineup.

- File manager: [Midnight Commander](#) – if you're old enough to remember the Norton Commander, you'll know what it looks like.
- File transfer: SSH/SFTP - normally handled by PuTTY and FileZilla on the desktop, you just use these 2 commands as provided :)
- Web browser: Lynx or Links – both classic character-mode browsers which significantly speed up the Internet Experience.

- Music player: yes, you can play music in a character-mode terminal! [Mpg123](#) is the name of the app, and when run as "`mpg123 -C`" it'll allow full keyboard control of all playback functions. If you want to be really cool you can alter the way Midnight Commander handles mp3 files by editing `/etc/mc/mc.ext` and adding the code snippet below. This allows you to browse and play back your music collection with ease.

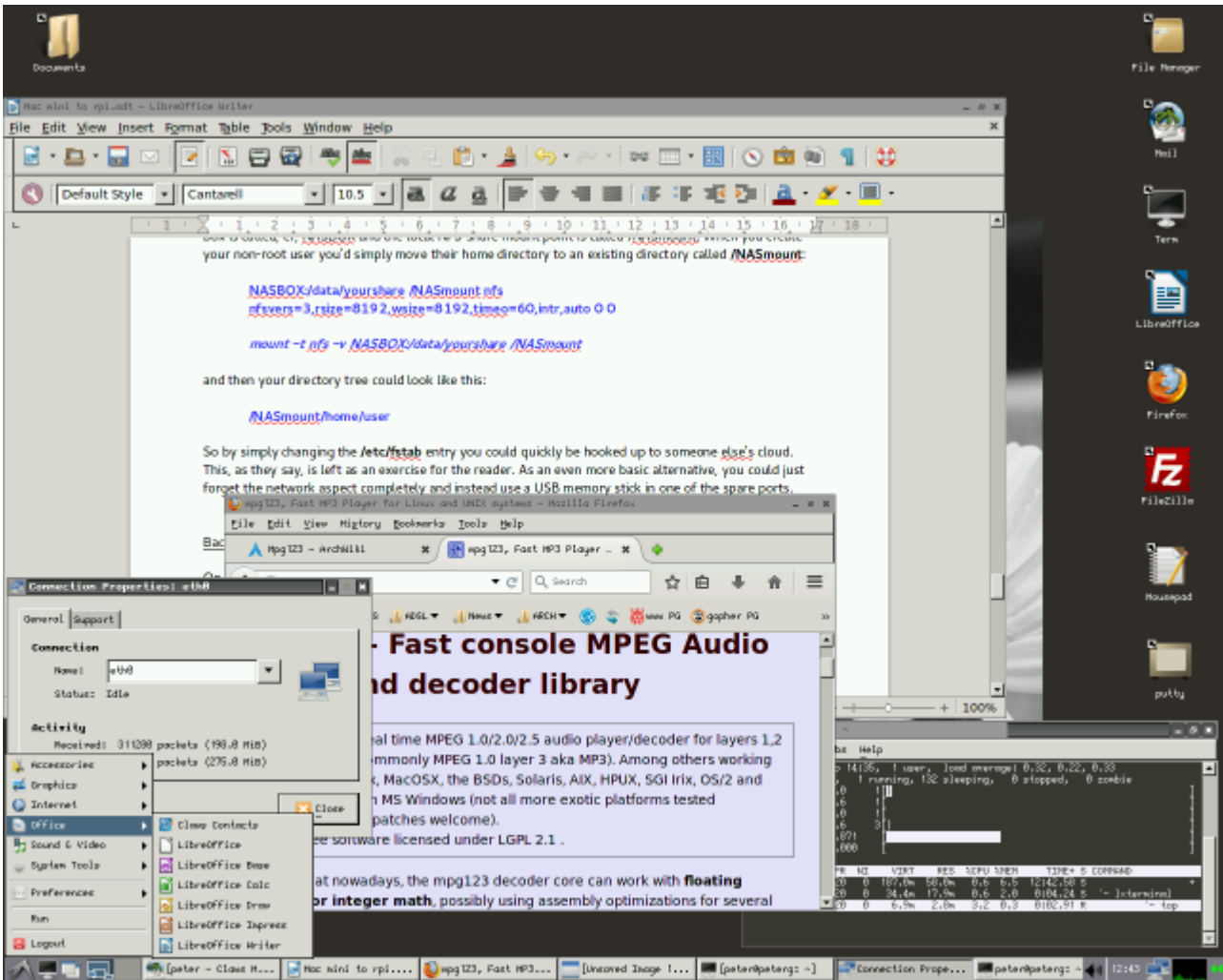
```
shell/i/.mp3
```

```
Open=/usr/bin/mpg123 -C %f
```

```
View=%view{ascii} /usr/lib/mc/ext.d/sound.sh view mp3
```

- Office: Don't be silly! Oh wait though; I installed the character-mode spreadsheet app called "sc" (Supercalc?), and there's always "vi" if you want to edit a text document, but don't expect to be able to edit any Microsoft files. If your need is truly great then you can install a supplementary application called "antiword" which will let you view a .doc file.
- Email: a bit of a problem as the Claws-mail mailbox format is not directly compatible with my character-mode app of choice, Mutt. There's a workaround, but I'm only going to do it if I get some spare time. For sending quick emails though, I installed `ssmtp`. Described as "*a send-only sendmail emulator for machines which normally pick their mail up from a centralized mail hub*". The setup is minimal and overhead is practically nil as it's normally only invoked when mail is being sent. So, you can do things like typing `echo "The donuts are on my desk" | mail -s"Important News" everybody@myoffice.com` from the command-line without firing up a GUI mail app.

For everything else it's just a question of flipping back to the GUI desktop. Speaking of which...



Raspberry Pi GUI desktop environment

Quite a busy screen, but the Raspberry Pi handles it well. Here, I'm using LibreOffice to write this article, there's a network status box, Firefox is on the mpg123 web page, and there's a terminal running TOP showing how much memory (isn't) being used. The drop-down menu at the left shows the Office Suite apps.

Other scenarios and thoughts

What's located where

With any hybrid system such as this, it's important to remember what is located where, so that in the event of any problems occurring, recovery will be easier. In my current configuration the microSDHC card contains only the Operating System, and *as far as possible* any system configuration files are also on there. Your own, user-land data will be on the NAS, in your home directory. Ideally, you should be to replace or update software on the microSDHC without having any adverse effects on your computing environment as a whole, but in IT it's never that straightforward.

If you take X11, the GUI desktop system, you'll find that although there is a default config file in `/etc/X11`, you will invariably have a customised version containing your own preferences (this is by design). Your *own* file on the NAS however will reference files on the microSDHC:

File contents	Location
<code>userresources=\$HOME/.Xresources</code>	NAS
<code>usermodmap=\$HOME/.Xmodmap</code>	NAS
<code>sysresources=/etc/X11/xinit/.Xresources</code>	OS
<code>sysmodmap=/etc/X11/xinit/.Xmodmap</code>	OS
...	
<code>if [-d /etc/X11/xinit/xinitrc.d] ; then</code>	OS
<code> for f in /etc/X11/xinit/xinitrc.d/?*.sh ; do</code>	OS
<code> [-x "\$f"] && . "\$f"</code>	
<code> done</code>	
<code> unset f</code>	
<code>fi</code>	

The overall effect is that if you either change one environment for another, you will invariably experience a loss of or change in functionality. Hopefully, the changes will be minor though, but you do need to be aware of the sometimes ambiguous links.

Please remember that the **root** user will *always* be on the microSDHC, and if your NAS box fails for any reason you'll still be able to boot your system and at least do some recovery work!

NAS alternative

While I'm in my "office" I have full access to my NAS box, which represents what in today's terminology would be a Personal Cloud. I much prefer this solution to a commercial cloud which is invariably managed by a company of unknown origin, location, security and motives. For those reasons I will always host my data where I can see it, and physically get to it as required. Having said that, if you care to go down the Pi Desktop route as I have done, then you may not be as paranoid as I am and want to hook it up to an external cloud share.

In that case, using an NFS mount as a basis for your home directory should mean that it's simply a matter of editing your **/etc/fstab** to point the NFS client at a different location. In my setup, the NAS box is called, er, NASBOX and the local NFS share mount point is called **/NASmount**. When you create your non-root user you'd simply move their home directory to an existing directory called **/NASmount**:

```
NASBOX:/data/yourshare /NASmount nfs  
nfsvers=3,rsize=8192,wsiz=8192,timeo=60,intr,auto 0 0
```

```
mount -t nfs -v NASBOX:/data/yourshare /NASmount
```

and then your directory tree could look like this:

```
/NASmount/home/user
```

So by simply changing the **/etc/fstab** entry you could quickly be hooked up to someone else's cloud. This, as they say, is left as an exercise for the reader. As an even more basic alternative, you could just forget the network aspect completely and instead use a USB memory stick in one of the spare ports.

Backups and security

On most computers, backups are done only occasionally, if at all, so I've ensured that I back up my Pi's on a regular basis.

The easiest way as far as I'm concerned is to just copy the whole SD card and put it somewhere safe. In fact, there's a script to do it, and even better, it can be done while the Pi is running, and without powering it down. **rpi-clone** uses rsync to fast-copy the contents of the current card onto an identical (capacity) one attached to a spare USB port, and will even partition and format the destination card as part of the process. An 8Gb card will take approximately 10 minutes end-to-end and once complete, you have the inner glow of satisfaction knowing that if the source card is corrupted you can simply swap it for your hot backup and carry on working. As regards security, a microSDHC is a tiny device, and if things get *really* desperate you could always swallow it. Only joking... well, sort of. Please note that on the Raspberry Pi model 2, inserting a USB card will result in the card being (hopefully) recognised and read. On the model B and possibly B+ it results in a reboot, so if you're doing something vital, please remember this.

As regards backing up your home directory, at some stage you should do it even though it's on a NAS. After all, by definition, the NAS is traditionally a place in which your stuff is held securely (as in resistance to hardware failure). The most reliable method I've found is to use **rsync** to an external device. In my case the external device is an encrypted USB memory stick which can be inserted into the Pi at any time. As always, there are a couple of points to note:

1. Is your target device big enough to hold your entire home folder? Surprisingly, my content and config files will comfortably fit into 2Gb as all my big media files are mounted as NAS shares.
2. I have to remember to unmount any large media repositories before I run the backup, otherwise rsync will happily follow the paths and I end up trying to copy 43,000 images and 9,000 music tracks onto a 2Gb memory stick.
3. Keep your memory stick safe and preferably encrypted – you're putting your whole digital life in one place and you wouldn't want to lose it.

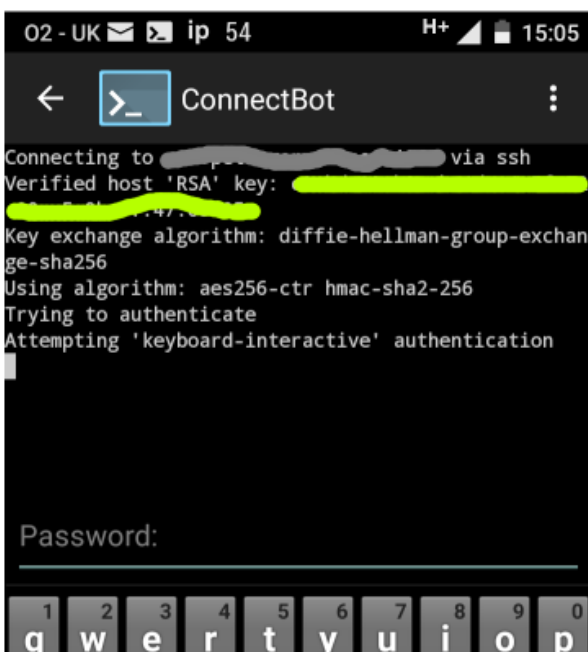
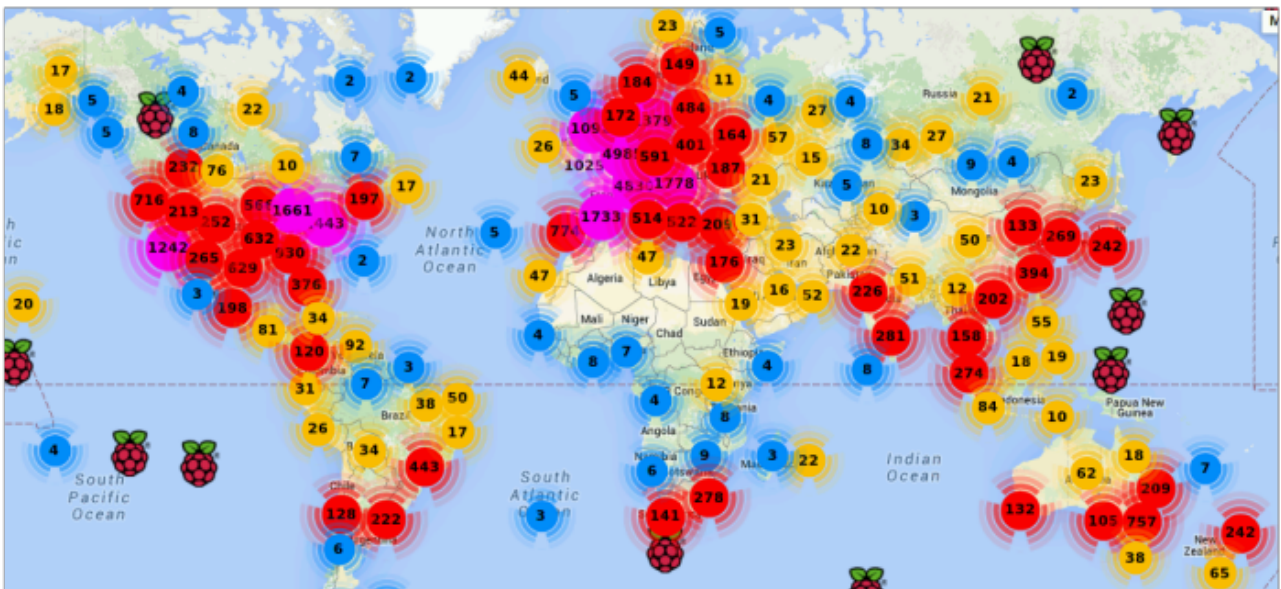
So here's the command to run (my) backup:

```
rsync -azvh --exclude /Downloads/ --exclude .cache/ /NASmount/home/user/  
/mnt/backupdevice
```

I've excluded a couple of folders as they invariably contain either lots of transient files or potentially very large downloaded files. You may need to add more folders as you see fit.

Versatility

There's no doubt in my mind that the Pi as a Single-Board-Computer is extremely versatile, not only in what it offers in software and hardware, but also in offering new, Green Computing options. It's physically small, so it can be transported easily, and the microSDHC is so small it could be kept in a wallet or purse. No more lugging a laptop through airport security: put the Pi in hold luggage and your "hard drive" is safe with you. Even if the Pi gets stolen, they're so common now that you can buy one almost anywhere.



Then there's the power options: it needs 5 volts at 2 amps, maximum, so you can run this from a battery: either a pocketable rechargeable unit, or one that's topped up by a solar panel. That's definitely something I have to try next!

Multi-user collaboration tool? Yeah, why not! Set up a Pi as an SSH server (hint, that comes as standard), and get your developers to login securely and create, edit and compile source code. This would be in a text-mode environment though: although SSH has the option to forward X11, the lack of power of the Pi would be a problem. Here's a picture of me connecting via my Android phone using "ConnectBot". Practically any platform /client combo could be used though.

So, where to next ?

Well, it's been quite a rough, but fun ride getting to this stage, and I can say that I've learnt substantially more about Linux compared with when I started out.

Right now I can say that my computing environment has now stabilised and I can do practically everything I used to do on the old Mac Mini, plus quite a bit more. The system remains lean and mean, and I don't intend to install anything major in the near future. My major annoyance is not being able to print, and I'm unlikely to be able to fix this in the near future, but I'll revisit this when I have time.

Likewise my video playback situation: to give you some idea, I tried to view the Pharrell Williams' "Happy" video the other day, and while the sound played fine, I was still watching the video in slo-mo 30 seconds after the song finished. But hey, you can't have everything...

Incidentally, the Raspberry Pi Model 2 has been hailed as a machine that can run Windows 10: I'm sure there's an element of truth in this but given the hardware limitations I don't see how this is possible, or even desirable. Maybe I'm wrong. We'll see (but I won't be trying it!).

In Part 2 I'll be looking to switch from the NAS to the Cloud or some other remotely-NFS-accessible storage. If you want to try it first then so much the better!

In Part 3 I'll be looking at alternative power sources. The Raspberry Pi takes only 5-10 watts at 5 volts DC, so there are options to potentially move it off grid.

End of Part 1