

Block Download and Import Strategies

Summary

It was observed that importing blocks and downloading blocks from network peers concurrently does not work very well. When importing starts, download peers will terminate when downloading takes place at the same time. On the other hand, keeping them idle while importing from a fully pre-filled queue of blocks seems to be the fastest strategy. When the queue becomes empty, some of the parked peers were still available for download.

- Typical download time to fill a single blocks queue record with 384 blocks is $\sim 15s$ (raw download round-trip time typically ranges 3 times $\sim 30ms.. \sim 2s$ – i.e. $\sim 90ms.. \sim 6s$ – as reported in the logs.)
- Blocks queue capacity is 4 records, so ramp up time to fill up the blocks queue ranges between $\sim 15s$ and $1m$ depending on the number of peers.
- Typical waiting time to recover/reconnect download peers $\sim 5m .. \sim 10m$, these values are anecdotal and taken from the metrics display from the *Base vs Latest* graph, the horizontal segments where there is no increase of block numbers.
- Improvement from *fully concurrent importing blocks and downloading* (see Test 2: Serialised Download or Import) to *serialised importing blocks or downloading* (see Test 3: Partially Concurrent Download and Import): the latter shows roughly double the throughput (i.e. needs half the time of the former.)
- Throughput time improvement for increasing staged queue with *serialised importing blocks or downloading*, see Test 5: Serialised Download or Import, Queue Size 8.

Test configuration

There were several tests runs all on a copy of a pre-loaded *Mainnet* database. The state block number of the database is #21773857. With current *nimbus_execution_client* binary based on PR #3089 from 20/02/2025, most synchronisation runs would eventually stall at block #21790594 due to a state root mismatch reported from the *FC module*.

Configurable parameters for this test

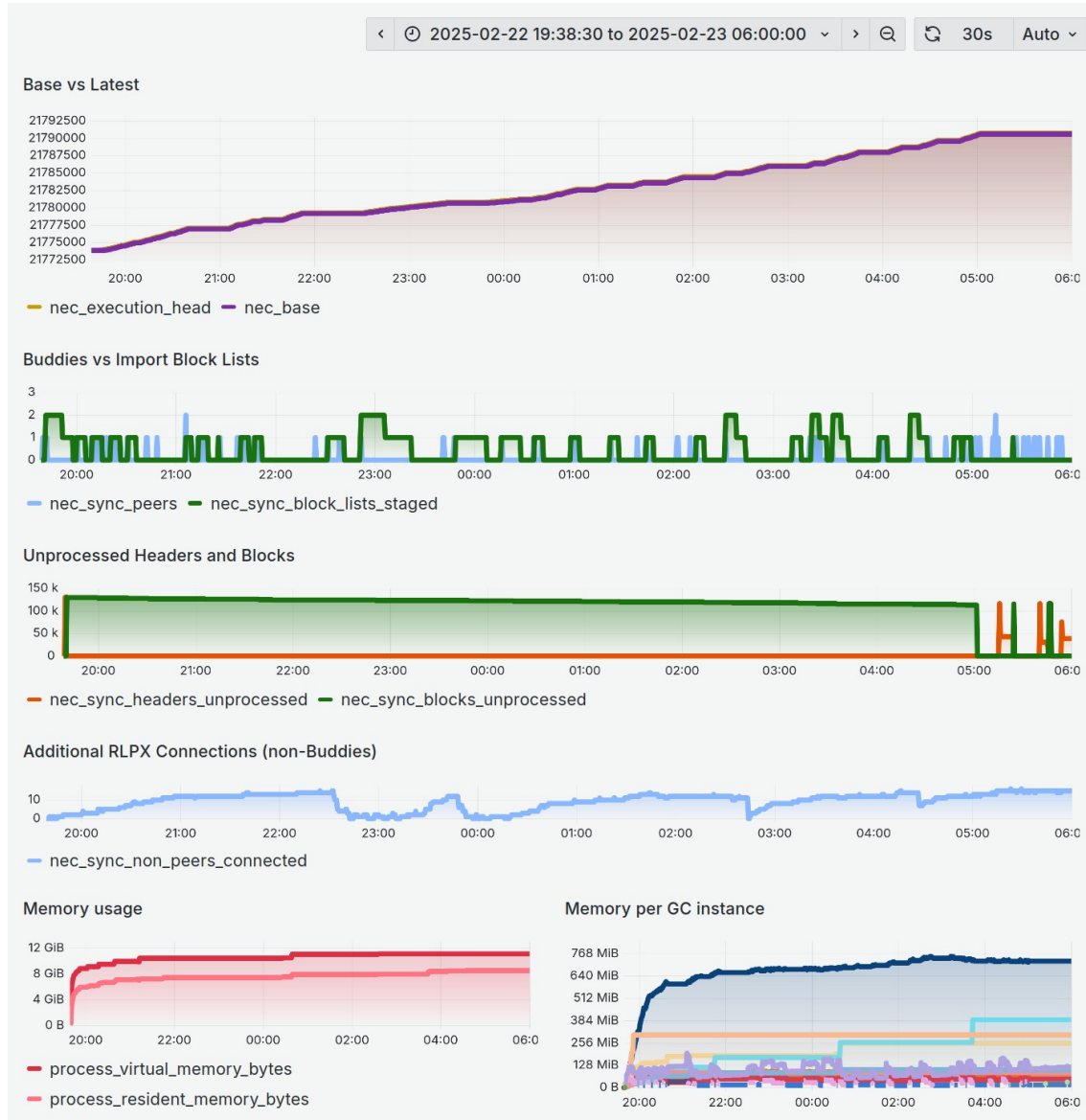
- Git branch: *20250224-sync-throughput-test-config* (Github draft PR #3107)
- `worker_config.nim(34): minPeersImportWhileFetching`
- `worker_config.nim(140): blocksStagedHwmDefault`

Additional considerations

Most tests were run at night when the host system was idle. The exception was Test 4: Serialised Download or Import, Queue Size 5. While the tests were running, *RPC* was full serviced. The *CI* ran on the same host as the test *EL*. There were no extraordinary network timeouts observed (set to $10s$ for the *eth/68* protocol) in the log files.

Test 1: Fully Concurrent Download and Import

- Config: `minPeersImportWhileFetching=0`, `nFetchHeadersBatch=4*nFetchHeadersRequest`
- Block numbers #21773857..#21790735 (16878 blocks)
- Import starts immediately when the block queue has blocks queued
- Fetching blocks from network peers continues concurrently



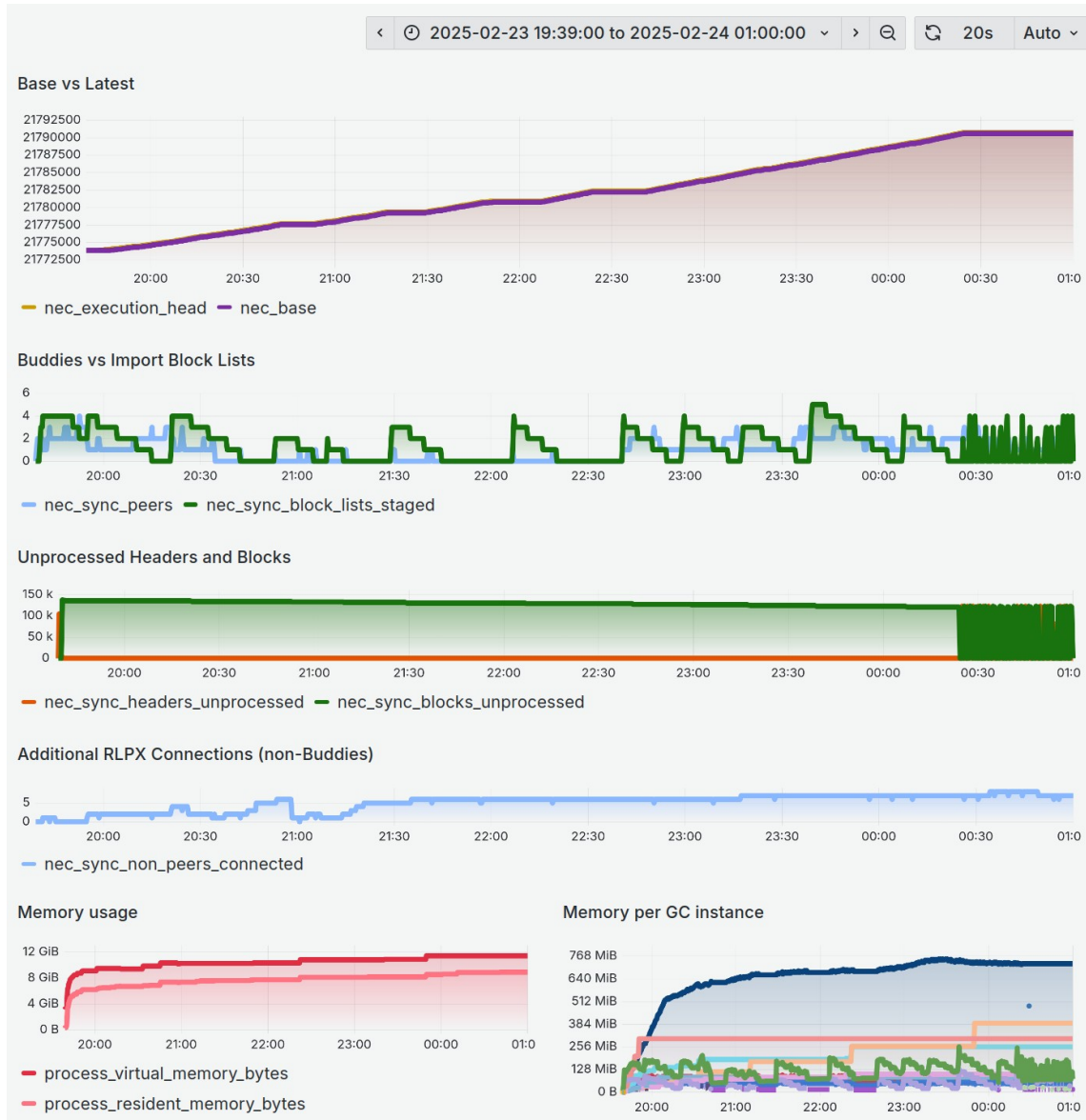
- Active download and import time 19:40:30..05:01:24 (~9h 21m)
- Mean throughput is 30.086 blocks / minute
- First error messages when stalling occurs

```
WRN 2025-02-23 05:01:24.388+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync"
file=blocks_staged.nim:344
n=210 iv=#21790526..#21790741 B=#21790589 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [..]e371571, got: [..]efbf1f9"
```

```
WRN 2025-02-23 05:24:14.139+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync"
file=blocks_staged.nim:344
n=0 iv=#21790736..#21791119 B=#21790589 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [..]e371571, got: [..]efbf1f9"
```

Test 2: Serialised Download or Import

- Config: `minPeersImportWhileFetching=99`, `nFetchHeadersBatch=4*nFetchHeadersRequest`
- Block numbers #21773857..#21790735 (16878 blocks)
- Import starts not before the block queue has been filled up completely
- Fetching blocks from network peers is suspended while importing



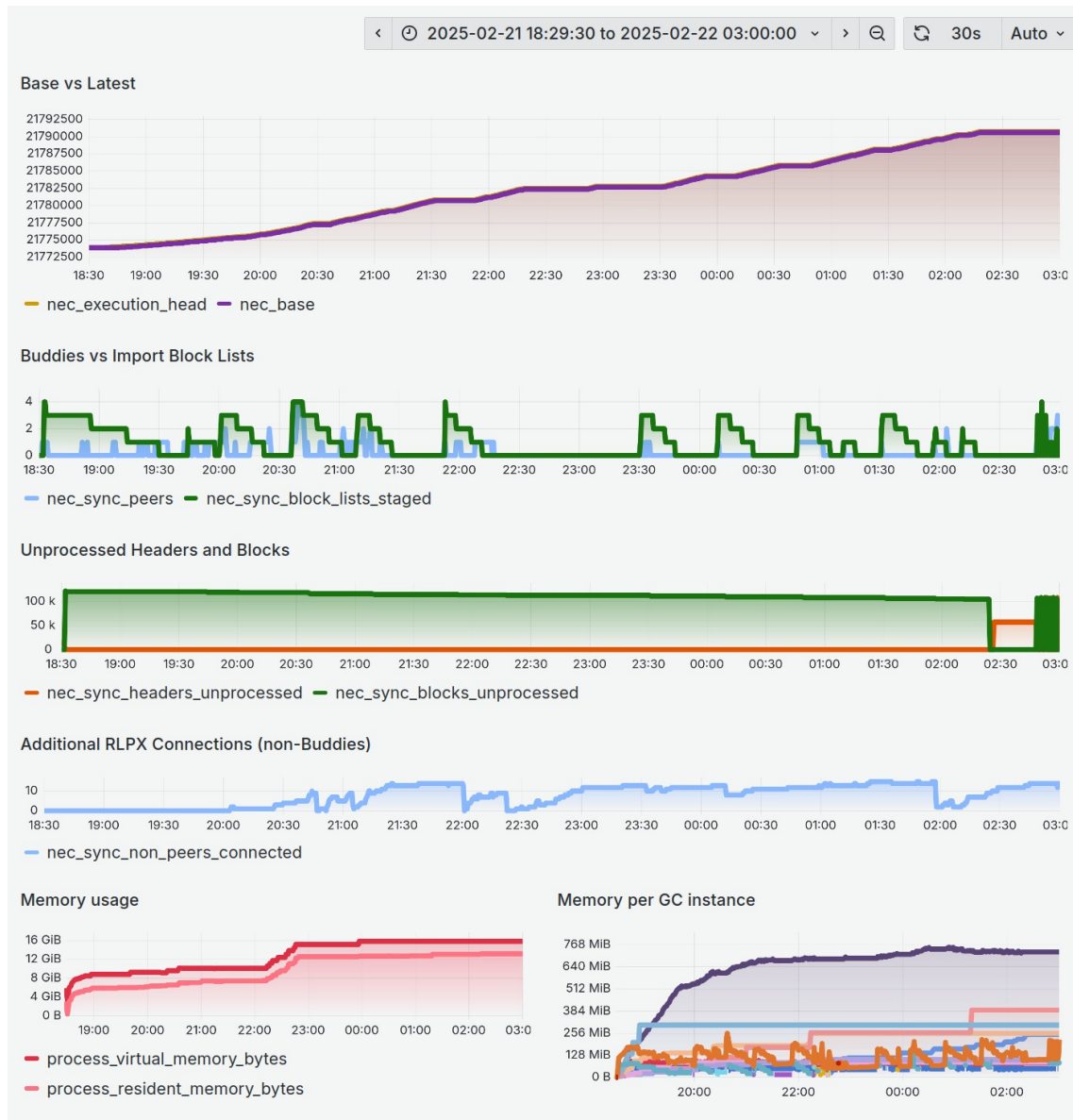
- Active download and import time 19:39:20..00:24:01 (~4h 42m)
- Mean throughput is 59.851 blocks / minute
- First error messages when stalling occurs

```
WRN 2025-02-24 00:24:04.621+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync" file=blocks_staged.nim:344
n=301 iv=#21790435..#21790818 B=#21790594 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [...]e371571, got: [...]efbf1f9"
```

```
WRN 2025-02-24 00:25:52.594+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync" file=blocks_staged.nim:344
n=0 iv=#21790736..#21791119 B=#21790594 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [...]e371571, got: [...]efbf1f9"
```

Test 3: Partially Concurrent Download and Import

- Config: `minPeersImportWhileFetching=3`, `nFetchHeadersBatch=4*nFetchHeadersRequest`
- Block numbers #21773857..#21790735 (16878 blocks)
- Import starts immediately if there are at least *three* active download peers available, otherwise it will start not before the block queue has been filled up completely
- Fetching blocks from network peers continues concurrently if there are at least *three* active download peers available, otherwise it is suspended while importing



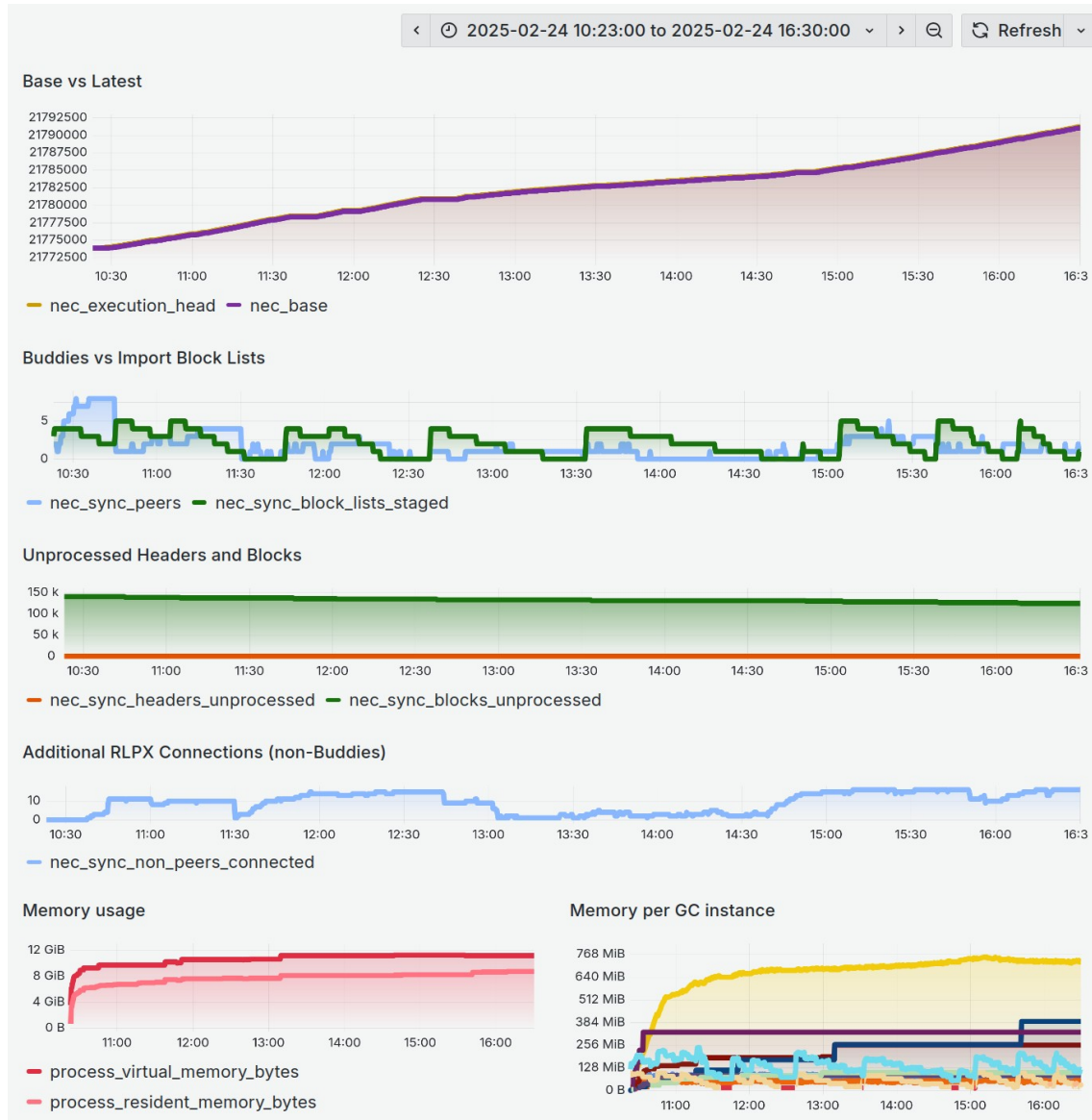
- Active download and import time 18:32:30..02:17:25 (~7h 35m)
- Mean throughput is 37.095 blocks / minute
- First error messages when stalling occurs

```
WRN 2025-02-22 02:17:45.572+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync" file=blocks_staged.nim:344
n=19 iv=#21790717..#21790955 B=#21790588 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [..]e371571, got: [..]efbf1f9"
```

```
WRN 2025-02-22 02:48:55.938+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync" file=blocks_staged.nim:344
n=0 iv=#21790736..#21791119 B=#21790588 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [..]e371571, got: [..]efbf1f9"
```

Test 4: Serialised Download or Import, Queue Size 5

- Config: `minPeersImportWhileFetching=99`, `nFetchHeadersBatch=5*nFetchHeadersRequest`
- Block numbers #21773857..#21791205 (17348 blocks)
- Import starts not before the block queue has been filled up completely
- Block queue is increased from 4 to 5 records (each holds 384 blocks)
- Fetching blocks from network peers is suspended while importing



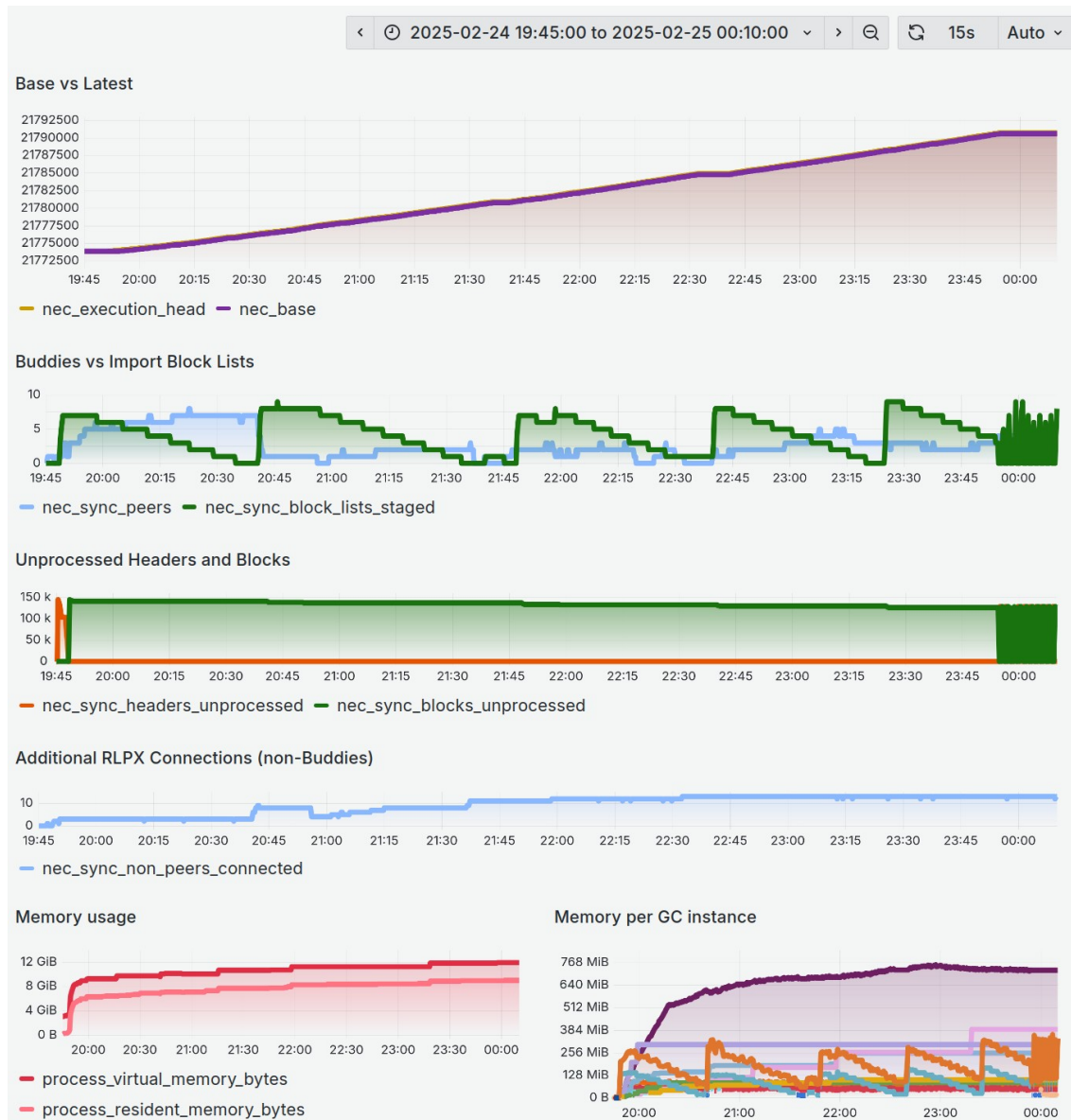
- Active download and import time 10:23:00..16:30:00 (~6h 7m)
- Mean throughput is 47.270 blocks / minute
- Manually stopped (no stalling)

This one seems to be an outlier because

- It was run at daytime rather than nighttime.
- Even though the *Buddies vs Import Block Lists* is smooth, the overall time is greater than the one from Test 2: Serialised Download or Import.
- All other tests eventually stall with a *state root mismatch error* reported from the *FC module*.

Test 5: Serialised Download or Import, Queue Size 8

- Config: `minPeersImportWhileFetching=99`, `nFetchHeadersBatch=8*nFetchHeadersRequest`
- Block numbers #21773857..#21791205 (17348 blocks)
- Import starts not before the block queue has been filled up completely
- Block queue is increased from 4 to 8 records (each holds 384 blocks)
- Fetching blocks from network peers is suspended while importing



- Active download and import time 19:49:15..23:54:07 (~4h 5m)
- Mean throughput is 70.808 blocks / minute
- First error messages when stalling occurs

```
WRN 2025-02-24 23:54:07.290+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync" file=blocks_staged.nim:344
n=43 iv=#21790693..#21791076 B=#21790596 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [...]e371571, got: [...]efbf1f9"
```

```
WRN 2025-02-24 23:55:47.307+00:00
RunDaemon: import block error (reorg triggered)
topics="beacon sync" file=blocks_staged.nim:344
n=0 iv=#21790736..#21791119 B=#21790596 L=#21790735
nthBn=#21790736 nthHash=d2c0ecbefd74
error="stateRoot mismatch, expect: [...]e371571, got: [...]efbf1f9"
```