This is the first time eccentricexit has posted — let’s welcome them to our community!

eccentricexit **Apr 28**

Hit the nail on the head.

Not to mention that compared to blobs, bumping gas limits is a contentious change and has a much lower bang for the buck.

dankrad **Apr 28**

> MicahZoltu:
>
> It feels like this should be a function of epoch time, not a function of number of epochs alone. There have been discussions of decreasing block times and depending on how that is implemented, that could result in a change in epoch length. If this algorithm is a function of epoch count alone, then it could end up accelerating towards its target too quickly.

If we change the slot time, we would also have to change the gas limit (as what we are really targeting is a gas limit per unit of time, not block). I think it’s ok to update this schedule with such a change.

It’s been a while since we’ve seen 0xBreadguy — their last post was 1 year ago.

0xBreadguy **Apr 28**

It’s a signal towards application builders that there is a future which encompasses scale to handle their growth.

Teams have left mainnet to this point because they were told “mainnet isn’t your home.”

Low gas prices on mainnet today should not be used as an excuse to not push to make the chain more inhabitable for applications and users.

dankrad **Apr 28**

> MicahZoltu:

> MicahZoltu:
>
> If the network can handle X gas per block today, then gas limits should be set to that. We should not assume that future handling of gas is a given and we should only increase the gas limit *after* clients can handle such an increase.

There is no exact, hard limit of how much the network can handle. There is currently no reason to believe what is proposed here poses direct safety issues: **https://x.com/notnotstorm/status/1915439259630317995**

The purpose of having a schedule is to give applications predictability, and make us solve performance issues as a priority. A lot of what needs to be done is performance engineering; you cannot do this well without understanding what the end goal is.

We also have a strong pipeline of changes that will make this a reality. I think we should commit with the end goal in mind, rather what we have been doing so far, which has led to stagnation of the Ethereum network (e.g. still no DA scaling today, more than 5 years after the rollup-centric roadmap became the goal). I think it's time to shake things up a little.

TimDaub **Apr 28**

As long as these arguments are not included in the EIP you are just hallucinating what you personally feel are the motivations for Dankrad to propose the EIP. Not meant as an offense ofc, but this is just structurally what can be observed.

Btw. I also don't think it is enough for @dankrad himself to state in the comments here or elsewhere what his motivation is for submitting. This has to be part of the EIP IMO. Otherwise Twitter etc. is going to front run the EIP with strawmans

---

dankrad **Apr 30**

Writing more about the reasons for this EIP. To be upfront, it is unconventional. I do think it is time for being unconventional, because the current way of doing things is likely to make Ethereum irrelevant over the next 5-10 years.

Key considerations why I think we should commit to this gas limit change schedule:

1. Strategic: Things happening on one layer (composability and UX-wise) is much more strategically important than previously thought.
2. Technical: Validity proving Ethereum L1 at 1-block latency will become possible this

year, and DAS with PeerDAS will also become reality
3. Execution: Working backwards from a goal tends to have better outcomes than making incremental changes as they become possible.

1. Strategic

Ethereum L1 needs to be the economic center of Ethereum. There are a number of reasons for this:

- Ethereum L1 will probably always drive fee revenue, as the moat for DA is much lower
- If L1 is unimportant and loses its attraction of liquidity and DeFi, there will also be less of a reason for L2s to even remain attached to Ethereum
- The Ethereum ecosystem does compete with other ecosystems, who are eager to get its market share. Fragmenting L1 liquidity across a number of different L2s is a very good way to lose this battle

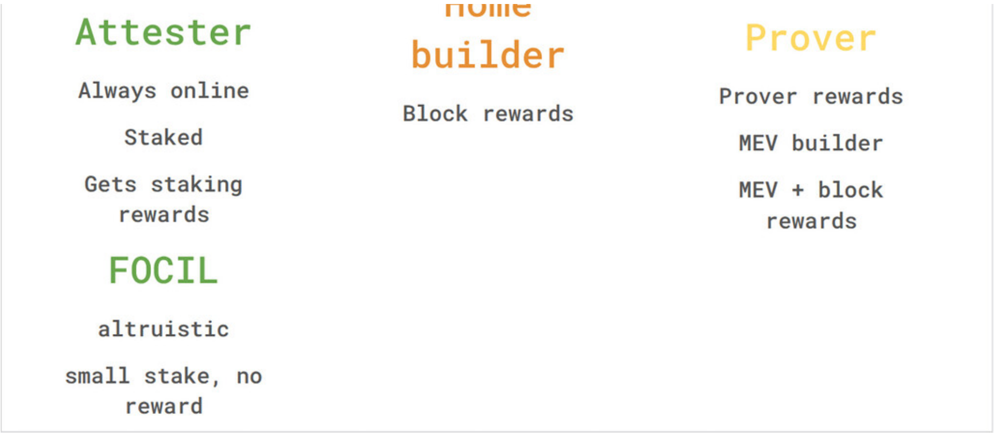
2. Technical

In the last year, proving Ethereum L1 blocks became first possible, and is now cheap (typical proof cost per block is a few cents: **https://ethproofs.org/**). Within this year, we will have proving within a single slot delay, and all teams expect order of magnitude improvements to continue for at least a few years.

This means that for the first time, we will have the ability to significantly scale the L1 so that a lot of activity can continue to happen on it. Not just just do 10x the scale, we can do 100x to 1000x the current scale while keeping Ethereum's most important properties:

- Verifiability (it is very easy to check that the current continuation of the blockchain is according to the rules)
- Censorship resistance (we can guarantee that any paying transaction will get included)

Ethereum's current node architecture is a copy of Bitcoin's from 2009. We will need to update the Ethereum node types for the 2020s and 2030s. Some roles (full nodes, attesting, FOCIL) are likely to be even lighter than they are today, while others (building, proving) will become more beefy; however, all the "beefy" node types come with an 1-out-of-n honesty assumption, making sure that they can always be easily replaced.



Barnabe discusses his vision for different node types more on this call (**https://www.youtube.com/live/m5HFO4DYckQ?si=4fD0Z7PkJM-4pBpX**). The key to maintaining security and keeping Ethereum unstoppable that all node types can still be run from home in some places.

3. Execution

In the past, we have been very reluctant to do certain things:

- Commit to timelines
- Plan actively for the future, several forks ahead
- Execute efficiently

Committing to a scaling timeline makes it clear what the goal is, and lets us plan backward from the goal:

- Most critical upgrades for Glamsterdam:
  - **Delayed Execution**
  - Shorter slot times
  - **Aggressive history expiry**
- L1 scaling upgrades over the next 2 years:
  - **Block-level access lists + parallel execution**
  - **Network-level erasure coding for execution blocks**
- Rollout of full zkEVM
  - Optional (this year) and then enshrined zkEVM proving (ca. 2 years)
  - Execution payload in blobs (to use DAS) (might be combined with network level erasure coding)
  - FOCIL to ensure continued censorship resistance

In addition to hard forks, scaling the EL 100x will require performance engineering. Having a concrete goal in mind will let us prioritize this work as well as the concrete upgrades as needed. A database for a 5x scaled EL might look very different from one upgrades as needed. A database for a 5x scaled L1 might look very different from one that is scaled to 100x. The mempool certainly does. A lot of decisions become much easier if we know where we want to go.

The same goes for application developers. Ethereum L1 is currently still the home for DeFi – this might not be true for much longer if we don't start strongly supporting applications. An important step in this direction is giving them dependable scaling timelines.

Doesn't that make us like a Datacenter chain or Solana?

I think it is irrelevant whether some superficial aspects of Ethereum look more like Solana or not. The core value proposition of Ethereum is not the home staker, it is verifiability and censorship resistance. While the future world looks significantly different from now, I would argue it's not clearly better or worse in those aspects, just a different set of tradeoffs. 99%+ of users today do not run their own node but use their wallet's default RPC. Having ZK verifiability will actually make it easier (obviously it's still going to be hard work getting it integrated into wallets). FOCIL or MCP can probably bring us better censorship resistance than we have today.

Another question might be – are we playing Solana's game? Why should we IBRL, aren't we certain to lose this?

I think it is not so certain we will lose this game. Ethereum L1 so far has the most liquidity of all chains. Why is it still here?

- Existing DeFi moat
- IBRL is not the only thing that matters
  At least as far as scaling is concerned, I do believe that at 100x the current scale, Ethereum L1 can support a very large range of value transaction that competing with it simply on scaling terms is not and interesting game to play anymore. On shorter block times, I think we are unlikely to go below ca. 1s to change proposers, but there are other options to improve UX (preconirmations) and DEX performance (MCPs).

Uniquely, Ethereum has a huge moat in DeFi liquidity, and those applications benefit significantly from being colocated. What we need to do is support these applications and making Ethereum the best home to them.

Conclusion

Due to these updates to the strategic and technical environments, I think by far the best strategy for Ethereum from here is, in addition to supporting L2s via blob scaling and UX/

strategy for Ethereum from here is, in addition to supporting L2s via blob scaling and UX/interop improvements, to significantly scale the L1. The endgame is scaling 100x-1000x. We need to commit to it as soon as possible, both because builders and applications need predictability, and because we need to prioritize properly so that it can actually get executed.

last visit

MicahZoltu Apr 30

> dankrad:
>
> If L1 is unimportant and loses its attraction of liquidity and DeFi, there will also be less of a reason for L2s to even remain attached to Ethereum

Why do we want/need L2s to be attracted to Ethereum? What is the problem if some L2s decide to build on Polkadot, Celestia, Solana, etc.? What harm is caused to Ethereum in such a world where that happens sometimes (but not all the time)?

> dankrad:
>
> The Ethereum ecosystem does compete with other ecosystems

I would like to see this reasoned out more thoroughly. Why does Ethereum need to compete with other ecosystems? What axis does it need to compete on? All of them, or is a particular niche sufficient?

> dankrad:
>
> 2. Technical

This section talks about why you think we can get away with a gas increase, but it doesn’t really speak to the motivation for why we need an increase, or more importantly why we need a gas increase right now. When you integrate all of this into the EIP’s motivation, I recommend leaving this part out or changing it significantly.

> dankrad:
>
> Committing to a scaling timeline makes it clear what the goal is, and lets us plan backward from the goal:

We can have a goal and work towards it without also counting our chickens before they hatch. We can say that we want to increase gas limits, and outline a roadmap that will enable us to do so, but wait to actually raise the gas limits until such time as those roadmap items are complete. If the goal of this strategy is to try to force our future hands, then I think that should be stated very clearly in the motivation section without beating around the bush.

---

green Apr 30

I was writing a reply on this thread with other ideas of how to "Increase the Gas Limit", but ended up using it as the response of a different thread. Still related though:
Formalizing decentralization goals in the context of larger L1 gaslimits and 2020s-era tech - #18 by green

---

Giulio2002 Apr 30

I'm glad this is no longer a taboo topic and that we can discuss it seriously.

I agree with almost everything you've said: Ethereum does have competitors, and it needs to make strategic decisions to maintain its status as the world computer. I also believe Ethereum wasn't built around home stakers—they were simply a "nice" side effect of The Merge. However, maintaining that narrative is costly, so dropping home stakers seems like the right compromise.

That said, a 100× improvement strikes me as unrealistic. A 10–20× gain (around 300 TPS) feels more attainable. The challenges are largely engineering-driven: so far, only Erigon has been engineered to handle state growth without breaking RPCs or write throughput. To hit 100×, you'd first need to shift engineering priorities across many teams (including ours—we do have performance gaps in some areas, though I'm comfortable with them given our core goals). Second, you'd require multi-order-of-magnitude "L1 scaling" upgrades, and we have no history of handling more than a single order of magnitude. That said, I could certainly be proven wrong.

I need to do some research in how you plan to achieve a 100x still. Nonetheless, I think it is doable on perhaps a larger time horizon (say 6 years)

This is the first time Sirmoremoney has posted — let's welcome them to our community!

Sirmoremoney Apr 30

You dropped this Sir 👑

As correctly pointed out, if ethereum remains anticompetitive, we will be the Nokia of blockchains in 1-2 years.

It is already happening. We completely lose our relevance if we don’t dramatically change.

I also appreciate the top-down approach to scaling the L1. Setting the goal to scale 100-1000x then planning and executing to make it happen.

This is the first time Stoff81 has posted — let’s welcome them to our community!

Stoff81 **May 1**

> Giulio2002:
>
> I’m glad this is no longer a taboo topic and that we can discuss it seriously.

Why has it been taboo till now? What are the “downsides” of increasing gas limit. Is it just state size?

So this proposal is looking to add state size (ie node HDs) to drive better economic incentives generated from fees. These fees are captured by node operators.

It’s been a while since we’ve seen Ariiellus — their last post was 3 years ago.

Ariiellus **May 1**

Today home stakers (soon home builders) can run on 2TB but many people recommend starting with 4TB.

This means that home stakers could be affected pretty soon?

> MicahZoltu:
>
> gas limit would increase the worst case scenario of state growth to 14TB/year (up from 140GB/year).

Or will they need to prune their nodes constantly in order to not run out of storage?

(increasing bad UX for NO). Also, this will impact in the minimum bandwidth necessary.

I’m agree on L1 being the economic center on Ethereum but this proposal feels that is leaving some aspects out of the table and seeing only one side of the moon.

> dankrad:
>
> Fragmenting L1 liquidity across a number of different L2s is a very good way to lose this battle

If the goal is to get back to mainnet and drop the rollup-centric roadmap, then why keep pushing the intents framework / interoperability efforts? Let’s not start a new battle front inside our barracks.

Solving the L1/L2 liquidity fragmentation UI/UX will increase the MOAT for eth eco.

MicahZoltu May 1

> Ariiellus:
>
> they need to prune their nodes constantly in order to not run out of storage?

The upper limit of 14TB/year I quoted was actual state, assuming you are running completely pruned with no overhead.

In reality it is quite unlikely that 100% of gas is spent on storage, but on the flip side there is also a significant amount of overhead beyond the raw state (like old state if you don’t prune instantly, the supporting data structures, indexes, etc.)

Pmatt328 May 2

As home staker, I totally and fully support this!

we are late but better starting now that never

Pmatt328 May 2

Rollup-centric raodmap was a clear mistake from start, glad to see them changing on this is the only thing that isn’t making me lose hope

Pmatt328 May 2

You cant describe it better, especially from an home staker point of view

forcing us to process blob to help killing our investment is not a good loyalty pratice

kladkogex May 2

> magicians:
>
> @dankrad

Hey Dankrad,

I completely agree with others — this is potentially a fantastic step forward!

A few remarks:

I understand why you’re proposing a fixed formula — it pushes progress and helps avoid the need for consensus every time the block limit is raised. That said, if you go this route, there needs to be a pre-assessment from client teams to ensure the schedule is feasible.

Raising the gas limit 100x roughly targets an end goal of around 1000 TPS.

For EVM implementations in C++ or Rust, this is achievable with relatively minor modifications.

The main bottleneck will be updating the full historical state, particularly for the few nodes that store the entire state. The issue is that current state updates are fully single-threaded and sequential, so adding a powerful multicore machine doesn’t help. At 1000 TPS, full-history nodes may not be able to keep up with nodes that only maintain the current state.

The simplest approach would be to shard the Merkle trie so it can be stored across multiple key-value database shards (e.g., LevelDB). The easiest method is to shard by the first bytes of the key.
You could then have, say, 256 shards, each potentially stored on separate SSDs or even separate virtual machines. With this kind of sharding, performance can scale nearly linearly with the number of shards

linearly with the number of shards.

However, this isn’t currently possible with the existing Merkle trie, as it can’t be represented as an aggregation of shard-level Merkle roots. This is a fundamental limitation — but one that could be addressed with a relatively simple change to the Ethereum spec.

Then there’s the issue of speeding up the consensus layer. I believe reaching 1000 TPS will require a formalized consensus specification. The current spec has many gaps, leaving clients to make ad hoc decisions. The Ethereum Foundation should launch an incentivized testnet to uncover security vulnerabilities and consensus instabilities. When PoS was launched, independent security researchers had no real opportunity to experiment — let alone receive bounties.

Like this attack

> **Kamikaze Attacks on ETH2**
>
> There is one type of an attack on ETH PoS, where attacker is willing to sacrifice 32 ETH slashed to potentially have a much larger gain. If an attacker shorts a significant amount of ETH, then the potential gain from even temporarily disruption...

I remember submitting it to the client team and the Ethereum Foundation, but I never received a single response—not even access to a testnet to try it out. I’m now considering building a testnet myself. In my opinion, the attack is entirely feasible and could potentially bring down the entire system.

This is the first time sanu2002 has posted — let’s welcome them to our community!

---

sanu2002 **May 4**

> dankrad:
>
> s

If Ethereum L1 becomes 100x–1000x faster and can handle massive TPS, then what’s the long-term role or need for L2s? Will L2s still be useful, or will they be abandoned once L1 can handle everything?

kladkogex May 5

Rollups are anyway either dying or turning into corporations like Base.

If Ethereum hits 1000 TPS, we'll finally have the bandwidth for real composable dapps. Definitely people that need decentralization wont use or care about rollups. People that need to trade memcoins will use Base. May be Binance will have a rollup like Base. Other rollups will die since they have no purpose.

ETH pushed users away for years with high gas fees and no real consumer focus. Gamers, retail—gone. Winning them back now is *much* harder than keeping them in the first place.

Here's the real bottleneck: **finality.** To make ETH consumer-grade, **finality needs to drop to a few seconds.**. Otherwise gamers and consumers wont come back to the ETH.

That's a *much* tougher challenge than just cranking up TPS having how outdated Ethereum's consensus layer is compared to other blockchains.

This is the first time omidm.eth has posted — let's welcome them to our community!

omidm.eth May 8

I'm a big proponent of scaling Layer 1, but I see a few issues with simply raising the gas limit. We need to match capacity with actual demand—just increasing the gas limit won't boost TPS in the short term. Instead, our first priority should be to shorten block times, giving users a faster, more convenient way to transact and creating a better environment for asset trading.

For instance, we could:
1. Reduce block time to 11 seconds to start.
2. Then cut it by one second each month until we hit the optimal minimum that maintains chain stability.
3. Once we've reached that stability cap, we can begin increasing the gas limit—or consider a hybrid approach combining both strategies.

This staged plan helps ensure reliability while gradually improving throughput.