Network boot parameters

Iurii Shyshatskyi [*1], Vinod Manoharan [†1], Taras Emelyanenko [‡1], and Lucas Leger [§1]

[1]Jax.Network, Kyiv, Ukraine

October 30, 2021

The launch of Jax.Network is approaching and it's time to discuss certain parameters that could be interesting for developers, miners, ASIC producers and researchers who want to estimate the prospective price of JAX coin. JaxNet is a protocol used in Jax.Network. It's described in our academic paper that can be found on arxiv.org.[3] Although it's a rather detailed description, there is a certain level of flexibility in the choice of certain parameters. We admit that some test results or future developments in hardware performance and network protocols might force us to adjust some of these parameters.

1 Shard chains

1.1 Initial shard count in Jax.Network MainNet is 3

Shard count is the parameter that directly affects the throughput of the blockchain network. The more shards you have, the higher tps can be achieved. However, maintenance of new shards has some cost. Therefore it's reasonable to start with a few chains and open extra shards when the network matures. Along with adoption of JAX, new shards will be created by miners according to the shard expansion protocol.

1.2 Beacon chain average block interval is 600 seconds

This parameter is the same as in Bitcoin. The same number is specified in our whitepaper and the rest of our documentation.

[*]iurii@jax.net
[†]vinod@jax.net
[‡]taras@jax.net
[§]lucas@jax.net

1.3 Shard chain average block interval is 37.5 seconds

This number differs from the one we wrote in the whitepaper. We decided to boot the network with block interval

$$600/16 = 37.5$$

seconds instead of

$$600/40 = 15$$

The reason for this update is that we don't know how active miners will be at the early stages of our network. We anticipate there will be some fluctuations in the hashrate and empty blocks. Also, we don't know how often miners will update jobs on their ASICs.

We emphasize that this change won't affect the throughput of shard chains since we increase the capacity of shard blocks proportionally. Also, we may switch values of parameters to the values written in the whitepaper in the future.

1.4 BC block body size is 768KB

We decided to increase the capacity of beacon chain blocks. This number is higher than numbers which appear in the early versions of the whitepaper.

1.5 SC block body size is 60KB

We increased the size of shard chain blocks to compensate for the decrease in intensity of their arrivals so that shard chain maximal throughput will remain the same.

1.6 BC difficulty epoch length is 2048 blocks

This parameter determines how often difficulty adjustments occur on the beacon chain. There are no changes here compared to the whitepaper.

1.7 SC difficulty epoch length is 4096 blocks

This parameter determines how often difficulty adjustments occur on the shard chains. So the length of the epoch measured in blocks has not been changed compared to the whitepaper. However, since the time interval between blocks on shard chains has been increased, the actual duration of the epoch is about 42 hours and 40 minutes.

1.8 BC K-coefficient epoch length is 4096 blocks

This parameter determines how often K-coefficient adjustments occur in Jax.Network. There are no changes here compared to the whitepaper.

1.9 SC K-coefficient epoch length is 65536 blocks

This parameter determines how often K-coefficient adjustments occur in Jax.Network. This parameter is not independent. It is determined by the block time intervals on chains and BC K-coefficient epoch length. It equals

$$4096 \cdot 16 = 65536$$

1.10 The BC timestamp window is 2 hours

The value of this parameter is the same as in the whitepaper and in Bitcoin.

1.11 The SC timestamp window is 15 minutes

In the whitepaper, the value of this parameter was 6 minutes. This time interval corresponds to 24 blocks on shard chains. Since the time interval between shard blocks has been increased, the timestamp window on shard chains has become longer too.

2 Genesis block

2.1 MainNet launch date is October 31

The date of the MainNet launch is October 31, 2021. It's the 13th anniversary of the Bitcoin whitepaper. All details will be specified in the separate announcement. The template of genesis will be published without one component. This component is the hash of the Bitcoin block. This block is about to be mined soon after the MainNet launch announcement. So anyone is able to start mining on top of the genesis block as soon as the anticipated Bitcoin block is mined. There is a small possibility that there will be a fork on the Bitcoin blockchain. This fork might induce the creation of two versions of Jax.Network. However, as soon as this fork gets resolved, we will learn what version of Jax.Network is valid.

2.2 Premine size is 36,000,000 JXN

The size of the premine is 9% less than it was announced before the public sale. For the public sale on Uniswap and Pancakeswap, about 40,002,164 WJXN tokens were issued to represent the premine. It was said in our tokenomics documentation that we would burn coins in the OpEx wallet, team wallet and ecosystem growth wallet, if the 30-day moving average coin price was above $1. We burnt 2.5 million coins during the public sale. However, the price was below the $1 threshold for a few weeks, so we stopped burning more WJXN tokens.

The team and the ecosystem growth wallets incurred a 6.25% cut each. On top of this, we burnt an extra batch of 670,676 tokens from the ecosystem growth wallet. The reason is that we keep full integers for the premine stack, setting it at 36 million coins.

2.3 BC block reward schedule

The block reward on the beacon chain is set at 20 coins per block after 5 years. The details will be specified in a separate paper.

3 Mining parameters

3.1 Mining hash function is SHA256d

We decided to use SHA256 (d) hash function for mining. Therefore every device which is used for mining Bitcoin or its forks can be used for merge-mining Jax.Network.

Merge-mining in Jax.Network is implemented in such a way that miners can merge-mine it along with any Bitcoin fork. So blocks merge-mined with BTC, BCH and BSV will be admitted in Jax.Network.

3.2 Hash sorting number is 10

Jax.Network uses a specific technique to reduce block propagation delay and the orphan block percent. The idea behind it is rather simple: to reduce the block propagation delay, you need to avoid situations, where one block producer generates blocks in many shards simultaneously. Such events often induce conjecture and increase block propagation delay.

We achieve this goal by distributing all chains into $2^{10} = 1024$ equivalence classes by assigning a chainID to every chain. A chainID is a number between 0 and 4095 that can be represented by 10 bits. Then we add an extra step to Proof-of-Work verification: the last 10 bits of the hash of the block candidate must represent the chainID of the target chain.

This technique is not new for blockchain space. The history of this idea could be traced back to the paper by Garay et al.[1] who introduced it under the name "2-for-1 pow".

In the future we might increase hash sorting number to 12 through the hard fork. So there will be 4096 equivalence classes.

3.3 Initial BC block difficulty is 2^{64} hashes

This parameter determines how hard it will be to find beacon chain blocks during the first epoch.

Notice that block difficulty here is expressed in hashes. In contrast, in Bitcoin difficulty is often expressed in "chunks" of size 2^{32} hashes which represent the range of the nonce field within the Bitcoin block header. If we convert 2^{64} hashes to "chunks", then it will be 2^{32} "chunks".

3.4 Initial SC block difficulty is 2^{60} hashes

Initial value of shard chain block difficulty will be 16 times less than on the beacon chain.

4 Tokenomics

4.1 The smallest unit of value on the BC is Haber-Stornetta

The Jax.Network team has decided to commemorate Stuart Haber and Scott Stornetta who were the first to describe blockchain as a method to keep data immutability. Their work was cited multiple times in the Bitcoin whitepaper. We decided to call the smallest non-divisible unit of value on the beacon chain after them. So in Jax.Network we have:

$$1 \text{ JXN} = 100,000,000 \text{ Haber-Stornetta}$$

4.2 The smallest unit of value on the shard chains is JURO

We decided to call the smallest non-divisible unit of value on shard chains with the Japanese name Juro. We set 1 JAX to be equal to 10,000 JURO.

$$1 \text{ JAX} = 10,000 \text{ JURO}$$

4.3 The initial value of the K-coefficient is 2^{-60} JAX/hash

It means that on average 2^{60} hashes should be computed in order to get 1 JAX. The value of the K-coefficient might be adjusted by miners at the end of every K-coefficient epoch. The minimum cost of an 8-byte chunk in SC block is 2 JURO. Shard chain transactions in Jax.Network have a minimum value of the transaction fee. It's calculated based on the space that they occupy in shard blocks. We define the minimum unit of space in shard blocks to be 8 bytes. We set the minimum price of this chunk to be 2 JURO.

4.4 BC block reward locks

In order to improve tokenomics of JXN coin, we set up timelocks on miners BC block rewards. It works as follows. Let x be a reward for the BC block of certain height. We know that x is not less than 20 JXN. We set a time lock on $(x - 20)$ coins out of x. The duration of this lock is $36,000$ BC blocks. So the locked coins could be transferred by the block producer after around 250 days.

The remaining 20 coins could be immediately used by the miner or get burnt if he decides to print JAX coins instead of BTC and JXN coins.

5 Protocol rules

5.1 Shard expansion protocol rules

One critical point of the JaxNet protocol is a shard expansion protocol. It determines the conditions which should be satisfied to open a new shard. The first condition is the support by the majority of miners, who put flags to BC blocks to indicate their support. The second condition is sufficient mining activity on shards.

We planned to set a third condition. According to this rule, miners of the BC would be minting JAX coins that would be locked and get released in the new shard chain upon its creation. It would be forbidden to open new shards unless there would be a sufficient amount of JAX coins to pour into them.

We decided to discard this condition and not to mint JAX coins for new shards in order to avoid possible fluctuations of the price of JAX coin.

5.2 K-coefficient adjustment rules

According to the whitepaper, we planned to set strict rules which determine the value of the K-coefficient based on the difficulty of previous BC blocks. We decided to allow miners to vote on the value of the K-coefficient and determine it through a consensus. So

the value of the K-coefficient for current epoch (n) is a median value of the vote_K field in the $(n-2)$ EPOCH.

5.3 Block height in block headers

During code review and testing, we determined that it's very convenient to have a block height included into block headers. We found this feature very useful for blockchains based on Merkle Mountain Range. It allows easy navigation for super-light clients based on ideas of FlyClient[2]. Also, it becomes easier to arrange hard forks which affect shard chains with this feature.

References

[1] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. "The bitcoin backbone protocol: Analysis and applications". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2015, pp. 281–310 (cit. on p. 5).

[2] Benedikt Bünz et al. "Flyclient: Super-Light Clients for Cryptocurrencies." In: *IACR Cryptology ePrint Archive* 2019 (2019), p. 226 (cit. on p. 7).

[3] Iurii Shyshatskyi et al. "JaxNet: Scalable Blockchain Network". 2021 (cit. on p. 1).