



Pine64: Synapse on a SBC Cluster

Matthew Petry

@fire219:matrix.org

(or @fire219:pine64.org)

Agenda

- The Two Sides of Pine64
- The RockPro64 Cluster
- Synapse on the Cluster?



The Two Sides of Pine64

(how to make FOSS-friendly devices)



Pine64

- So what is it?
- According to Wikipedia:

Pine64

From Wikipedia, the free encyclopedia

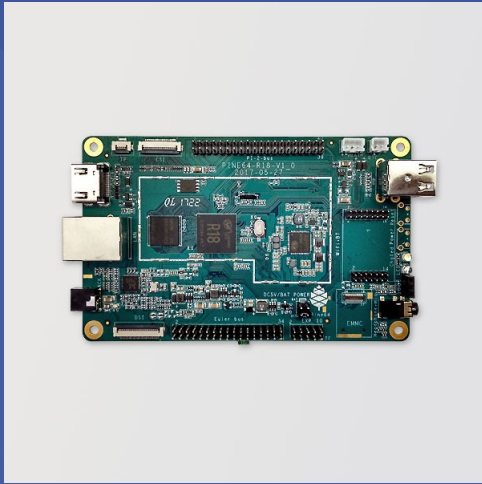
Pine64 is an organization which designs, manufactures and sells [single-board computers](#), [notebook computers](#) and smartphones.

- I guess that's not really wrong, but there's a bit more to it...



Pine64 Hardware

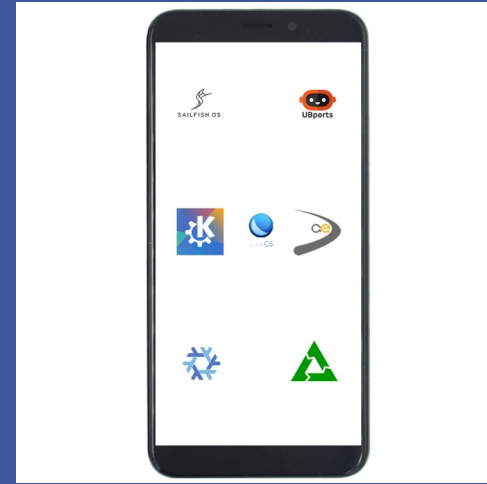
- Do one thing and do it well: Make FOSS-friendly ARM hardware at lowest possible cost



Pine A64



Pinebook Pro



PinePhone



Pine64, a bit more in depth

- Rather unusual: zero employees!
- Everyone within the organization is either a contractor, or volunteer.

Two distinct entities:

- Business: Pine Store Ltd. (*Manufacturing, Logistics, Sales, Official Support*)
- Community: Pine64 (*Main website, forum, wiki, chat rooms, social media*)



It takes two to tango

- Hardware is nice and all, but it's not much good without the software to run on it!
- Most people watching this probably agree: Hardware businesses and FOSS usually don't mix well
 - Official firmware often full of blobs, undecipherable without access to NDA-gated documentation



Separation of Responsibilities

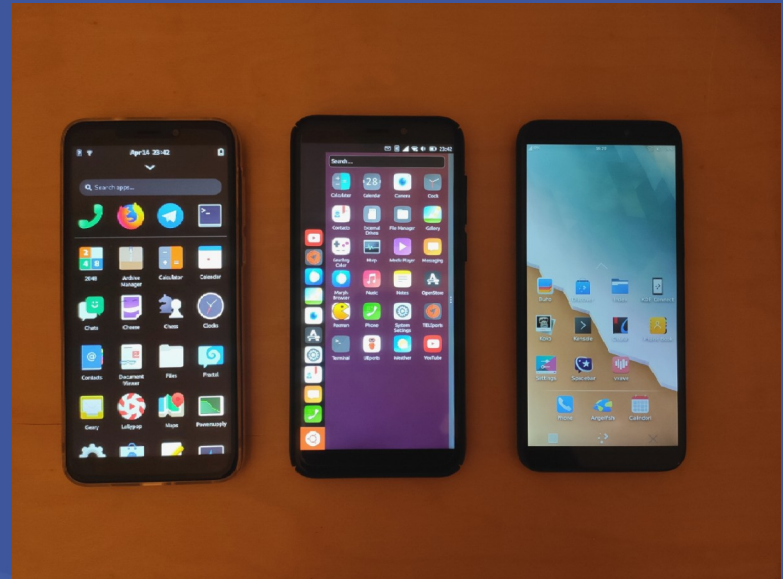
- The solution is rather simple: **have the community handle software development!**
- In short:
 - Business → Pine Store Ltd. → Hardware
 - Community → Pine64 → Software
- The two sides run independently, but work together to make a complete experience.

Here's a couple more benefits...



Community Software is Diverse Software

- As an example, the PinePhone has firmware images available spanning several Linux distributions (*PostmarketOS, UBports, Mobian, etc.*)
- Shown Right: PinePhones running three different desktop environments – *Phosh*, *Lomiri (Unity8)*, and *Plasma Mobile*



Community Software is Up-To-Date Software

- Many embedded platforms end up relying on (semi-)proprietary software
 - BSPs (Board Support Packages) built around a specific Linux kernel version, with blobbed modules
 - Very quickly end up obsolete: missing out on performance, feature, and security upgrades
- Businesses update software when it'll make them money, enthusiast communities update it just because they want the latest and greatest.



It's not a one way street...

- At this point, you might think this is all just a ploy to get free labor
 - *Software R&D departments are expensive, after all*
- But here's the other side of the deal: **The community (especially developers) get to shape the future of Pine64 devices.**
- This can be as simple as polling preference between spring-loaded or friction-fit microSD card slots, or influencing an entire hardware development cycle...



The PinePhone development cycle

- Phase 1: “Project Anakin”
 - Basically just a bunch of off-the-shelf Pine64 products put together with an included USB cell modem
 - Given to developers late 2018



The PinePhone development cycle

- Phase 2: Project “Don’t Be Evil”
 - Hardware refined using feedback from Phase 1 developers, now into something actually smartphone shaped
 - Given to developers in early 2019



The PinePhone development cycle

- Phase 3: Production-ready PinePhone
 - Mainboard revision 1.0: Dev
 - Late 2019
 - Mainboard revision 1.1: “Braveheart”
 - January 2020
 - First version ready for consumers
 - Mainboard revision 1.2(a): Community Editions
 - June 2020
 - Fix all known remaining bugs
 - Ready for daily use



Practice what you preach...

This presentation is being given from a Pinebook Pro...

...Running community-developed software (Manjaro ARM)...

...with a fully-FOSS (untainted) build of Linux kernel 5.8

```
[matthew@MPetry-PinebookPro ~]$ neofetch
matthew@MPetry-PinebookPro
-----
OS: Manjaro ARM Linux aarch64
Host: Pine64 Pinebook Pro
Kernel: 5.8.0-2-MANJARO-ARM
Uptime: 6 hours, 51 mins
Packages: 831 (pacman)
Shell: bash 5.0.17
Resolution: 1920x1080
DE: Plasma
WM: KWin
WM Theme: Layan
Theme: Layan [Plasma], Breeze [GTK3]
Icons: Tela-dark [Plasma], Tela-dark [GTK2/3]
Terminal: konsole
CPU: (6) @ 1.416GHz
Memory: 1209MiB / 3796MiB
```

```
[matthew@MPetry-PinebookPro ~]$ dmesg | grep taint
[matthew@MPetry-PinebookPro ~]$
```

:)



The RockPro64 Cluster

(how to dog-food your own products)



The RockPro64 Cluster

(how to dog-food your own products)


Eating your own dog food


From Wikipedia, the free encyclopedia


Eating your own dog food or **dogfooding** is the practice of an organization using its own product.^[1]




People have been asking for a long while...


 **Human Sandrapede** @Sinister... Jun 8
Replying to @thepine64
Can we expect an article on the move at some point? Would be fascinating to read a play-by-play of how the migration occurred and how you deal with problems (because there are always problems).
1 1 3 ...

 **Manu** @manu@mastodon.sdf.org
@PINE64 hey are you going to document your cluster build? It would be nice to be able to build a similar al-in-one-2u cluster.

 **exprez135** @exprez135@fosstodon.org
@PINE64 do you have a blog post about the technical aspects of the cluster?

 **wasabiBurger** @wasbpr Apr 21
Replying to @thepine64
I am a noob when it comes to electronics, but I am very much interested in learning how to power an SoC cluster like this. Could you share some insights?
1 1 1 ...

 **selea** @selea@social.linux.pizza
@PINE64
I would love to read more about the cluster configuration!

 **DJ Sundog - from the toot-lab** @djsundog@toot-lab.reclaim.technol...
@PINE64 looking forward to seeing more about that cluster configuration ;)

@PINE64 is there some kind of tutorial how you build the cluster for your website stuff?

 **temptage** @temptage1 Jun 9
Replying to @thepine64 @SinisterMrCrea...
I agree this would be cool to read about the move and information about the cluster!!
1 1 1 ...



People have been asking for a long while...

We've been promising a blog write-up on our cluster set-up for a long while.

Still hasn't happened: perfectionism getting in the way of "good enough".

This isn't a substitute for that blog post.

But it is the first time the configuration has been discussed in public.



The Cluster: Hardware

- 24x RockPro64 (4GB) SBCs
- 2x 1TB Samsung 970 EVO SSDs
- Custom designed 2U rackmount chassis
- Built in Gigabit Ethernet switching
- Backplane for Remote Reboot capability
 - *(not implemented yet... this will be an issue later)*

- *Colocated at BBXNET*



The Cluster: Software *(in people's dreams)*

Whenever the cluster has been mentioned in our social media, people's expectations go wild.

They expect we've made some incredible distributed-load system.

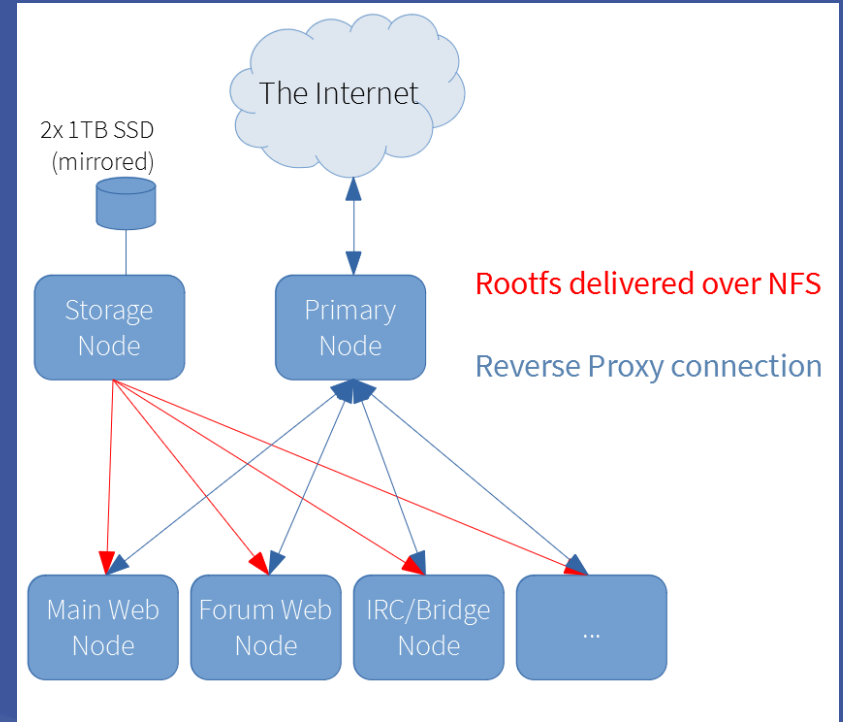
Maybe using cool stuff like Kubernetes?

Sorry to let you down. We ~~got lazy~~ ran out of time.



The Cluster: Software *(in reality)*

- All nodes run/compute independently
- Custom Debian 10 images with kernel 5.7
- Storage node handles all root filesystems, delivers to other nodes via NFS
 - *(will change to iSCSI soon... for good reason)*
- All external (internet) access to services is NGINX reverse proxied through primary node



The Cluster: Software *(in reality)*

- This relatively simple setup was much faster to bring up than a “proper” distributed cluster.
- Want to bring up another service? Just bring up another node, set up the software, and route reverse proxy accordingly.
- The downside? Well...



Synapse on the Cluster

(The trials and tribulations of 4GB RAM)



Long Time Coming...

For quite a while now, we've had our various chatrooms bridged on several platforms

- *Matrix, Discord, IRC, Telegram*

For even longer than that, people have been asking when we'll have an official Pine64 Matrix homeserver.

Until recently, we just didn't have the resources to try.

But now we have *twenty-four* RockPro64 boards sitting in a datacenter. After initial deployment, only six were in use.

So, why not?



Doing it by the book.

Soon after our initial cluster deployment, I get to work setting it up by the ~~book~~ INSTALL.md

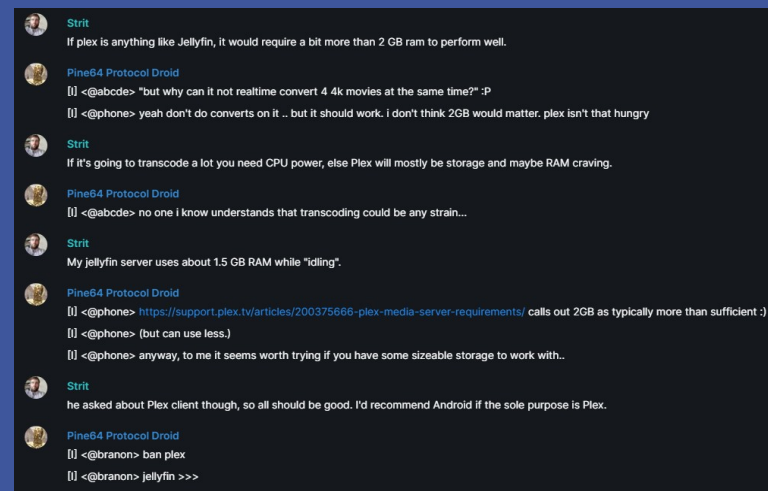
- Fight with dependencies
- Install Synapse
- Make a config
- Start Synapse
- Make another config after remembering .well_known exists
- Restart Synapse
- Realize that sqlite is awful
- Install postgresSQL
- Restart Synapse

```
mkdir -p ~/synapse
virtualenv -p python3 ~/synapse/env
source ~/synapse/env/bin/activate
pip install --upgrade pip
pip install --upgrade setuptools
pip install matrix-synapse
apt install zlib-dev
apt search zlib
apt install zlib1g-dev
pip install matrix-synapse
ls
cd ~/synapse
python -m synapse.app.homeserver --server-name matrix.pine64.org
omeserver.yaml --generate-config --report-stats=no
ls
nano homeserver.yaml
```



It Works!

- After that sequence of events, the pine64.org homeserver is running!
- Happily ever after, right?
- *Not so fast...*



Strit
If plex is anything like Jellyfin, it would require a bit more than 2 GB ram to perform well.

Pine64 Protocol Droid
[] <@abcde> "but why can it not realtime convert 4 4k movies at the same time?" :P
[] <@phone> yeah don't do converts on it .. but it should work. i don't think 2GB would matter. plex isn't that hungry

Strit
If it's going to transcode a lot you need CPU power, else Plex will mostly be storage and maybe RAM craving.

Pine64 Protocol Droid
[] <@abcde> no one i know understands that transcoding could be any strain...

Strit
My jellyfin server uses about 1.5 GB RAM while "idling".

Pine64 Protocol Droid
[] <@phone> <https://support.plex.tv/articles/200375666-plex-media-server-requirements/> calls out 2GB as typically more than sufficient :)
[] <@phone> (but can use less.)
[] <@phone> anyway, to me it seems worth trying if you have some sizeable storage to work with..

Strit
he asked about Plex client though, so all should be good. I'd recommend Android if the sole purpose is Plex.

Pine64 Protocol Droid
[] <@branon> ban plex
[] <@branon> jellyfin >>>



The Pine64 Community is quite large. *</humblebrag>*

If this were a small personal homeserver, I'm sure the story could end here.

But it's not.

Pine64 Matrix room sizes:

- Pinebook: **~400 users**
- Pine64 (General): **~430 users**
- PinePhone: **~1340 users**

Many different homeservers among those.



It Works... for about 2 hours.

- All seemed well. Until all of a sudden...
- oom-killer'd.
- Long story short, Synapse apparently **really** likes to eat RAM when told to deal with big rooms.
- Why not just implement swap space?
 - On an NFS share? Nope.
 - Give node local storage? Maybe, but rather not.

```
[14198.603112] python3 invoked oom-killer: gfp_mask=0x100cca(GFP_HIGHUSER_MOVABLE), order=0, oom_score_adj=0
[14198.603958] CPU: 5 PID: 1271 Comm: python3 Not tainted 5.7.0 #1
[14198.604475] Hardware name: Pine64 RockPro64 v2.1 (DT)
[14198.604918] Call trace:
[14198.605144] dump_backtrace+0x0/0x1c0
[14198.605468] show_stack+0x14/0x20
[14198.605762] dump_stack+0xb4/0x1fc
[14198.606055] dump_header+0x44/0x1e8
[14198.606361] oom_kill_process+0x204/0x208
[14198.606712] out_of_memory+0xe8/0x328
[14198.607037] __alloc_pages_slowpath.constprop.0+0x954/0x9f8
[14198.607524] __alloc_pages_nodemask+0x1fc/0x260
[14198.607922] pagecache_get_page+0x188/0x378
[14198.608288] filemap_fault+0x57c/0x820
[14198.608615] __do_fault+0x4c/0x190
[14198.608915] __do_fault+0x11c/0x620
[14198.609211] __handle_mm_fault+0x14c/0x7d0
[14198.609571] handle_mm_fault+0xb0/0x170
[14198.609809] do_page_fault+0x130/0x488
[14198.610239] do_translation_fault+0x5c/0x78
[14198.610606] do_mem_abort+0x3c/0x98
[14198.610912] el0_ia+0x44/0xb0
[14198.611174] el0_sync_handler+0x104/0x178
[14198.611527] el0_sync+0x140/0x180
[14198.611837] Mem-Info:
[14198.612044] active_anon:939258 inactive_anon:4869 isolated_anon:0
active_file:65 inactive_file:31 isolated_file:0
unevictable:0 dirty:0 writeback:0 unstable:0
slab_reclaimable:4770 slab_unreclaimable:10140
mapped:104 shmem:4927 pagetables:2234 bounce:0
free:4720 free_pcp:359 free_cma:10
```

```
[14198.645443] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-cl.scope,task=python3,pid=1271,uid=0
[14198.646877] Out of memory: Killed process 1271 (python3) total-vms:4169936kB, anon-rss:3333756kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:7036kB oom_score_adj:0
[14198.969366] oom_reaper: reaped process 1271 (python3), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
```



Take Trouble, and Make it Double

Well, Synapse does support splitting into workers. Lets try that.

Chain of events:

- Read workers.md
- Spin up another RP64 node
- Write worker scripts for both nodes
- Get lazy for a month
- Get a Matrix DM about an event called OTWSU
- Scramble to try to get Synapse up and running in time
- Crash node 2 hard enough to require physical intervention
- Finally get Synapse running again! ... mostly.

```
2020-08-11 23:18:25,258 - synapse.storage.data_stores - 77 - INFO - None - Starting 'state' data store
2020-08-11 23:18:25,265 - synapse.storage.data_stores - 90 - INFO - None - Database 'master' prepared
2020-08-11 23:18:25,267 - synapse.server - 284 - INFO - None - Finished setting up.
2020-08-11 23:18:25,337 - synapse.push.pusher - 42 - INFO - None - email enable notifs: False
started synapse.app.federation_reader('workers/fedread.yaml')
```

```
ase 'master' prepared
2020-08-11 19:02:43,282 - synapse.server - 284 - INFO - None - Finished setting up.
2020-08-11 19:02:43,346 - synapse.push.pusher - 42 - INFO - None - email enable notifs: False
started synapse.app.synchrotron('workers/synchrotron.yaml')
```



Right when I felt good about myself...

Synapse is running, but federation seems a bit broken.

It's talking properly to most other homeservers as far as I can tell.

But not matrix.org for some reason?

This results in many missing messages.

(I guess that's why we need more homeservers. :))

When I start trying to troubleshoot that...



Right when I felt good about myself...

```
[791829.784962] [ 22751] 106 22751 53513 35634 421888 0 0 postgres
[791829.785714] [ 22755] 106 22755 53419 31866 421888 0 0 postgres
[791829.786464] [ 22757] 106 22757 53521 33549 421888 0 0 postgres
[791829.787213] [ 22759] 106 22759 53505 32910 421888 0 0 postgres
[791829.787963] [ 22773] 106 22773 53909 36434 425984 0 0 postgres
[791829.788713] [ 22776] 106 22776 53781 36101 421888 0 0 postgres
[791829.789464] [ 22778] 106 22778 53895 36320 425984 0 0 postgres
[791829.790214] [ 22780] 106 22780 53787 36104 421888 0 0 postgres
[791829.790963] [ 22782] 106 22782 53763 36151 421888 0 0 postgres
[791829.791728] [ 22785] 0 22785 1083 242 53248 0 0 htop
[791829.792454] [ 23184] 106 23184 53438 16688 421888 0 0 postgres
[791829.793206] [ 23257] 106 23257 53015 1380 167936 0 0 postgres
[791829.793957] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global
_oom,task_memcg=/user.slice/user-0.slice/session-c8.scope,task=python3,pid=22656,uid=0
[791829.795469] Out of memory: Killed process 22656 (python3) total-vm:2281768kB, anon-rss:1600780
kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:3332kB oom_score_adj:0
[791829.980302] oom_reaper: reaped process 22656 (python3), now anon-rss:0kB, file-rss:0kB, shmem-
rss:0kB
```

~_(\ツ)_/~



That's All, Doc!

That's the status as of the night before OTWSU 5.

I wanted to have this all working in time, but that's how life goes.

If anyone has any ideas as to what's going on with the federation issue, I'm listening.

Totally not a cry for help. :)

If you still want to give it a try, Synapse is running (albeit with the questionable federation.)

matrix.pine64.org

Registration is open.



Questions?

(Sorry for the likely excessively long presentation.)

