

PCP - Parameter Committee Proposal

This form is used to initiate discussions in the parameter committee, an advisory group consisting of experts in the fields of Network, Technical and Economic principles and Cardano protocol parameters. Approved PCPs contain a clear description of the topic and questions formulated neutrally, enabling the committee to give a recommendation.

| PCP # | Title | Proposer | | |
|--|---|-------------------------------|---|--|
| 00? | Increasing either max_tx_ex_mem or max_block_ex_mem | Pi Lanningham | | |
| | | Email/Contact | | |
| | | pi@sundaeswap.finance | | |
| Description | | Date | | |
| <p>The cardano protocol has several parameters to limit the resources consumed per transaction and per block. This is important from a denial of service perspective, to allow honest actors to continue to publish transactions, to limit the amount of work done by the stake pool operator network, and to put bounds on the propagation and adoption times for blocks. The last in particular is particularly sensitive: if the block propagation time is an appreciable fraction of the average slot time for an extended period, it has implications on the security arguments of the Ouroboros consensus algorithm.</p> <p>However, they also dramatically limit the amount of economic activity that can settle on-chain. With an active DeFi ecosystem leading into an optimistic market, there is active agitation and worry about whether the current limits will be sufficient. As soon as one of the execution unit limits is reached, only “simple transfer” transactions can fit into the block to fill up the remaining space.</p> <p>Any of these limits ultimately cause queues and backpressure to form, either at the network level (to settle transactions) or at the dApp level (batching and processing the execution of user orders, for example on a batch).</p> <p>This is particularly relevant to me twice over: as a key builder on the SundaeSwap protocol, and as a survivor of the SundaeSwap launch. 😊 At the time, with intense demand for the SundaeSwap protocol, protocol queue times reached upwards of 11 days at the peak, an obviously abysmal experience for users.</p> <p>One of the major limiting factors at the time was block size, because of the requirement that scripts be attached to every transaction that executed them. As part of that, we saw several block size increases to alleviate the pressure.</p> <p>With the advent of Plutus v2 and reference scripts, the next major bottleneck is poised to be memory units.</p> | | 19 Dec 2023 | | |
| | | Relevance/Importance | PC Input required | |
| | | Focus fields | <ol style="list-style-type: none"> Technical | |
| | | Parameters primarily affected | | |
| | | max_tx_ex_mem | The maximum limit for memory in a transaction. | |

SundaeSwap’s contracts, for example, hit the memory limit for transactions at 8 orders per batch. Our new v3 contracts hit that limit much higher (benchmarked as high as 25 orders), but an active trading environment could still cause frustrating queues to form. Raising this limit per transaction could allow many protocols to fit far more economic work “per transaction”, resulting in fewer transactions, and ultimately reducing load on the chain on all dimensions.

Additionally, those conditions in early 2022 were created with just a few protocols receiving active traffic. Today, we have dozens of DeFi protocols, with many many more preparing to launch. These protocols will be vying for space within the block, and aggressively trying to crowd each other out. Because of that, we should also inspect and consider the block limit parameters.

Of the three main dimensions we can tweak, I believe there is the strongest evidence for memory units having the biggest impact on the user experience, while also having the smallest impact on node requirements or block propagation time.

First, [Smaug has done some analysis](#) on the correlation between transaction steps and transaction memory units which shows that we are almost always bottlenecked on memory as opposed to steps. Based on this research, it suggests we could double the memory units before we start being bottlenecked by transaction size or transaction steps.

Second, [myself and a few community members have done some work](#) to correlate the transaction memory units to block propagation time. The result suggests that memory units have very little impact on block propagation times.

Logically, this also makes sense. At the current limit, each transaction can allocate up to 14mb, and each block can allocate up to 62mb. The nature of transactions means that it should be fairly for the nodes to block allocate and discard this memory, and even processed in parallel, 62mb is absolutely dwarfed by the size of the UTXO held in memory.

I’d love to hear input from other people with access to benchmarking or environments where the impact of this can be tested, but to me it seems very low risk to start slowly increasing memory units in anticipation of increasing blockchain load.

| Questions | Known adjacent Parameters | |
|---|---------------------------|---|
| 1. What mixture of tx memory and block memory limits should we increase? | max_block_ex_mem | The maximum limit for memory in a block. |
| 2. Are the assumptions from the above informal research valid? | | |
| 3. Does IOG or anyone else have a more extensive framework for evaluating the impacts of these changes? | max_tx_ex_steps | The maximum limit for CPU steps in a transaction. |
| 4. How should we measure the impact of these changes, and minimize our risk in doing so? | max_block_ex | The maximum limit for CPU steps in a block. |

| | | |
|--|----------------|--|
| 5. Should we consider increases to other parameters? | _steps | |
| | max_block_size | The maximum number of bytes that can fit within a block. |
| Desired Output | Other | |
| A specific recommendation for target increases, and any staged roll-out in pursuit of that target. Alternatively, a concrete and evidence based framework that shows the danger of this increase that the community can use to evaluate other increases in the future. | | |