

ВПЛИВ ЯВИЩА STATE BLOAT НА ПРОДУКТИВНІСТЬ ВАЛІДАТОРІВ У БЛОКЧЕЙН-МЕРЕЖАХ НА БАЗІ ETHEREUM

Вінницький національний технічний університет

Анотація

У роботі проведено аналіз проблеми неконтрольованого зростання обсягу даних стану (State Bloat) у блокчейн-мережах на базі протоколу Ethereum. Досліджено механізми деградації продуктивності вузлів-валідаторів через збільшення глибини дерева Merkle Patricia Trie та виникаючу латентність підсистеми пам'яті. Розглянуто апаратні обмеження сучасних серверних архітектур та запропоновано вектори оптимізації через впровадження архітектурно-орієнтованих алгоритмів.

Ключові слова: блокчейн, Ethereum, State Bloat, Merkle Patricia Trie, валідатор, продуктивність, Hyperledger Besu.

Abstract

This paper analyzes the issue of uncontrolled state data growth (State Bloat) in blockchain networks based on the Ethereum protocol. The study investigates the mechanisms of validator node performance degradation resulting from the increased depth of the Merkle Patricia Trie and the subsequent memory subsystem latency. Furthermore, hardware limitations of modern server architectures are explored, and optimization pathways are proposed through the implementation of architecture-aware algorithms.

Keywords: blockchain, Ethereum, State Bloat, Merkle Patricia Trie, validator, performance, Hyperledger Besu.

Вступ

Трансформація блокчейн-мереж від платіжних систем до глобальних обчислювальних платформ призвела до виникнення критичної проблеми масштабування — «State Bloat». У мережах на базі Ethereum стан (state) включає баланси всіх активних рахунків, код смарт-контрактів та дані їхнього внутрішнього сховища (storage slots). На відміну від історії транзакцій, яку можна архівувати, дані стану є «живими»: вони необхідні валідатору в режимі реального часу для верифікації кожного нового блоку. Станом на початок 2026 року, розмір повного стану мережі Ethereum продовжує зростати експоненціально, що створює бар'єри для входу нових учасників та підвищує ризики централізації [1].

Архітектурна специфіка Merkle Patricia Trie (MPT)

Для збереження та криптографічного підтвердження стану Ethereum використовує 16-арне префіксне дерево — Merkle Patricia Trie. Кожен вузол дерева ідентифікується за його криптографічним хешем [2]. Зі зростанням кількості облікових записів та DeFi-протоколів, глибина дерева збільшується, що призводить до наступних наслідків:

- Кожна операція читання або запису вимагає послідовного обходу вузлів від кореня до листка.
- У програмній інженерії цей процес визначається як «Pointer Chasing», де адреса наступного вузла стає відомою лише після обробки поточного.
- Даний патерн доступу є найменш ефективним для сучасних конвеєрних процесорів через неможливість ефективного предиктивного завантаження даних (Hardware Prefetching).

Деградація підсистеми I/O та ієрархії пам'яті

Основний вплив State Bloat проявляється через критичне зростання латентності доступу до даних. Коли обсяг MPT перевищує ємність доступної оперативної пам'яті (RAM), блокчейн-клієнти, такі як Hyperledger Besu, змушені звертатися до дискового сховища (SSD) [3]. Навіть при використанні NVMe-накопичувачів, різниця в латентності між L3-кешем процесора та SSD становить кілька порядків:

1. L3 Cache: ~10-15 нс.
2. RAM (DRAM): ~80-100 нс.
3. NVMe SSD: ~10 000 - 50 000 нс.

Це призводить до того, що процесор проводить більшу частину часу в очікуванні даних (I/O Wait), що знижує загальну пропускну здатність вузла (TPS). Використання сучасних процесорів із великим об'ємом L3-кешу дозволяє частково компенсувати цей ефект шляхом утримання верхніх рівнів дерева в пам'яті процесора [4].

Аналіз стратегій зберігання у Hyperledger Besu

Блокчейн-клієнт Hyperledger Besu пропонує кілька стратегій для боротьби з розростанням стану, зокрема Bonsai Tries та Forest [5].

- Bonsai Tries: Використовує плоску структуру даних для зберігання стану, що дозволяє уникнути надлишкового дублювання вузлів при кожному оновленні. Це знижує навантаження на дискову підсистему, проте вимагає інтенсивних обчислень для реконструкції дерева при верифікації блоків [5].
- Forest: Використовує класичний підхід до збереження всіх версій дерева, що призводить до швидкого заповнення дискового простору, але полегшує доступ до історичних даних стану [5]. При інтенсивному потоці транзакцій, стратегія Bonsai Tries є більш стійкою до State Bloat, проте вона стає вразливою до промахів кешу на рівні LLC (Last Level Cache).

Зростання часу обробки транзакцій через State Bloat безпосередньо впливає на стабільність мережі. Якщо валідатор витрачає занадто багато часу на виконання блоку (Block Execution Time), він ризикує не встигнути розповсюдити результати верифікації в межах встановленого слота (slot duration).

Це призводить до:

1. Збільшення кількості пропущених слотів (Missed Slots).
2. Зростання ймовірності тимчасових розгалужень (Forks) через затримку фіналізації.
3. Підвищення апаратного порогу входу, що виштовхує дрібних валідаторів із мережі на користь великих серверних центрів [1].

Висновки

Аналіз підтверджує, що State Bloat є не просто проблемою дефіциту дискового простору, а системним викликом для архітектури обчислювальних вузлів. Традиційні методи оптимізації вичерпують свій потенціал. Необхідним є впровадження архітектурно-орієнтованих алгоритмів планування, які б враховували ієрархію кеш-пам'яті та структуру дерева станів для мінімізації латентності. Експериментальна частина подальшої роботи буде зосереджена на тестуванні гіпотези про ефективність префіксної кластеризації транзакцій для стабілізації часу виконання блоків у високонавантажених режимах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Wood, G. (2014). *Ethereum: A secure decentralised generalised liveness ledger*. Ethereum Project Yellow Paper.
2. Buterin, V. (2021). *State expiry and statelessness*. Ethereum Research.
3. Mittal, S. (2016). A survey of techniques for managing last-level caches. *ACM Computing Surveys (CSUR)*, 48(3).
4. Kushwaha, S. S., et al. (2022). *Performance Analysis of Hyperledger Besu for Private Blockchain Networks*. IEEE International Conference on Blockchain.
5. Hyperledger Besu Documentation. (2025). *Bonsai Tries vs Forest Storage Strategies*.

Бистрик Максим Васильович – аспірант групи F2-125a, факультет Інформаційних технологій та комп'ютерних інженерії, Вінницький національний технічний університет, Вінниця, email: max.bystryk@gmail.com

Науковий керівник: **Хошаба Олександр Мирославович** - к.т.н, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, e-mail: oleksandr.khoshaba@gmail.com

Maksym Bystryk – PhD Student, group F2-125a, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: max.bystryk@gmail.com

Supervisor: **Oleksandr Khoshaba** – PhD, Associate Professor, Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: oleksandr.khoshaba@gmail.com