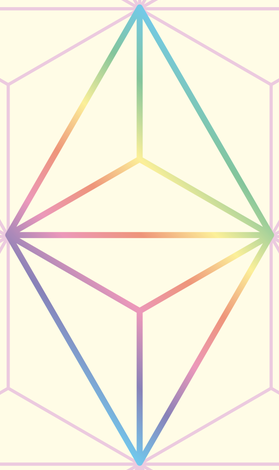
Ethereum 2.0 on a Raspberry Pi

Security, Scalability, Sustainability

Mamy Ratsimbazafy

Status



We focus on

Low-powered devices

And core infrastructure

- Geth on mobile
- Whisper
- Vyper
- **Nimbus**
  - Ethereum 1
  - **Ethereum 2**

~~The program~~

The Program

- Some context: Why running on a Pi is significant.
- Pi* experiments
- Outcomes, future and unknowns

*No raspberry was hurt in the making of this talk

Context: Byzantine Fault Tolerance

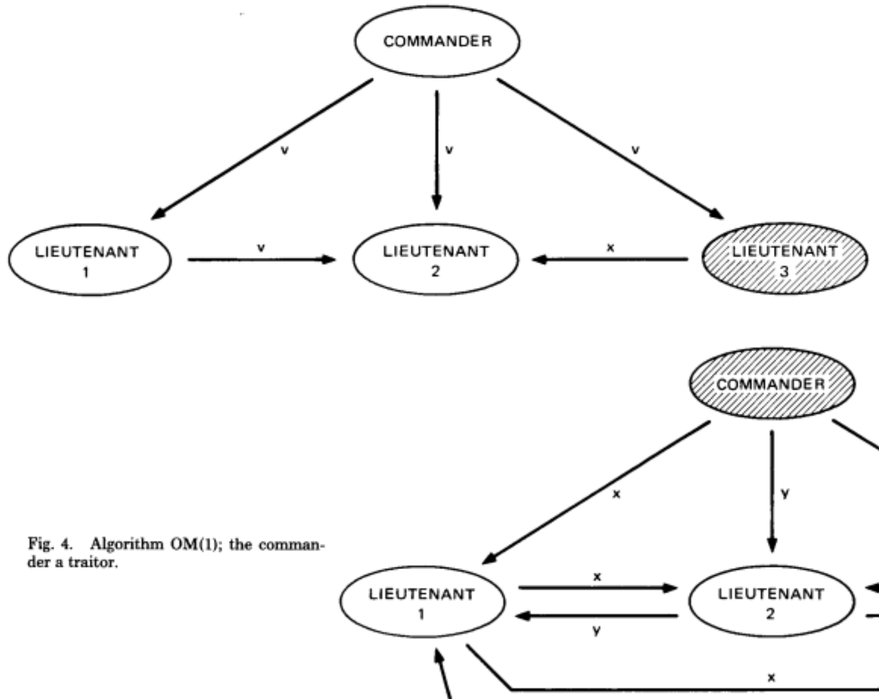


Fig. 3. Algorithm OM(1); Lieutenant 3 a traitor.

Fig. 4. Algorithm OM(1); the commander a traitor.

Lamport et al (1982).

"The Byzantine Generals problem".

doi:10.1145/357172.357176

Context: Proof-of-"costly resource"

https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto | Thu, 13 Nov 2008 19:34:25 -0800

James A. Donald wrote:
> It is not sufficient that everyone knows X. We also
> need everyone to know that everyone knows X, and that
> everyone knows that everyone knows that everyone knows X
> - which, as in the Byzantine Generals problem, is the
> classic hard problem of distributed data processing.

The proof-of-work chain is a solution to the Byzantine Generals' Problem. I'll
try to rephrase it in that context.

A number of Byzantine Generals each have a computer and want to attack the King's wi-fi by brute forcing the password, which they've learned is a certain
number of characters in length. Once they stimulate the network to generate a
packet, they must crack the password within a limited time to break in and erase the logs, otherwise they will be discovered and get in trouble. They only have enough CPU power to crack it fast enough if a majority of them attack
at the same time.

They don't particularly care when the attack will be, just that they all agree.
It has been decided that anyone who feels like it will announce a time, and
whatever time is heard first will be the official attack time. The problem is that the network is not instantaneous, and if two generals announce different
attack times at close to the same time, some may hear one first and others hear
the other first.

They use a proof-of-work chain to solve the problem. Once each general receives whatever attack time he hears first, he sets his computer to solve an
extremely difficult proof-of-work problem that includes the attack time in its
hash. The proof-of-work is so difficult, it's expected to take 10 minutes of them all working at once before one of them finds a solution. Once one of the
generals finds a proof-of-work, he broadcasts it to the network, and everyone
changes their current proof-of-work computation to include that proof-of-work
in the hash they're working on. If anyone was working on a different attack time, they switch to this one, because its proof-of-work chain is now longer.

Context: Definitions

Externalities

- “In economics, an externality is the cost or benefit that affects a party who did not choose to incur that cost or benefit.” Wikipedia

Sustainability

- “Meeting the need of the present without compromising the ability of future generations to meet their needs.” United Nations, 1987.

Negative Externalities (1)

1. Environmental sustainability of blockchains

As is widely known, updating and securing the Bitcoin blockchain – as well as other blockchains that make use of proof-of-work consensus – requires a great deal of electricity. By recent estimates, Bitcoin consumes the equivalent amount of electricity in a day as the country of Singapore[2].

EU Blockchain Observatory report for the European Commission

https://www.eublockchainforum.eu/sites/default/files/reports/report_scalaibility_06_03_2019.pdf

nature climate change

Comment | Published: 29 October 2018

Bitcoin emissions alone could push global warming above 2°C

Camilo Mora, Randi L. Rollins, Katie Taladay, Michael B. Kantar, Mason K. Chock, Mio Shimada & Erik C. Franklin

Nature Climate Change **8**, 931–933 (2018) | Download Citation

5939 Accesses | **18** Citations | **1437** Altmetric | Metrics »

A Publisher Correction to this article was published on 14 November 2018

A Matters Arising to this article was published on 28 August 2019

A Matters Arising to this article was published on 28 August 2019

A Matters Arising to this article was published on 28 August 2019

Bitcoin is a power-hungry cryptocurrency that is increasingly used as an investment and payment system. Here we show that projected Bitcoin usage, should it follow the rate of adoption of other broadly adopted technologies, could alone produce enough CO_2 emissions to push warming above 2 °C within less than three decades.

Negative Externalities (2)

Asbru, a former US Navy air base and home to several bitcoin farms

https://openskiesmagazine.com/the-land-of-fire-ice-and-gold/

Positive externalities

The well-known

- Permissionless
- Trustless
- Decentralized
- Transparent
- Censorship-resistant

but also driving

- Opensource
- Academia
  - Cryptography
  - Distributed systems
  - Game theory
  - Formal Verification

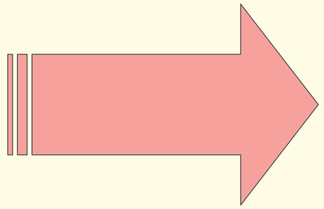
Ethereum 2.0 Proof-of-stake (1)

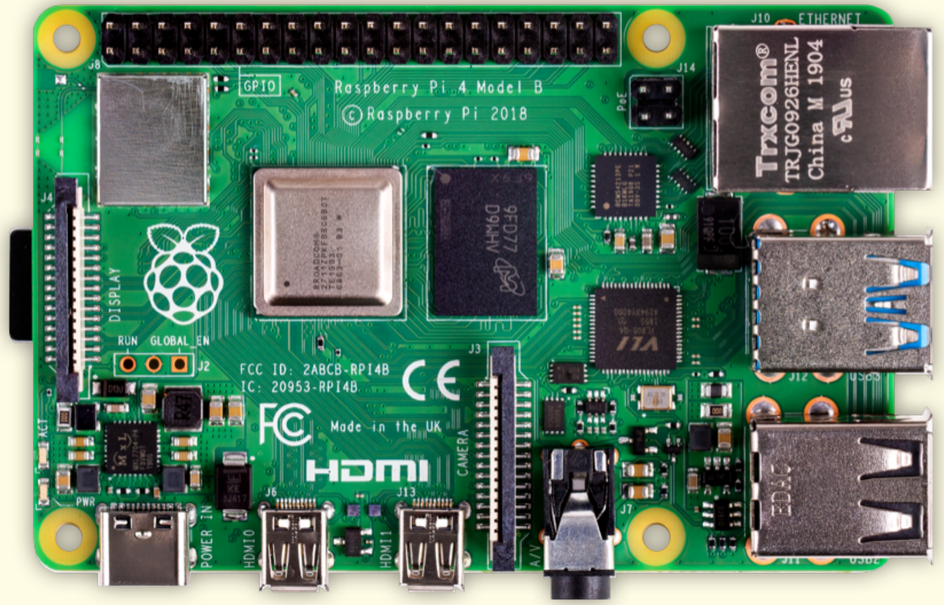
The scarce resource used to secure the system is now Ether itself

Ethereum 2.0 Proof-of-stake (1)

Credit: Stefen Chow

Ethereum 2.0 Proof-of-stake (1)





Credit: Stefen Chow

The gear

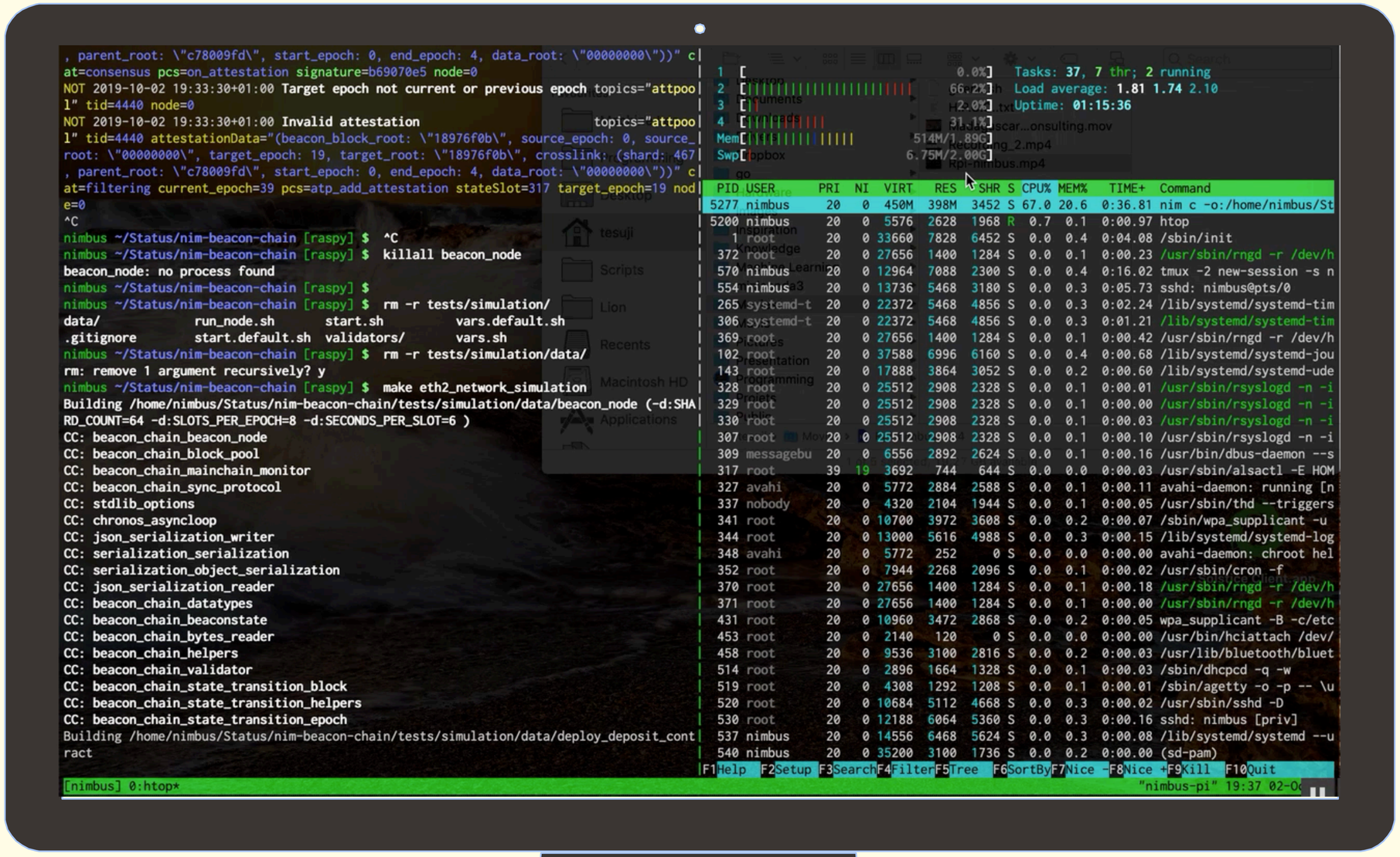
- Raspberry Pi 4 - 2GB* of RAM (8GB of unused swap)
  - Raspbian Buster Lite (official distribution, 32-bit)
- Nimbus Eth2 (https://github.com/status-im/nim-beacon-chain)
  - Preset “minimal”, same as interop (instead of “mainnet”)
  - 192 validators on 3 nodes
  - 64 shards (minimal 8, mainnet 1024)
  - 8 slots per epoch (minimal 8, mainnet 64)
  - Epoch every 48 seconds instead of 6 min 24 s

The gear

- https://github.com/ethereum/eth2.0-specs/blob/v0.8.3/configs/
- Epoch processing is costly
  - Justification and finalisation
  - Crosslinks
  - Rewards & Penalties
  - Validators eligibility and ejections
  - Slashings
  - Final updates (balances, RANDAO, …)

The unused gear

Credit: Fluke


, parent_root: \"c78009fd\", start_epoch: 0, end_epoch: 4, data_root: \"00000000\"))" c
at=consensus pcs=on_attestation signature=b69070e5 node=0
NOT 2019-10-02 19:33:30+01:00 Target epoch not current or previous epoch topics="attpoo
l" tid=4440 node=0
NOT 2019-10-02 19:33:30+01:00 Invalid attestation topics="attpoo
l" tid=4440 attestationData="(beacon_block_root: \"18976f0b\", source_epoch: 0, source_
root: \"00000000\", target_epoch: 19, target_root: \"18976f0b\", crosslink: (shard: 467
, parent_root: \"c78009fd\", start_epoch: 0, end_epoch: 4, data_root: \"00000000\"))" c
at=filtering current_epoch=39 pcs=atp_add_attestation stateSlot=317 target_epoch=19 nod
e=0
^C
nimbus ~/Status/nim-beacon-chain [raspy] $ ^C
nimbus ~/Status/nim-beacon-chain [raspy] $ killall beacon_node
beacon_node: no process found
nimbus ~/Status/nim-beacon-chain [raspy] $
nimbus ~/Status/nim-beacon-chain [raspy] $ rm -r tests/simulation/
data/ run_node.sh start.sh vars.default.sh
.gitignore start.default.sh validators/ vars.sh
nimbus ~/Status/nim-beacon-chain [raspy] $ rm -r tests/simulation/data/
rm: remove 1 argument recursively? y
nimbus ~/Status/nim-beacon-chain [raspy] $ make eth2_network_simulation
Building /home/nimbus/Status/nim-beacon-chain/tests/simulation/data/beacon_node (-d:SHA
RD_COUNT=64 -d:SLOTS_PER_EPOCH=8 -d:SECONDS_PER_SLOT=6)
CC: beacon_chain_beacon_node
CC: beacon_chain_block_pool
CC: beacon_chain_mainchain_monitor
CC: beacon_chain_sync_protocol
CC: stdlib_options
CC: chronos_asyncloop
CC: json_serialization_writer
CC: serialization_serialization
CC: serialization_object_serialization
CC: json_serialization_reader
CC: beacon_chain_datatypes
CC: beacon_chain_beaconstate
CC: beacon_chain_bytes_reader
CC: beacon_chain_helpers
CC: beacon_chain_validator
CC: beacon_chain_state_transition_block
CC: beacon_chain_state_transition_helpers
CC: beacon_chain_state_transition_epoch
Building /home/nimbus/Status/nim-beacon-chain/tests/simulation/data/deploy_deposit_cont
ract
1 [0.0%]
2 [66.2%]
3 [2.0%]
4 [31.1%]
Mem[514M/1.89G]
Swp[6.75M/2.00G]
Tasks: 37, 7 thr; 2 running
Load average: 1.81 1.74 2.10
Uptime: 01:15:36
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
5277 nimbus 20 0 450M 398M 3452 S 67.0 20.6 0:36.81 nim c -o:/home/nimbus/St
5200 nimbus 20 0 5576 2628 1968 R 0.7 0.1 0:00.97 htop
1 root 20 0 33660 7828 6452 S 0.0 0.4 0:04.08 /sbin/init
372 root 20 0 27656 1400 1284 S 0.0 0.1 0:00.23 /usr/sbin/rngd -r /dev/h
570 nimbus 20 0 12964 7088 2300 S 0.0 0.4 0:16.02 tmux -2 new-session -s n
554 nimbus 20 0 13736 5468 3180 S 0.0 0.3 0:05.73 sshd: nimbus@pts/0
265 systemd-t 20 0 22372 5468 4856 S 0.0 0.3 0:02.24 /lib/systemd/systemd-tim
306 systemd-t 20 0 22372 5468 4856 S 0.0 0.3 0:01.21 /lib/systemd/systemd-tim
369 root 20 0 27656 1400 1284 S 0.0 0.1 0:00.42 /usr/sbin/rngd -r /dev/h
102 root 20 0 37588 6996 6160 S 0.0 0.4 0:00.68 /lib/systemd/systemd-jou
143 root 20 0 17888 3864 3052 S 0.0 0.2 0:00.60 /lib/systemd/systemd-ude
328 root 20 0 25512 2908 2328 S 0.0 0.1 0:00.01 /usr/sbin/rsyslogd -n -i
329 root 20 0 25512 2908 2328 S 0.0 0.1 0:00.00 /usr/sbin/rsyslogd -n -i
330 root 20 0 25512 2908 2328 S 0.0 0.1 0:00.03 /usr/sbin/rsyslogd -n -i
307 root 20 0 25512 2908 2328 S 0.0 0.1 0:00.10 /usr/sbin/rsyslogd -n -i
309 messagebu 20 0 6556 2892 2624 S 0.0 0.1 0:00.16 /usr/bin/dbus-daemon --s
317 root 39 19 3692 744 644 S 0.0 0.0 0:00.03 /usr/sbin/alsactl -E HOM
327 avahi 20 0 5772 2884 2588 S 0.0 0.1 0:00.11 avahi-daemon: running [n
337 nobody 20 0 4320 2104 1944 S 0.0 0.1 0:00.05 /usr/sbin/thd --triggers
341 root 20 0 10700 3972 3608 S 0.0 0.2 0:00.07 /sbin/wpa_supplicant -u
344 root 20 0 13000 5616 4988 S 0.0 0.3 0:00.15 /lib/systemd/systemd-log
348 avahi 20 0 5772 252 0 S 0.0 0.0 0:00.00 avahi-daemon: chroot hel
352 root 20 0 7944 2268 2096 S 0.0 0.1 0:00.02 /usr/sbin/cron -f
370 root 20 0 27656 1400 1284 S 0.0 0.1 0:00.18 /usr/sbin/rngd -r /dev/h
371 root 20 0 27656 1400 1284 S 0.0 0.1 0:00.00 /usr/sbin/rngd -r /dev/h
431 root 20 0 10960 3472 2868 S 0.0 0.2 0:00.05 wpa_supplicant -B -c/etc
453 root 20 0 2140 120 0 S 0.0 0.0 0:00.00 /usr/bin/hciattach /dev/
458 root 20 0 9536 3100 2816 S 0.0 0.2 0:00.03 /usr/lib/bluetooth/bluet
514 root 20 0 2896 1664 1328 S 0.0 0.1 0:00.03 /sbin/dhcpcd -q -w
519 root 20 0 4308 1292 1208 S 0.0 0.1 0:00.01 /sbin/agetty -o -p -- \u
520 root 20 0 10684 5112 4668 S 0.0 0.3 0:00.02 /usr/sbin/sshd -D
530 root 20 0 12188 6064 5360 S 0.0 0.3 0:00.16 sshd: nimbus [priv]
537 nimbus 20 0 14556 6468 5624 S 0.0 0.3 0:00.08 /lib/systemd/systemd --u
540 nimbus 20 0 35200 3100 1736 S 0.0 0.2 0:00.00 (sd-pam)
F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit
[nimbus] 0:htop*
"nimbus-pi" 19:37 02-Oc

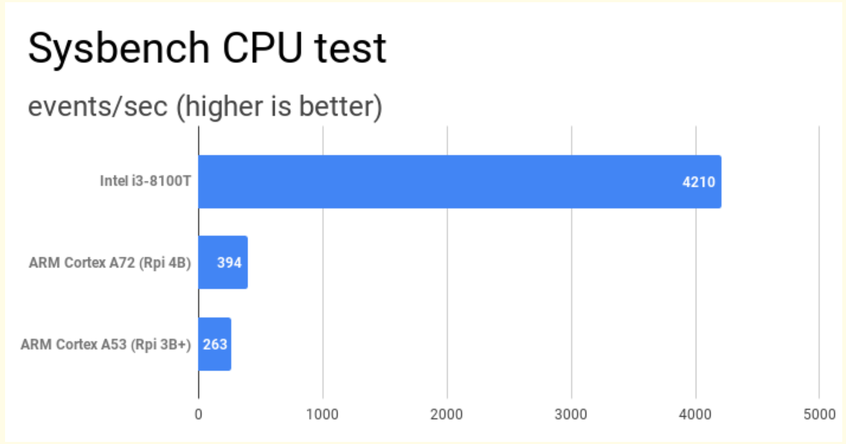
Key figures

- Power consumption: **7.6W**

  https://www.raspberrypi.org/magpi/raspberry-pi-4-specs-benchmarks/

- Eth staked: 192 x 32 Eth == 6144 Eth

- Rpi 4 is about 10x slower than an entry-level CPU (i3-8100T)



Credit: Jim Salters, Ars Technica

Outcomes (The Good)

- It run, even unoptimised on a non-trivial representative workload
- You don’t need a power plant in your backyard to protect yourself against electric downtime
- We have clear targets to optimise. It’s not “optimize everything”

Outcomes (The Bad)

- We can run the interop config or even a more demanding one
  - 192 validators instead of 16
  - 64 shards instead of 8
- But we cannot run “mainnet” configuration reliably yet.

  Processing takes over 6 seconds so the RPi cannot catch up

Outcomes (The Ugly)

- You cannot rely on blockchain to heat up your home in winter anymore

Credit: @PolDelft on Twitter

Future is challenging but promising

- RPi was run in 32-bit mode
- No multithreading yet
- No optimized BLS crypto library (like libsecp256k1 for Bitcoin)
- We’re not alone aiming at performance tour-de-force
  - Team Lodestar wants to do Javascript
- Predictions
  - Raspberry Pi 5
  - VPS + “Ethereum image”

The known unknowns

- Verifying validator attestations is costly. 30~50% depending of client.
- And when we scale to hundreds of thousands?
  - Aggregation protocol still in research
  - No need to connect to every peer
  - BLS crypto is being standardized across blockchains
- Phase 0 only
  - Phase 2: Smart Contract Execution.
  - Eth 1: 99.999%* of the time spent on consensus. *may contain rounding errors

Key takeaways

- Some teams have a vested interest in making Eth 2 as efficient as possible
  - Low-powered devices
  - Browsers
- Eth 2 will be a significant upgrade in terms of carbon footprint and electrical bill
- The consensus layer is solid and just need a little push.
- The big unknowns are signature aggregation, phase 1 and phase 2.

YOU FOUND A STAR

Speaker Track

dropparty.tech