DIY NAS (Ubuntu 20.04 + mergerfs + snapraid)

This is a rough overview about my self build 6-Bay NAS (with external 4-Bay hard drive USB enclosure for backup drives).

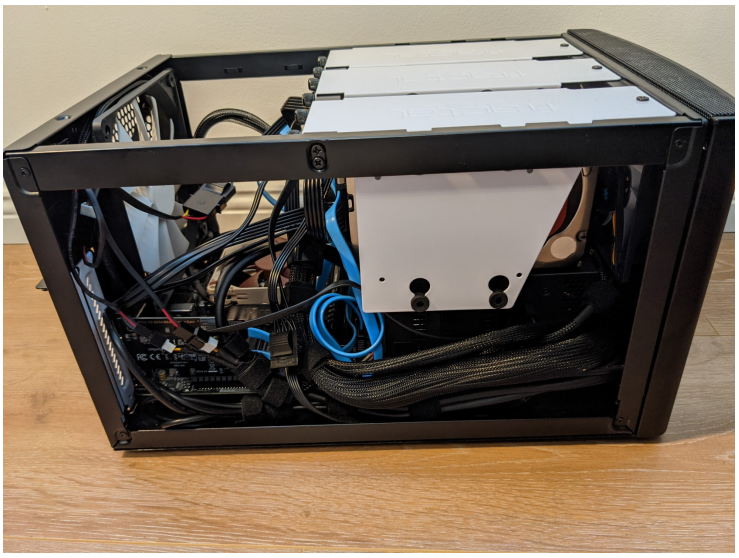
I build it from scratch with focus on low cost, low power consumption but still good performance

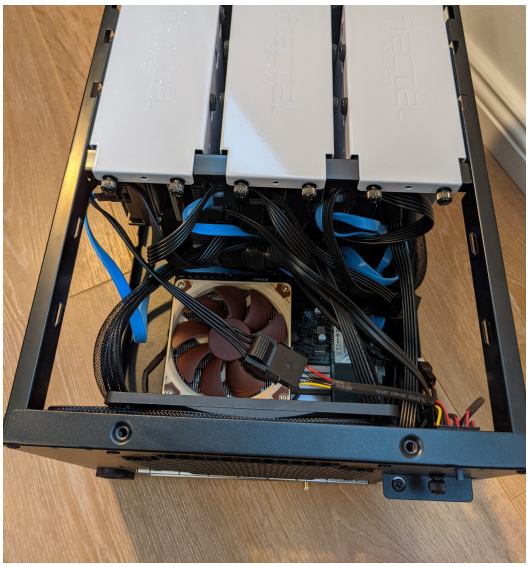
Total capacity currently: 65TB

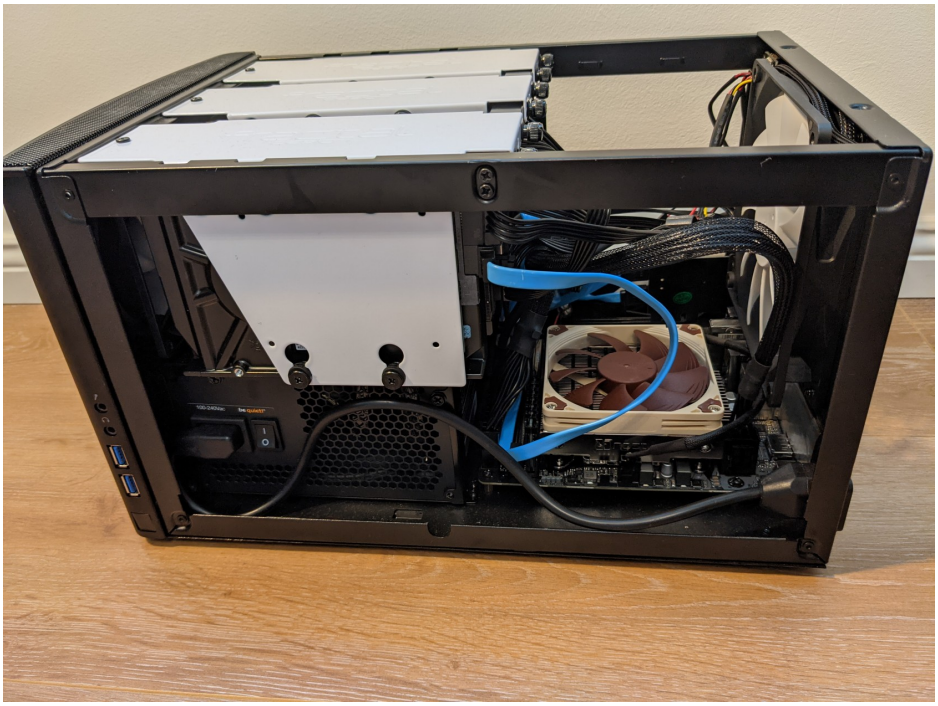
Available for normal usage: 26TB
Available for snapraid: 8TB
Available for backups: 29TB

Hardware

NAS







Components

Component	Model
CPU	Intel Core i3-10100 3.6 GHz Quad-Core
CPU Cooler	Noctua NH-L9i
Mainboard	ASRock H510M-ITX/ac Mini ITX LGA1200
Memory	Crucial RAM CT8G4DFRA266 8GB DDR4 2666 MHz CL19

Case	Fractal Design Node 304 Mini ITX Tower
Power Supply	be quiet! Pure Power 11 CM 400W (80+ Gold Certified Semi-modular ATX)
Extension Card	BEYIMEI PCI Express to 2 Port SATA III 6Gbps
Drives	• Transcend 120GB SSD SATA III 6Gb/s MTS820S • WD Red 6TB 3.5" 5400 RPM (WD60EFRX) • WD Red 6TB 3.5" 5400 RPM (WD60EFRX) • WD Red 8TB 3.5" 5400 RPM (WD80EFAX) • WD Red 8TB 3.5" 5400 RPM (WD80EFAX) • Seagate IronWolf 8TB 3.5" 7200 RPM (ST8000VN004) • Seagate IronWolf 8TB 3.5" 7200 RPM (ST8000VN004)

Notes

- At first I bought the Intel Core i3-10100F CPU (F => no integrated GPU), because it did only cost half of what the version with GPU costs and to minimize the power usage of the CPU (why should I need a GPU on a NAS anyway). I wasn't aware that I couldn't even get into the mainboards BIOS/UEFI without GPU, so that did not work.
- The mainboard only has 4x SATA III interfaces but the case supports 6x 3.5" drives, this is why I needed the PCI-E extension card (I found no cheap mini ITX mainboard with 6x SATA interfaces)
- I opted for the (more expensive) gold certified PSU so that I can utilize as much of the drawn power as possible to save energy and cost
- I currently use only one memory module. I plan to upgrade this to two memory modules in the future, so I can utilize dual channel RAM for a bit of performance gain and be more safer in case one module dies.

Backup

Components

Component	**Model**
Hard Drive Enclosure	Xystec 4-Bay Hard Drive Enclosure, USB 3.0 or eSATA

Drives	• Seagate Barracuda 5TB 3.5" (ST500DM003) • Toshiba 6TB 3.5" (MD04ACA600) • Seagate Barracuda 8TB 3.5" (ST800DM004) • WD 10TB 3.5" (WD1EMAZ)

Notes

- I used the USB hard drive enclosure in combination with a Raspberry Pie as my old NAS. Now that I have build my new NAS I did no longer have a usage for it.
- All backup drives are from shucked external drives which I used as backup drives in my old setup. I decided it was more practical to combine the external drives into one enclosure instead of having them separate.

Software

This is a quick overview about what software I am using and why.

Operating system:

Ubuntu 20.04

I used openmediavault (omv) in the past, but the process of upgrading from one version to another seemed tedious and the fact that the version of Debian omv was using is missing required drivers for the network interfaces of my mainboard made me look for an alternative OS. I decided to go with Ubuntu because I have another server running it and have had mostly good experiences with it.

I wanted to use Ubuntu Server for a lightweight installation without needless bloat, but that was missing drivers for my mainboard too and I couldn't even boot via live USB, so I had to go with the Ubuntu Desktop version. After successfully installing the OS and setting up a SSH server, I deleted the Ubunutu desktop package and changed the system runlevel target to multi-user so that I can negate the negatives of having to install the Desktop version.

Drive management:

mergerfs

I used mergerfs on my old NAS and loved it, so I wanted to go with it again. A few reasons why:

- easy to setup and maintain
- able to add/remove a disk of any size and format whenever I want
- failure of one disk does not break everything (only the files on the failed drive are lost)
- all individual drives are combined together into one pool (which than can be separate again into individual directories/shares)

More about how mergerfs works here.

snapraid

Backup program for drive arrays, stores parity information of data and it recovers from drive failures.

I didn't want to go with a conventional RAID to be more flexible in my infrastructure, but still be able to recover from a drive failure. Snapraid seemed like the ideal solution for that. Like mergerfs it is easy to setup and very flexible, I could add a new drive or replace an old one whenever I want in whatever size (as long as the new drive is not bigger than the parity drive) without fearing data lose.

Unfortunately, there is no snapraid release for Ubuntu 20.04 available atm, but it is super easy to compile snapraid yourself. Download the latest release and follow the install instructions.

Monitoring

- hddtemp: Monitoring drives temperatures
- netdata: Monitoring system performance/load/temperature/etc. via web browser
- smartmontool: Monitoring drives health with S.M.A.R.T.

Networking

- ssh: Captain obviouse
- samba: Sharing directories with Windows and Linux systems (see no reason to use nfs currently)

Replace data disk

What happend?

I have a nightly cronjob configured to execute a script which

- runs a SnapRAID sync and scrub when some requirements are met (e.g. number of deleted files do not exceed a threshold)
- and sends me an email to notify me about what happend during the script execution

This is how I got notified that the latest SnapRAID sync could not be executed: Some files on one of the data disks could not be read because of an "Unexpected input/output read error".
This did not really surprise me because SMART was reporting the affected disk to have a 100% estimated probability to fail soon for a while now. Additonally, this already happend on this disk before, but I could usually easily fix it by replacing the broken file. Not this time. Now much more files were affected (including files I newly added trying to fix the broken ones as before).

There was no way around it now: I had to replace the disk.

....