Warthog Network

Rethinking the Blockchain

2024-10-29

Whitepaper

Table of contents

I. Introduction 4
I.1. History 4
I.2. What is Warthog Network? 4
II. Revolutionary features 5
III. Browser nodes 5
III.1. WasmFS 5
III.2. P2P communication over WebRTC 5
IV. DeFi2 5
IV.1. Custom Matching engine 5
IV.2. New DeFi features 6
Balance cloning 6
Paying dividends to holders 6
Scriptless airdrops 6
IV.3. Orderbook Surface Propagation (Later) 6
V. Janushash 6
V.1. Proof of Balanced Work 6
Introduction 6
V.2. Janushash 7
Balancing 7
V.3. Hashrate Decentralization 7
Fighting Farms 7
Satoshi's vision 7
V.4. ASIC Resistance 8
Inherited ASIC Resistance 8
Detection of suspicious hashrate 9
Simple Algorithm Adaption 9
V.5. Other Benefits 9
Escaping one-dimensional mining boredom 9
Favoring the little guy 10
VI. Technical Details 10
VI.1. Retarget Logic 10
VI.2. Emission Scheme 10
VI.3. Coin Precision 11
VI.4. One-of-a-kind chain descriptor based sync 11
VI.5. SQLite backed block store 12
VI.6. Account based architecture 12
VI.7. Fee specification 12
VII. Roadmap 2024 - 2025 13
VIII. Summary 13
Appendix A. Block Structure 13
A.1. Mining section 14

A.2. New address section 14
A.3. Reward section 14
A.4. Transfer section 15
Appendix B. Link Collection 16
Bibliography 16

I. Introduction

I.1. History

Originally, Warthog was written as a fun and experimental side project of its original developers Pumbaa, Timon and Rafiki has who work in blockchain industry. Initially there was no specific purpose or use case planned. Instead, the goal was firstly to revive the days when crypto was a fun and an interesting experiment, and secondly to try out new things and learn how blockchain technology works in detail.

However soon after its inception a vibrant community started to form around the young project and with it new contributors joined the project bringing a fresh wind of development support and innovative ideas. The project started to grow up and is still growing today.

I.2. What is Warthog Network?

Warthog is the first **Proof-of-Balanced-Work** (PoBW) Layer 1 network trying to push the boundary of what is technically possible by implementing multiple **highly innovative** and unique features including

- Janushash (first anti-farm balanced CPU/GPU combo PoBW algo)
- Chain descriptor sync (new efficient and resource-friendly sync)
- SQLite blockstore (easy cross-platform copy of chain file)
- Full browser nodes (start a full node by opening a website)
- WasmFS support (browser nodes can persist whole chain)
- WebRTC support (browser nodes can communicate P2P)

Unlike may other cryptocurrency projects we are **not a fork** of any other project. Instead, the **code is freshly written** in the C++ programming language. New software always bears the risk of unforeseen bugs but at the same time the real effort put into this project sets it apart from most competitors.

As a community project we are trying to be as transparent and fair as possible and avoid the fishy and questionable practice currently seen in most other new projects: we had a **fair launch** with **no team allocation or premine**, therefore donations for developments are always welcome!

Warthog is neither a company nor an organization. It is rather a loose team of passionate crypto enthusiasts who are contributing to the project in their free time. Bear in mind that team members may decide to leave the project at any time, in fact Timon, one of the original developers has already left the team.

Therefore we are constantly trying to expand the stronger community behind Warthog. The community is welcome to take actively part in the evolution of Warthog. The logo, the explorer and other milestones were contributed by volunteers. If you think you can help, please let us know!

II. Revolutionary features

There are several noatable features that are exclusive to Warthog across the industry. Section V.

III. Browser nodes

TODO

III.1. WasmFS

TODO

III.2. P2P communication over WebRTC

P2P communication between Browsers is possible via the WebRTC protocol. This protocol has to be signaled on connection establishment, i.e. a service needs to perform negotiation work before both ends are connectd directly. Nodes themselves will be configured to do this such that no external services will be necessary apart from being connected to the Warthog network.

IV. DeFi2

We will hard-code DeFi within Warthog nodes. On the one hand this puts additional burden on core developers because DeFi code must be written instead of leaving this task for smart contract developers. But on the other hand it yields several advantages and extends DeFi possibilities:

- Unified and systematic treatment of DeFi and assets
- No Service fragmentation like countless swap service clones
- No unfair practices for scammers like additional fees or supply inflation hidden behind smart contract logic.
- Clear foundation to check how each token was initially distributed (Fair auction or just minted out of thin air)
- Custom matching engine to **solve MEV extraction issue** that plagues DeFi, see Section IV.1..
- Additional features such as **clone balance distribution**, **dividends**, **scriptless airdrops**.

We call this concept and its extended capabilities *DeFi2.0*.

IV.1. Custom Matching engine

A matching engine that determines a single price for all buy and sell orders within a block will invalidate front and back running and save Warthog from MEV bots without the need to hide the order book. Such a matching engine cannot inspect DeFi orders separately as is done in today's DeFi implementations, but has to be aware of the collection of all buy and sell orders and pool liquidity to find a fair price. This means that **nodes need to be DeFi aware**

and this is the most important difference to DeFi based on smart contracts. To solve the MEV problem, nodes need to talk DeFi and have appropriate code at block processing level, or even at mempool and networking level, see Section IV.3. for details.

IV.2. New DeFi features

TODO

Balance cloning

Paying dividends to holders

Scriptless airdrops

IV.3. Orderbook Surface Propagation (Later)

At a later stage we will focus on implementing a new method to lower order fees. Nodes can inspect order prices and only share orders from their mempools which have better buy/sell price than what peers know. This way only the surface of the order book is transmitted between nodes up to the point where buy and sell overlap, i.e. where order matching is possible. This idea shall be more elaborated at a later stage.

V. Janushash

V.1. Proof of Balanced Work

Introduction

Proof of Balanced Work (PoBW) was first formulated in 2023 by Warthog community developer “CoinFuMasterShifu” [1] and was specifically implemented for Warthog for the first time. Compared to classical Proof of Work, the class of Proof of Balanced Work (PoBW) algorithms is very different. Instead of only employing one hash function they combine multiple hash functions in a multiplicative way. The mathematical theory of considering a multiplicative combination of hashes, i.e. hash products, was established in this paper for Warthog and currently there is no other crypto project using Proof of Balanced Work for consensus.

Combining different hash functions multiplicatively has the advantage that

1. different hash functions can be mined in parallel devices at different hashrates using a multi-stage filtering approach and
2. efficient mining requires mining of all involved algorithms for each block in contrast to previous failed attempts to construct multi-algorithm block chains by using individual difficulties for each algorithm (Myriad Coin, DigiByte, Verge). For example Verge was hacked by focusing only on one algorithm.

V.2. Janushash

Essentially, Proof of Balanced Work algorithms are simply the multiplicative combinations of existing hash functions. Warthog's Janushash algorithm combines two hash functions:

1. triple Sha256 (Sha256t) and
2. Verushash v2.2

Balancing

Energy-efficient mining of Proof of Balanced Work algorithms requires finding a good balance between Sha256t and Verushash hashrates. The best combination depends on hardware and energy cost but it is clear that mining without a GPU or with a weak CPU won't be competitive. The balancing requirement coined the name "Proof of Balanced Work".

V.3. Hashrate Decentralization

Fighting Farms

Interestingly, the Janushash algorithm keeps away both, GPU farms and CPU farms:

GPU farms usually save on the CPU side, because CPU performance is not relevant for mining GPU algorithms. Therefore such farms perform poorly on Janushash. GPU farm owners would need to make significant investments in efficient and performant motherboards and CPUs to improve their GPU/CPU balance for efficiently mining Janushash.

CPU farms perform very poorly on Janushash because of the lack of accelerated triple Sha256 hash evaluation. The same applies to most botnets.

This means large mining farms and botnets play a much smaller role in Warthog than they do in other proof of work cryptocurrencies, which increases decentralization of hashrate.

Satoshi's vision

Originally, Satoshi Nakamoto had an idealized hope for mining being a democratized way of establishing consensus. This can be seen for example in his famous whitepaper [2] where it says:

> Proof-of-work is essentially one-CPU-one-vote.

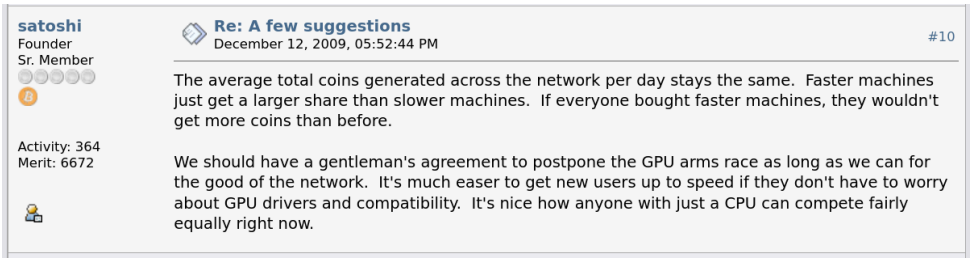
From this article [3] about Laszlo Hanyecz's correspondence with Satoshi we can observe that Satoshi was not amazed about the fact that GPU mining would disrupt this idealized hope:

> One of the first emails Satoshi had sent the man was in response to him describing his proposed GPU miner. Mainly, Satoshi was none-too-pleased, asking Hanyecz to slow down with this.

> Satoshi explained that, at the time, one of the biggest attractions possible is the fact that anyone can download Bitcoin and start mining with their laptops. Without that, it wouldn't have gained as much traction.

He knew that with the advent of GPU mining, many CPU miners would be kicked out of the network, which would be against his vision of fair, equal and decentralized mining. Therefore he hoped to delay this as long as possible. Figure 1 shows one of his posts on Bitcointalk.

Figure 1 — Satishi's hope to postpone GPU arms race.



We all know that his hopes have not been fulfilled, today Bitcoin is mined on specialized expensive hardware and only those with access to this hardware can participate in mining. After all, Satoshi was not able to solve the issue of centralized mining.

We are confident that the use of Proof of Balanced Work solves this issue to a large extent when the combined hash functions are carefully selected. In Janushash, the two hash functions Sha256t and Verushash where chosen to require a GPU and a CPU connected with sufficiently large bandwidth. This was done to target typical gaming PCs. As described above, with this choice farms cannot easily join the network without being forced to make additional investments just for mining Warthog. This democratizes mining and brings Warthog closer to Satoshi's vision.

V.4. ASIC Resistance

As technology advances, so does specialized mining hardware, especially when potential profits are high. There is nothing that can be done against this fact. However there are three reasons why Warthog is more robust against ASIC threats than other PoW cryptocurrencies:

Inherited ASIC Resistance

When it comes to ASIC resistance, Proof of Balanced Work is stronger than its strongest ingredient. To accelerate mining, an ASIC would need to be able accelerate computation of all combined hash functions to avoid a bottleneck

effect. In addition an ASIC would need enough bandwidth between the hardware sections computing different hash functions as well as calibration and tuning to optimize their intercommunication and coordination.

In particular, Janushash inherits ASIC-resistance from Verushash v2.2 which is currently mined on CPUs and GPUs, but not on FPGAs/ASICs, and the need to also require SHA256t hashrate makes Janushash even more ASIC-resistant.

Detection of suspicious hashrate

In traditional Proof of Work networks we only have one marker to analyze network hashrate, namely the network difficulty. It can be used to estimate the total hashrate of all miners in the network. However we cannot tell whether some actors use specialized hardware to gain an unfair advantage over normal miners.

Janushash however combines two hash functions and harnessing the probability theory and statistics, we can extract information about the Sha256t/Verushash hashrate ratio used to mine a block. This information is shown publicly in the blockchain explorer.

In addition to the network difficulty, this second marker provides useful information on the network hashrate: It allows to spot suspicious hashrate immediately. In Warthog it is much more difficult for ASICs to stay undetected because they must not only successfully mine blocks, but also mimic the hashrate ratio used by honest miners. This is another unique property of Proof of Balanced Work.

Simple Algorithm Adaption

The fundamental reason for the favorable properties of the Janushash algorithm is not the particular choice of the combined hash functions itself, but the choice to rely on Proof of Balanced Work to combine different hash functions multiplicatively. This means that if ASICs really join the network one day, we can simply exchange the combined hash functions, for example for Blake3 on GPU and RandomX on CPU, while preserving all the advantages listed here. Combining established hash functions allows the creation new algorithms fast while benefiting from their maturity and proven properties at the same time. This allows Warthog to adapt quickly when needed.

V.5. Other Benefits

Warthog tries to revive the good old days when mining was fun. The unique properties of the Janushash algorithm help to achieve this goal:

Escaping one-dimensional mining boredom

In a way, traditional mining in cryptocurrency is one-dimensional, the goal is simply to find the best hardware for evaluating some hash function. In

contrast, mining Warthog is two-dimensional: there are two hash functions Sha256t and Verushash v2.2, and both hashrates are relevant for the mining efficiency. This leads to much more versatile options and motivates miners to experiment with endless hardware setups. Vivid discussions about the best combinations bring Warthog mining to life.

Favoring the little guy

As explained above, established farms require substantial investments in order to mine Warthog efficiently and making such investment only for mining Warthog might not be reasonable for most farms.

On the other hand gamers usually have systems with modern platforms and CPUs paired with sufficiently good GPUs to mine Warthog efficiently. Since farms and botnets are less of a direct competitor in Warthog than they are in other Proof of Work cryptocurrencies, this will reflect in increased mining returns for the average gamer or miner, which will in turn contribute to Warthog's popularity.

VI. Technical Details

VI.1. Retarget Logic

Similarly to Bitcoin, the warthog blockchain will scale its difficulty periodically to adjust for changing hashrate. Changes in difficulty is partitioned into two phases:

1. In the initial phase the difficulty is adjusted every 720 blocks which corresponds to approximately 4 hours.
2. In the second phase the difficulty is adjusted every 8640 blocks which corresponds to 2 days.

The reason for this two-phase approach is the high variability of hashrate in early stages of a project's life which initially requires a more frequent difficulty adjustment. On the other hand too short intervals also have disadvantages such as the tendency to oscillate and a possibly higher impact of faked timestamps. Therefore the second phase stretches the difficulty adjustment interval after the initial phase.

While in Bitcoin the difficulty change is capped by factor 4, we have implemented a factor 2 cap because our difficulty adjustment is more frequent than 2 weeks.

VI.2. Emission Scheme

Warthog was started without any premined or reserved amount of coins on June 29, 2023. The project implements a classical halving-based emission scheme with halvings occurring every 3153600 blocks (every 2 years). The emission for the next 4 years is summarized in the following table:

Table 1 — Emission scheme

Date	Lifetime	% of total supply in circulation
June 29 2023	0 years	0%
~June 29 2024	1 years	25%
~June 29 2025	2 years	50%
~June 29 2026	3 years	62.5%
~June 29 2027	4 years	75%

There is no tail emission which means there is a hard cap of the amount in circulation. The hard cap is `18921599.68464 WART` (around 19 million coins).

Before halving occurs every block yields `3 WART` as miner reward. Since the block time is 20 seconds, every day approximately $\frac{60}{20} \times 60 \times 24 = 4320$ blocks and `12960 WART` are mined daily before halving.

VI.3. Coin Precision

The reference implementation uses the C++ data type `uint64_t` for storing amounts of `WART` . This is a 64 bit unsigned integer. To represent fractions of a coin these values are interpreted in fixed point arithmetic with 8 digits precision. This means that `1 WART` is internally represented as `uint64_t` number with value 100000000. The smallest representable step is `0.00000001 WART` and represented as `uint64_t` number with value 1.

For easier integration all API endpoints return both, the `WART` amount as a string (like `"amount": "12.0"`), and the internal integer representation indicated with label “E8” (like `"amountE8": 1200000000`).

VI.4. One-of-a-kind chain descriptor based sync

This project is an experiment where the developers try out new things and push the boundary of what is possible in blockchain technology. We invented a completely unique and new way of syncing nodes which is not presently not known to the industry.

Traditionally during synchronization new nodes request block bodies identified by block hashes. The replying node has to look up the block body based on the hash and then sends it back.

In contrast we have invented a node communication protocol which works without block hashes for block body lookup. In our setup nodes keep track on fork heights with other nodes. A chain descriptor is used to identify a specific chain on the peer. When a node appends to its chain, the chain descriptor remains unchanged, however the current chain descriptor is increased when the consensus chain switches to a longer fork. Block bodies for previous chains are also kept for some time in case a peer requests them.

When syncing nodes request block bodies identified by a chain descriptor and a block range. This way we avoid overhead in communication and lookup.

VI.5. SQLite backed block store

SQLite is a battle-proven and well-established embedded SQL database engine. Warthog nodes use SQLite as their main storage engine for both, blocks and state. Nodes also index transactions and can provide basic blockchain explorer functionality directly via API thanks to SQLite.

SQLite databases are also portable across 32-bit and 64-bit machines and between big-endian and little-endian architectures such that chain snapshots can easily shared. Furthermore SQLite supports transactions which are essential for data integrity even in case of a power outage or node crash.

The default SQLite database file name used for the chain is `chain.db3` and can be configured via the `--chain-db` command line option

VI.6. Account based architecture

Warthog implements an account based architecture. This is similar to Ethereum and different from Bitcoin's UTXO model. Every account along with its balance is stored in the `State` table of the chain database. For efficiency reasons accounts are referred by their id: Every account is assigned a unique auto-incremented id value on first use. This makes blocks more space-efficient since a block id only requires 8 bytes of storage whereas an address would require 20 bytes.

VI.7. Fee specification

For efficiency and compactness transaction fees are encoded as 2-byte floating-point numbers (16 bits), where the first 6 bits encode the exponent and the remaining 10 bits encode a 11 bit mantissa starting with an implicit 1. This means that fee values cannot be `0` and are of lower precision than regular amount values which use 4 bytes. A fee of value of `0` specified on transaction generation will automatically transform into the minimal fee value of `0.0000001 WART` .

VII. Roadmap 2024 - 2025

- [x] New “herominers” pool support (please aim for pool decentralization)
- [◧] Collect sufficient donation funds
- [◧] New website with design by BalkyBot (logo designer)
- [◧] Browser nodes
  - [x] Allow chain to be saved within browsers persistently in WASMFS.
  - [x] Port node code to Webassembly
  - [◧] Browser Node GUI (started)
  - [◧] Add new realtime API methods for browser nodes (started)
  - [x] Refactor network code to abstract away the communication layer (raw TCP, Websocket, WebRTC)
  - [x] Invent a robust protocol for exchangeing peers and negotiating (signaling) P2P WebRTC connections between nodes.
  - [◧] Protocol implementation
  - [ ] Testing
- [◧] DeFi2.0
  - [◧] Implement custom matching engine (early demo available)
  - [ ] Design database tables for modeling pools and cloning token balance with copy-on-write
  - [ ] Implement new token generation
  - [ ] Implement hard-coded pools with merged (liquity + limit orders) matching
  - [ ] Implement protocol for exchanging orders between nodes
  - [ ] Change block structure to support order matching

VIII. Summary

In this whitepaper we have presented the Warthog Network crypto project which stands out of the masses in terms of decentralization, technology and innovation. Unique flagship features include the specifically designed Janushash algorithm based on newly invented Proof of Balanced Work technology, which honors Satoshi’s ideals of mining democratization, the almost finished browser full nodes with planned P2P WebRTC communication and notably the planned DeFi2 implementation which will solve current DeFi’s biggest problem *MEV extraction* and add new features such as dividends, scriptless proportional airdrops to token holders, ICOs and more in a clean and user-friendly way. With the fair initial coin distribution based purely on mining and its thriving community Warthog’s future shines bright.

Appendix A. Block Structure

The binary content of a block is a concatenation of the following sections in their specified order:

1. Mining section

2. New address section
3. Reward section
4. Transfer section

Below we describe the above sections. All numbers and id values are in network byte order.

A.1. Mining section

This section allows miners to put 4 bytes of arbitrary data to affect the merkle hash.

Table 2 — Mining Section

byte range	content
1-4	arbitrary data

A.2. New address section

This section lists new addresses that receive payments in this block and therefore need to be added to the `state` table. This way they will be assigned a new id value which is referenced in the other sections to specify a particular account.

Table 3 — New Address Section

byte range	content
1-4	number `n` of new addresses
5-(4+n*20)	`n` addressess of 20 bytes each

Miners are responsible to ensure that the addresses appearing in the new address section are not already present in the state table and are actually referenced in this block. Otherwise the block is considered invalid.

A.3. Reward section

Mining reward is distributed to at least one reward address. (Need to be reworked as last commit change this)

Table 4 — Reward Section

byte range	content
1-2	number `r` of reward entry
3-(4+n*16)	`r` reward entries

Every reward entry consists of 16 bytes:

Table 5 — Reward entry

byte range	content
1-8	accountId
9-16	amount

The sum of the amounts received by the addresses listed in the mining reward section must not exceed the total mining reward (block reward + transaction fees), otherwise the block is considered invalid.

The total size of the mining section is `2 + r * 16` bytes.

A.4. Transfer section

The transfer section contains the transfers made in this block. Its binary outline is as follows:

Table 6 — Transfer Section

byte range	content
1-4	number `t` of transfer entries
5-(4+t*99)	`t` transfer entries

Every transfer entry has the following structure:

Table 7 — Transfer structure

byte range	content
1-8	fromAccountId
9-16	pinNonce
17-18	fee
19-26	toAccountId
27-34	amount
35-99	recoverable signature (65 bytes)

Each payment entry has length 99 bytes. Compare this to the average transaction size of around 200 bytes per Bitcoin transfer.

Appendix B. Link Collection

- Website: https://www.warthog.network/
- Github: https://github.com/warthog-network
- Gui Wallet: https://github.com/andrewcrypto777/wart-wallet
- PoBW whitepaper: https://github.com/CoinFuMasterShifu/ProofOfBalancedWork/blob/main/PoBW.pdf
- Janushash: https://warthog.network/docs/janushash
- Guide for pool devs: https://warthog.network/docs/developers/integrations/pools/
- Guide for miner devs: https://warthog.network/docs/developers/integrations/miners/
- Guide: https://github.com/warthog-network/warthog-guide
- Explorer: https://wartscan.io/
- Discord: https://discord.com/invite/QMDV8bGTdQ
- Telegram: https://t.me/warthognetwork
- Bitcointalk: https://bitcointalk.org/index.php?topic=5458046.0
- API documentation: https://github.com/warthog-network/Warthog/blob/master/doc/API.md
- Reddit: https://www.reddit.com/r/warthognetwork/
- Pool: https://warthog.acc-pool.pw/
- Wart-Dapp: https://github.com/warthog-network/wart-dapp/releases
- Coingecko: https://www.coingecko.com/en/coins/warthog
- Exbitron: https://exbitron.com/trade?market=wart-usdt
- Tradeogre: https://tradeogre.com/exchange/WART-USDT
- Xeggex: https://xeggex.com/market/WART_USDT
- Miningpoolstats: https://miningpoolstats.stream/warthog
- Coinpaprika: https://coinpaprika.com/coin/wart-warthog/
- Livecoinwatch: https://www.livecoinwatch.com/price/WarthogNetwork-WART

Bibliography

[1] C. M. Shifu, “Proof of Balanced Work: The Theory of Mining Hash Products.” Accessed: Oct. 20, 2024. [Online]. Available: https://raw.githubusercontent.com/CoinFuMasterShifu/ProofOfBalancedWork/main/PoBW.pdf

[2] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” May 2009, [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[3] A. Raza, “Laszlo Hanyecs Claims Satoshi Invented GPU Mining To Prevent 51% Attacks.” Accessed: Oct. 20, 2024. [Online]. Available: https://insidebitcoins.com/news/laszlo-hanyecs-claims-satoshi-invented-gpu-mining-to-prevent-51-attacks