

Contract execution can be parallelized perfectly (ie. maximum Amdahl's speedup) under the transactional memory model as a purely implementation-level optimization, however that would make current block gas limits ineffective: it would be possible to create two blocks with ostensibly identical total gas, but one with no possible parallelism (ie. purely sequential) and another where every

storage access + calls are independent from other transactions. The former would take a much longer time to execute.

This can be solved by requiring block creators to provide an execution schedule tree

and compute a sequential gas limit from it. For example for a block with 3 transactions (#0

- first transaction etc):

Each node can only be executed after all its parents are done. The sequential gas used is defined as maximum use from all execution paths. Assuming for simplicity that step = 1 gas, used sequential gas for a block with this graph is  $30+150+140 = 320$  as that's the most expensive path.

A block is considered invalid if results are different from the (current) purely sequential execution model. This can be checked by tagging every state change (storage+contract creation/destruction) with a changing transaction's index (only temporarily in ram and per-block). Invalid ordering results in a transaction #x

that accesses state changed by #>x.

It's also possible to define 'parallel gas used' as a maximum concurrent gas consumption per step. For the previous tree (and step = 1 gas) that would be 3: for the first 20 steps three instructions are executed in parallel. Thus a 'block parallel gas limit' could be added. At a minimum it must be equivalent to the current highest priced possible individual operation.

The parallelization algorithm used during block creation stays completely out of consensus and is optional: a list is a valid execution schedule too.

All these changes are invisible to everyone else: nothing changes as far as contract writers and users are concerned. The only difference: transactions that are cheaper to parallelize would likely be accepted with a lower gas price.

Even with a single-threaded execution this would help due to parallel db access. A single random read is slow even on a high-end hardware, while throughput for many parallel reads continues to rise - for 960 it's 30x faster:

[

Samsung-960-Pro-2TB-AS-SSD

503×502 22.6 KB

](<https://ethresear.ch/uploads/default/original/1X/70dd1ae443db859ddb635f337d37d96883f97292.png>)

Thoughts? Would you support adding this?