Recently, we decided to stop working on preparing the Merry-Go-Round snapshot sync algorithm in turbo-geth. Why? Because we think there is a way to achieve most of what it would deliver, but without most of the complexity. What is described here will not be a part of the first turbo-geth release. In the first release, the only way to sync would be to download all blocks starting from the Genesis, and execute them. The timing is not that bad - it definitely won't take a month. On a machine with an NVMe device, it can take around 50 hours to obtain the node synced to the head of the mainnet and all history present and indexed.

In the subsequent releases, we would like to introduce the ability to sync from more recent state than genesis. Initial idea is this. Let's say, every 1m blocks (~6 months time), we will manually (or in the future automatically) create a snapshot of the entire state, and of all the blocks and receipts prior to that point. This will result in 3 files (and approximate sizes if this were done about now):

1. State snapshot file, contains all the accounts, contract storage, and contract bytecodes. ~50Gb
2. Blockchain file, containing all block headers and block bodies from genesis up to the snapshot point. ~160Gb
3. Receipt file, containing all the historical receipts from genesis up to the snapshot point. ~130Gb

These 3 files would be seeded on the BitTorrent (and perhaps Swarm) by the turbo-geth nodes (we want to try to plug in the bitTorrent library).

One slight technical challenge for such seeding is the ability to utilise the state snapshot file, while seeding it, otherwise these 50Gb would just be "wasted", meaning this space is only useful for seeding to other nodes, but not for anything else. It should be possible to organise the state database as an overlay, where actively modifiable state "sits" on top of the immutable snapshot file. Anytime we try to read anything from the state, we look up in the modifiable state, and if not found there, we look up in the snapshot file (that means that snapshot file needs to have an index in it).

As you might have guessed, there would be two alternative ways to sync:

1. Download blockchain file, and execute all the blocks from genesis. Result - entire history from genesis, receipts, current state.
2. Download the most recent state snapshot file, download blocks (from eth network) after the snapshot point, and execute blocks after the snapshot points. Result - history only starting from the snapshot point, current state. If historical receipts are required, they can be downloaded as the receipt file.

How large would that "modifiable" state be (the one that sits on top of the state snapshot file as overlay)? Here is some rough calculation. As of the block 10'416'641, there were 92'430'646 accounts and 334'797'797 storage items in the state, or 427'228'443 items in total. That makes 125 bytes per item on average.

Number of modified accounts between blocks 9'416'641 and 10'416'641 was 25'191'312, and number of modified storage items: 88'451'010, or around 13G. This is how large the modified state would grow during 1m blocks. After that point, it will be merged into the snapshot, and the new snapshot will be seeded over the BitTorrent (or Swarm) network.

The second approach to syncing (from recent state snapshot) still requires executing at most 1m blocks. Depending on the performance of implementation, it might take anything from few hours to couple of days.