Why Monad: decentralization + performance

Decentralization matters

A blockchain has several major components:

- Consensus mechanism for achieving agreement on transactions to append to the ledger
- Execution/storage system for maintaining the active state
-

In increasing the performance of these components, one could cut corners, for example by requiring all of the nodes to be physically close to each other (to save on the overhead of consensus), or by requiring a huge amount of RAM (to keep much or all of the state in memory), but it would be at a significant cost to decentralization.

And decentralization is the whole point!

As discussed in[Why Blockchain?](#) , decentralized shared global state allows many parties to coordinate while relying on a single, shared, objective source of truth. Decentralization is key to the matter; a blockchain maintained by a small group of node operators (or in the extreme case, a single operator!) would not offer benefits such as trustlessness, credible neutrality, and censorship-resistance.

For any blockchain network, decentralization should be the principal concern. Performance improvements should not come at the expense of decentralization.

Today's performance bottlenecks

Ethereum's current execution limits (1.25M gas per second) are set conservatively, but for several good reasons:

- Inefficient storage access patterns
- Single-threaded execution
- Very limited execution budget, because consensus can't proceed without execution
- Concerns about state growth, and the effect of state growth on future state access costs
-

Monad addresses these limitations through algorithmic improvements and architectural changes, pioneering several innovations that will hopefully become standard in Ethereum in the years to come. Maintaining a high degree of decentralization, while making material performance improvements, is the key consideration.

Addressing these bottlenecks through optimization

Monad enables pipelining and other optimizations in four major areas to enable exceptional Ethereum Virtual Machine performance and materially advance the decentralization/scalability tradeoff. Subsequent pages describe these major areas of improvement:

- [MonadBFT](#)
- [Deferred Execution](#)
- [Parallel Execution](#)
- [MonadDb](#)
-

On this page * [Decentralization matters](#) * [Today's performance bottlenecks](#) * [Addressing these bottlenecks through optimization](#)