

[Get started](#)[Open in app](#)

## Lee Bardon

[Follow](#)

30 Followers

[About](#)

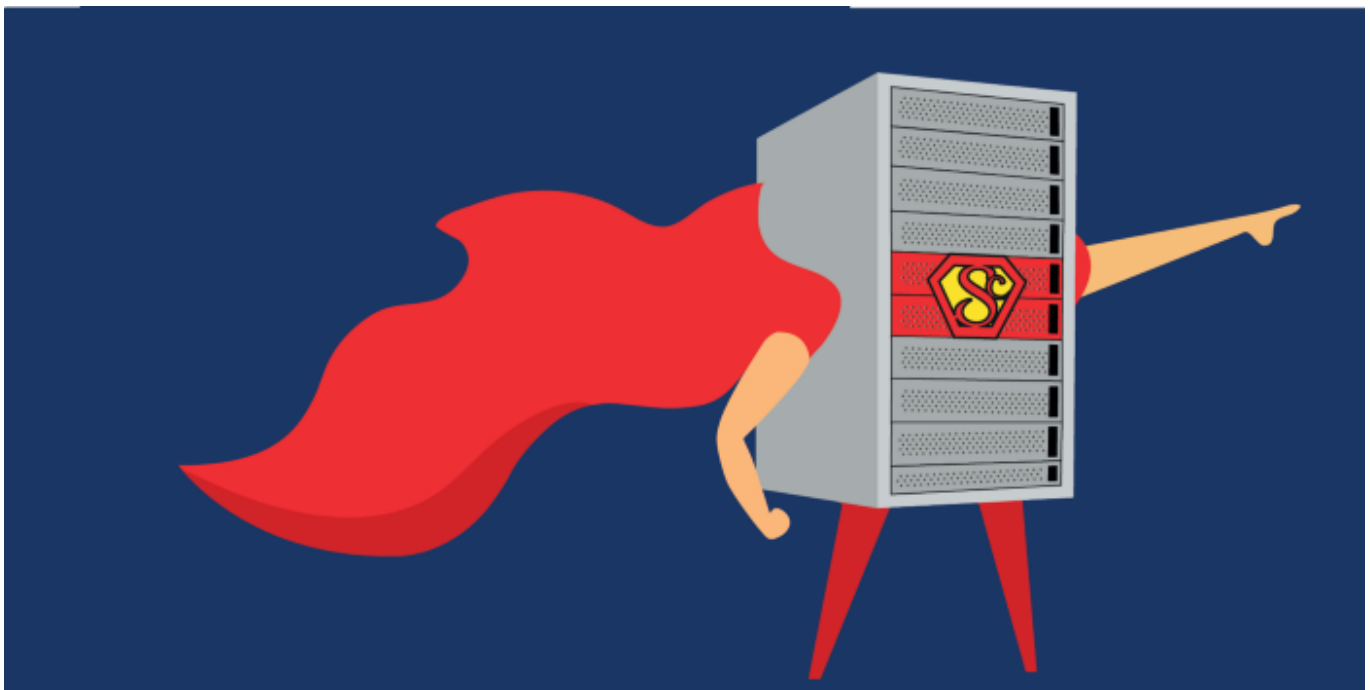
You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

# Build A DIY Supercomputer: Part 1!

Hardware Build (Odroid N2+ cluster)



Lee Bardon · Aug 8, 2020 · 6 min read ★



k, so perhaps ‘Supercomputer’ is a bit grand, but by the end of this three-part series,

[Get started](#)[Open in app](#)

only is it inexpensive to run (thanks to the ARM-based architecture), but the whole project will cost you less than £400, plus a day or two of your time.

## 1. WHY?

The chances are that if you've read this far, you already understand the potential use cases and benefits. But just in case, I'll first outline my own reasons for undertaking this project.

Once upon a time, I was a PhD student in computational geophysics. Shortly after (upon a time), I left to pursue an entirely unrelated, nontechnical business opportunity... before ultimately deciding that I missed the challenge and thrill of research. As a result, I've dedicated the greater part of the last year to doing everything I can to improve my programming and data science skills. In doing so, I have reinitiated relationships with contacts in the academic world, and have recently entered into some informal research collaborations.

### 1a. Computational Need

My role in these collaborations involves processing large geospatial datasets in a variety of ways, before applying some machine learning analyses. Unfortunately, since I'm not currently an official member of any research institution, my only computational tool is my laptop. While it's certainly possible to perform most of these analyses on my laptop, it can take many hours at a time in some cases, during which the CPU load is consistently high.

Of course, building your own home cluster is not the only solution for data science enthusiasts with relatively demanding computational requirements: there are numerous cloud computing options that could accomplish a similar feat, and perhaps even less expensively (depending on your requirements). But computational need is not my only reason...

### 1b. Opportunity to Learn

During my last stint as a computationally-focused research student, we had two options when it came to running simulations and/or processing data.

[Get started](#)[Open in app](#)

capabilities than the average PC desktop; and a much greater price tag to go along with it. Given my minimal budget, I knew this wouldn't be an option. In addition, while powerful and usually available, the workstations were not able to utilise parallel computing methods.

The second option involved being allocated time on the lightening-fast, state-of-the-art university or national supercomputer clusters, that could handle anything we threw at them. However, they are also shared by many, many researchers, and resources are allocated very sparingly. There was often a queue for submitted jobs, and, especially for new and inexperienced PhD students, there was a good chance that your scripts or simulations would fail for some reason, bumping you right back down the queue.

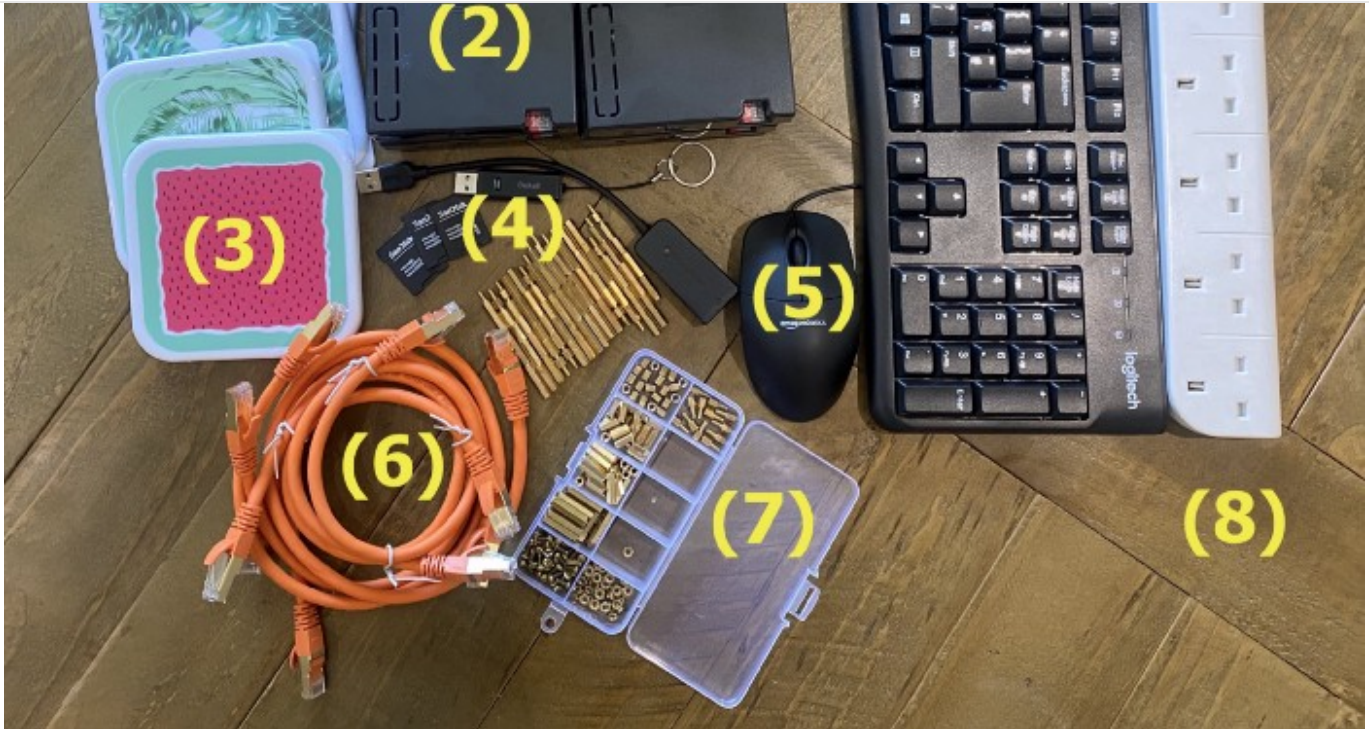
It often struck me that new HPC users would be better off learning the do's and don'ts of parallel and distributed computing using smaller scale clusters, rather than wasting precious national supercomputer resources. Indeed, some academics champion the use of Raspberry Pi clusters for this exact reason, and online tutorials already exist within the Raspberry Pi community.

So, the decision was easy: *I was going to build my own home cluster!*

## 2. EQUIPMENT

We'll go through the details of network design later in this post, with OS set-up in subsequent posts. For now, we'll stick to gathering and assembling the required hardware.



[Get started](#)[Open in app](#)

**Fig 1.** My Set Up (your specific requirements may vary)

1. *Ethernet Switch with power cable (8-port);*
2. *Odroid N2 x 4 with power cables (I'll detail why I chose these SBC's below);*
3. *Bases For Each SBC ~15cm x 15cm (I repurposed tupperwear box lids);*
4. *MicroSD cards x 4 (64GB), MicroSD card reader x 1, Ethernet dongle x 1;*
5. *USB keyboard and mouse (required for initial set-up);*
6. *Ethernet cables x 5;*
7. *Male/female spacer and standoff kit (M3);*
8. *Six-way surge-protected power adapter.*

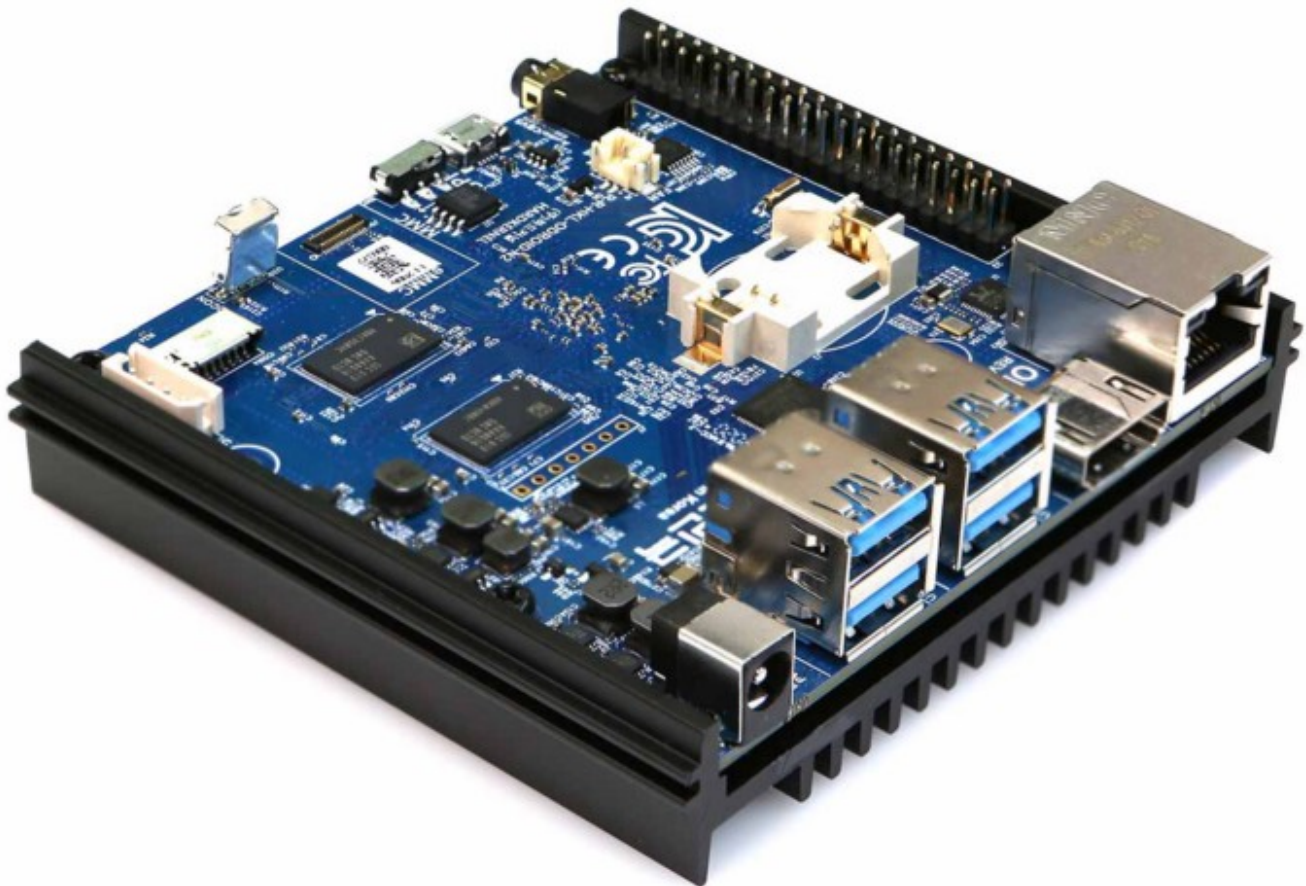
I mentioned above that Raspberry Pi's have been used for several years to build home clusters. The large and active Raspberry Pi community has many resources that will



[Get started](#)[Open in app](#)

against the home Pi clusters (this does not extend to legit Pi supercomputers), is that they're a little too underpowered to do any 'serious' computation, with the most affordable models having e.g. single 700MHz CPU cores, and 512MB RAM.

With the release of the Raspberry Pi 4, we see a considerable increase in capabilities. Nonetheless, I opted for the recently released Odroid N2+ from South Korean manufacturer HardKernel. Each board has 6 cores at 2.4 GHz and 4GB RAM, a highly-efficient passive aluminium heatsink, and 800MHz GPU. In other words; it's quite a little beast!



**Fig 2.** The new Odroid N2+ from HardKernel

### 3. THE BUILD

[Get started](#)[Open in app](#)



[Get started](#)[Open in app](#)

The first stage of the build involved attaching the SBC's to each of the bases. On the bottom of the N2, you'll find that the heatsink has four small screw holes. The aim is to attach the each SBC to a base via these screws, and then use the M3 brass spacers to attach each unit together, one on top of the other, as per the above image.

For this, you'll need a ruler, a pen, a drill with a 2.5–3mm bit, four of the screws that come with the M3 spacer and standoff kit, and four 12mm brass spacers.



[Get started](#)[Open in app](#)

**Fig 4.** Red crosses align with the holes on the heatsink, blue crosses are where we'll attach our M3 spacers.

First, screw the 12mm spacers into the bottom of the heatsink. Then, drill the corresponding holes into the base, as well as the holes for the longer 60mm M3 spacers that span between each unit. Use the screws to secure each of the N2's, and assemble your tower according to the image in Fig 3.

## 4. NETWORK DESIGN

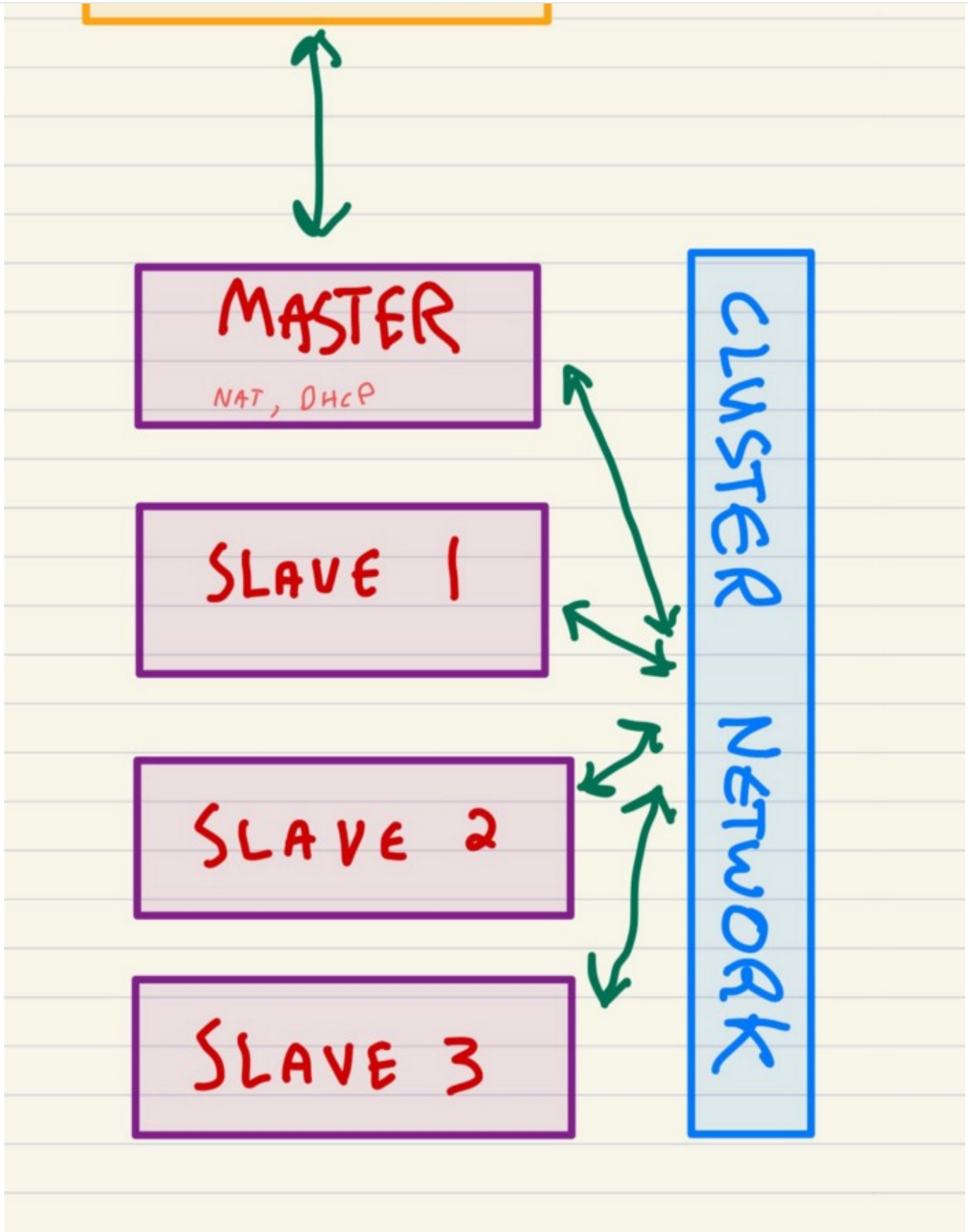
Of key consideration in building our cluster, is understanding how the nodes will interact with each other. In this case, we have 4 compute nodes, one of which will be designated as the 'master' node with the others operating as 'slave nodes'.

The master node will receive incoming directions, for example, an analysis job, and then distribute that job across the other nodes in the cluster, before compiling and relying the results. In order for this to occur, each node will be connected to each other via the ethernet switch. However, in order to maximise throughput, we will set-up the cluster to operate as a single unit within your home network. Of course, this could have been achieved by purchasing a router instead of a switch, albeit at significant extra costs.

Instead, we can instruct the master node to operate as a router through the use of a USB ethernet dongle. This will require that we configure it to be both a DHCP server for the cluster network, and provide the NAT service between the external network and the cluster network.





[Get started](#)[Open in app](#)

**Fig 5.** Network design for Odroid N2+ cluster

[Get started](#)[Open in app](#)

Get started

Open in app



Data Science

Parallel Computing

Hpc

Linux

Raspberry Pi

About Help Legal

Get the Medium app

