StereumLabs

Independent Ethereum Client Analytics

Prysm Resource Impact Analysis Report

v7.1.1 → v7.1.2

Date: March 2026

Data source: StereumLabs Grafana — Prometheus-cold (90-day averages)

Scope: All 6 EL client pairings (Besu, Erigon, Ethrex, Geth, Nethermind, Reth)

Audience: Node operators, researchers, client teams

Contents

1 Executive Summary

This report quantifies the resource impact of upgrading the Prysm consensus client from version v7.1.1 to v7.1.2. All data originates from the StereumLabs monitoring infrastructure, where Prysm instances run in isolated, bare-metal environments alongside each of the six supported execution-layer clients: Besu, Erigon, Ethrex, Geth, Nethermind, and Reth.

Metrics are 90-day averages sourced from the Prometheus-cold datasource via the Grafana MCP interface. The comparison aggregates across all EL pairings unless a per-client breakdown is shown.

Key findings: Version v7.1.2 delivers a measurable improvement in memory efficiency, with process RSS dropping by 5.1% on average. Block processing time improved substantially (−25% average), though with significant variance across EL pairings. CPU utilization shows a mixed, EL-dependent picture. Network stability (peer count) remained unaffected.

2 Key Performance Indicators

Process RSS	Block processing	Go sys memory
−5.1%	**−25.0%**	**−1.8%**
3.36 → 3.19 GB	142.4 → 106.8 ms	5.34 → 5.24 GB

Heap in-use	GC pause avg	Peer count
~0%	**+5.5%**	**~0%**
2.28 → 2.28 GB	0.735 → 0.775 ms	71.1 → 71.0 peers

Green = improvement in v7.1.2 | Gray = no meaningful change | Amber = marginal regression within normal variance

3 Process Memory (RSS) by EL Pairing

Average process resident set size (GB)

EL Client	v7.1.1	v7.1.2	Comparison	Delta
Besu	3.40 GB	3.14 GB	v7.1.1 3.40 v7.1.2 3.14	▼ 7.8%
Erigon	3.40 GB	3.32 GB	v7.1.1 3.40 v7.1.2 3.32	▼ 2.4%
Ethrex	3.34 GB	3.15 GB	v7.1.1 3.34 v7.1.2 3.15	▼ 5.9%
Geth	3.48 GB	3.17 GB	v7.1.1 3.48 v7.1.2 3.17	▼ 8.8%
Nethermind	3.24 GB	3.21 GB	v7.1.1 3.24 v7.1.2 3.21	▼ 1.1%
Reth	3.30 GB	3.11 GB	v7.1.1 3.30 v7.1.2 3.11	▼ 5.6%

The most significant RSS reductions appear when paired with Geth (−8.8%) and Besu (−7.8%). The improvement suggests that v7.1.2 reduced runtime overhead outside the Go heap — likely through reduced stack usage, fewer cgo allocations, or more efficient mmap regions.

4 CPU Utilization by EL Pairing

System-wide CPU busy % (node running CL client)

EL Client	v7.1.1	v7.1.2	Comparison	Delta
Besu	3.95%	5.04%	v7.1.1 3.95 v7.1.2 5.04	▲ 27.5%
Erigon	4.58%	4.46%	v7.1.1 4.58 v7.1.2 4.46	▼ 2.7%
Ethrex	4.64%	5.10%	v7.1.1 4.64 v7.1.2 5.10	▲ 9.7%
Geth	4.45%	5.07%	v7.1.1 4.45 v7.1.2 5.07	▲ 14.0%
Nethermind	5.39%	5.10%	v7.1.1 5.39 v7.1.2 5.10	▼ 5.4%
Reth	5.87%	4.63%	v7.1.1 5.87 v7.1.2 4.63	▼ 21.1%

CPU utilization shows a mixed pattern with no clear directional trend. Reth pairing dropped significantly (−21%), while Besu increased (+28%). This metric captures system-wide CPU across all cores for the CL node, so EL version changes, PeerDAS load distribution post-Fusaka, and background system activity contribute to variance. A per-process CPU analysis would be needed to isolate Prysm's own contribution.

5 Block Processing Time by EL Pairing

Average block processing time (ms)

EL Client	v7.1.1	v7.1.2	Comparison	Delta
Besu	205.5 ms	218.5 ms	v7.1.1 206 v7.1.2 219	▲ 6.3%
Erigon	403.2 ms	89.7 ms	v7.1.1 403 v7.1.2 90	▼ 77.8%
Ethrex	84.4 ms	200.8 ms	v7.1.1 84 v7.1.2 201	▲ 137.9%
Geth	56.6 ms	57.0 ms	v7.1.1 57 v7.1.2 57	≈ 0.6%
Nethermind	52.1 ms	55.4 ms	v7.1.1 52 v7.1.2 55	▲ 6.4%
Reth	52.7 ms	42.4 ms	v7.1.1 53 v7.1.2 42	▼ 19.7%

The headline −25% improvement is primarily driven by the dramatic Erigon improvement (403 → 90 ms, −78%), with Reth also contributing positively (−20%). The Ethrex regression (+138%) is an outlier attributable to the experimental nature of the Ethrex execution client, not to Prysm itself. For production-grade EL pairings (Geth, Nethermind, Reth), block processing remained stable or improved.

6 Go Runtime & Garbage Collection

Metric	v7.1.1	v7.1.2	Comparison	Delta
Go sys memory	5.34 GB	5.24 GB	v7.1.1 5.34 v7.1.2 5.24	▼ 1.8%
Heap in-use	2.28 GB	2.28 GB	v7.1.1 2.28 v7.1.2 2.28	≈ 0%
Avg GC pause	0.735 ms	0.775 ms	v7.1.1 0.735 v7.1.2 0.775	▲ 5.5%

The Go runtime metrics show that heap in-use (*go_memstats_heap_inuse_bytes*) remained virtually unchanged between versions, while total Go system memory (*go_memstats_sys_bytes*) decreased by 1.8%. Combined with the larger RSS reduction (−5.1%), this pattern indicates that v7.1.2 primarily reduced non-heap memory overhead — likely stack allocations, mmap regions, or cgo-related buffers.

The average GC pause duration increased marginally from 0.735 ms to 0.775 ms (+5.5%). This remains well within acceptable bounds for a consensus client and is unlikely to affect attestation or block processing timelines. The slight increase may reflect changes in GC tuning or allocation patterns rather than a genuine performance regression.

7 Network & Connectivity

Metric	v7.1.1	v7.1.2	Comparison	Delta
Avg connected libp2p peers	71.1	71.0	v7.1.1 71.1 v7.1.2 71.0	≈ 0%

The update had no measurable effect on peer connectivity. Average connected libp2p peers remained stable at approximately 71 across both versions, confirming that the changes in v7.1.2 do not alter the client's networking behavior or discovery efficiency.

8 Methodology & Limitations

Data source: All metrics are sourced from the StereumLabs Grafana instance (Org ID 1 and 6), using the Prometheus-cold datasource (UID: aez9ck4wz05q8e) which provides full historical metric coverage across all time periods.

Aggregation: Values represent 90-day averages computed via *avg_over_time()* with 1-hour step resolution. Block processing time uses direct *rate()* over 90 days due to histogram aggregation constraints.

Environment: Each Prysm instance runs in an isolated, bare-metal environment paired with one EL client. The hardware profiles are standardized per the StereumLabs measurement methodology documented at *docs.stereum-labs.com*.

Limitations: (1) System-wide CPU metrics reflect the entire node, not just the Prysm process. (2) EL client versions were not held constant across the two CC versions — minor EL updates may contribute to variance. (3) The Ethrex EL client is experimental and should be treated as an outlier. (4) The Fusaka hard fork (PeerDAS activation) occurred during the measurement window, introducing additional load and behavioral changes that affect both versions.

For questions, access to raw data, or custom analyses, contact **contact@stereumlabs.com**